Classification and Regression with the BigML Dashboard

The BigML Team

Version 2.3



MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

Copyright © 2024, BigML, Inc.

Copyright© 2024, BigML, Inc., All rights reserved.

info@bigml.com

BigML and the BigML logo are trademarks or registered trademarks of BigML, Inc. in the United States of America, the European Union, and other countries.

BigML Products are protected by US Patent No. 11,586,953 B2; 11,328,220 B2; 9,576,246 B2; 9,558,036 B1; 9,501,540 B2; 9,269,054 B1; 9,098,326 B1, NZ Patent No. 625855, and other patent-pending applications.

This work by BigML, Inc. is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. Based on work at http://bigml.com.

Last updated June 7, 2024

About this Document

This document provides a comprehensive description of how to solve classification and regression problems using the BigML Dashboard. BigML models, ensembles, linear regressions, logistic regressions, deepnets, and fusions are covered in detail. Learn how to use the BigML Dashboard to configure, visualize, and interpret these supervised predictive models and use them to create evaluations of their performance and real-time predictions or batch predictions.

This document assumes that you are familiar with:

- Sources with the BigML Dashboard. The BigML Team. June 2016. [11]
- Datasets with the BigML Dashboard. The BigML Team. June 2016. [9]

To learn how to use the BigML Dashboard to build time series and unsupervised models read:

- Time Series with the BigML Dashboard. The BigML Team. July 2017. [12]
- Cluster Analysis with the BigML Dashboard. The BigML Team. June 2016. [8]
- Anomaly Detection with the BigML Dashboard. The BigML Team. June 2016. [6]
- Association Discovery with the BigML Dashboard. The BigML Team. June 2016. [7]
- Topic Modeling with the BigML Dashboard. The BigML Team. November 2016. [13]

Contents

1	Mod		1
	1.1		uction
	1.2	Under	standing BigML Models
		1.2.1	Streaming Model
		1.2.2	In-Memory Model
		1.2.3	Scoring and Splitting
			1.2.3.1 Early Splitting
		1.2.4	Pruning 4
		1.2.5	Field Importance
		1.2.6	Confidence and Probability 5
			1.2.6.1 Confidence
			1.2.6.1.1 Formulation
			1.2.6.1.2 Interpretation
			1.2.6.2 Probability
		1.2.7	Expected Error
			1.2.7.1 Formulation
			1.2.7.2 Interpretation
		1.2.8	Models with Images
	1.3		ng Models with 1-Click
	1.4		Configuration Options
		1.4.1	Objective Field
			Automatic Optimization
			1.4.2.1 Training duration
			1.4.2.2 Model candidates
		1.4.3	Pruning
		1.4.4	Missing Splits
		1.4.5	Node Threshold
		1.4.6	Weight Options
		1.4.0	1.4.6.1 Balance Objective 16
			1.4.6.2 Objective Weights
			1.4.6.3 Weight Field
		1.4.7	Sampling Options
		1.4.7	
			0
			1.4.7.3 Sampling
			1.4.7.4 Replacement
		4.4.0	1.4.7.5 Out of Bag
		1.4.8	Advanced Ordering
			1.4.8.1 Deterministic Shuffling 19
			1.4.8.2 Linear Shuffling 19
			1.4.8.3 Random Shuffling
		1.4.9	Creating Models with Configured Options
		1.4.10	API Request Preview

1.5	Visualizing BigML Models	21
		21
		21
		22
		23
		24
		25
		31
1.6		33
		33
		34
1.7		35
	0	35
		36
		37
		38
		41
		41
		41 45
		45 50
	0 0	51
	5 5	51
		52
		55
		55
		56
	5	57
	•	57
	I	59
		60
		60
		61
		62
	9	63
		63
	1.7.5.2 Using Model Predictions via the BigML Bindings	63
	1.7.6 Descriptive Information	64
	1.7.6.1 Name	65
		65
	1.7.6.3 Category	65
	1.7.6.4 Tags	66
	1.7.7 Model Predictions Privacy	66
		67
		67
		67
1.8		68
_		68
		69
		70
1.9		70
		70
1.10		71
		71
		72
	5 7	73
	5	73 73
4 4 4		73 74
1.12	2 Models in the BigML Gallery	74

		1.12.1	Publishir	ng Models in the	Gallery	 		 								 75
		1.12.2	Cloning	Models From Ga	llery .	 		 								 77
	1.13			to Another Proje												79
	1.14	Stoppi	ng Model	Creation		 		 								 80
																81
																83
						 • •	•••	 	• •	• •	 •	• •	• •	•		
2	Ens	embles														85
	2.1	Introdu	iction			 		 								 85
	2.2	Unders	standing I	Ensembles		 		 								 87
		2.2.1	-	Forests Versus												87
			2.2.1.1	Single trees .												87
			2.2.1.2	Ensemble predi												88
		2.2.2		portance												88
		2.2.3		les with Images												89
	2.3			bles with 1-Click												90
	2.4			iguration Options												91
	2.4	2.4.1		e Field												93
				ic Optimization												93
		2.4.2														93 95
			2.4.2.1	Training duration												
		0.4.0														96
		2.4.3														96
		2.4.4		of Models												97
		2.4.5		of Iterations												98
		2.4.6														99
			2.4.6.1	Pruning												99
			2.4.6.2	Missing Splits												99
			2.4.6.3	Node Threshold												99
			2.4.6.4	Randomize and												99
		2.4.7	Boosting	Parameters		 		 								 100
			2.4.7.1	Early stopping		 		 								 100
			2.4.7.2	Learning rate .		 		 								 101
		2.4.8	Weight F	Field Options												
			2.4.8.1	Balance Objecti												
			2.4.8.2	Objective Weigh												
			2.4.8.3	Weight Field .												
		2.4.9	Trees Sa	mpling												
				Rate												103
			2.4.9.2	Sampling												
			-	Replacement .												
		2410		Sampling												
		2.4.10		Rate												
				Range												
				Sampling												
				Replacement .												
		0 4 1 1		Out of Bag												
		2.4.11		d Ordering												
				Deterministic SI												
				Linear Shuffling												
				Random Shuffli												
				Ensembles with												
	_			uest Preview .												
	2.5			embles												
		2.5.1		ependence Plot												
			2.5.1.1													
				Interpreting Par												
		2.5.2		st												
	2.6	BigML	Ensembl	e Predictions .		 		 								 119

		Introduction
	2.6.2	Creating Ensemble Predictions
		2.6.2.1 Predict
		2.6.2.1.1 Local Predictions
		2.6.2.1.2 Predictions with Images
		2.6.2.2 Batch Predictions
		2.6.2.2.1 Batch Prediction with Images
	2.6.3	Configuring Ensemble Predictions
		2.6.3.1 Missing Strategies
		2.6.3.2 Combine single tree predictions: probability, confidence or votes 136
		2.6.3.3 Probability, confidence, and votes thresholds
		2.6.3.4 Default Numeric Value
		2.6.3.5 Field Mapping
		2.6.3.6 Output Settings
	261	Visualizing Ensemble Predictions
	2.0.4	2.6.4.1 Single Predictions
		2.6.4.1.1 Prediction explanation
		2.6.4.2 Batch Predictions
		2.6.4.2.1 Output File
		2.6.4.2.2 Output Dataset
		2.6.4.2.3 Batch Prediction 1-Click Actions
	2.6.5	Consuming Ensemble Predictions 152
		2.6.5.1 Using Ensemble Predictions via the BigML API
		2.6.5.2 Using Ensemble Predictions via the BigML Bindings
	2.6.6	Descriptive Information
		2.6.6.1 Name
		2.6.6.2 Description
		2.6.6.3 Category
		2.6.6.4 Tags
	2.6.7	Ensemble Predictions Privacy
	2.6.8	Moving Ensemble Predictions to Another Project
	2.6.9	Stopping Ensembles Predictions
		Deleting Ensemble Predictions
07		
2.7		ming Ensembles
	2.7.1	Exporting and Downloading Ensembles
		Using Ensembles Via the BigML API
		Using Ensembles Via the BigML Bindings
		ble Limits
2.9	Descri	ptive Information
	2.9.1	Name
	2.9.2	Description
	2.9.3	Category
	2.9.4	Tags
	2.9.5	Counters
2.10	Ensem	ble Privacy
		Ensembles to Another Project
	-	ng Ensemble Creation
		g Ensembles
		vays
2.14	Tancav	vays
Line	ar Reg	ressions 168
3.1		ction
		standing Linear Regressions
0.2	3.2.1	
	-	Input Field Transformations
	3.2.2	Missing Values
		Number of Predictors
		III-Conditioned Problems
3.3	Creatir	ng Linear Regressions with 1-Click

3

3.4	Linear	Regression Configuration Options 174
	3.4.1	Objective Field
	3.4.2	
		3.4.2.1 Training duration
		3.4.2.2 Linear regression candidates
	3.4.3	Default Numeric Value
	3.4.4	Weights
		3.4.4.1 Weight Field
	3.4.5	Bias
	3.4.6	Field Codings
		3.4.6.1 Dummy Coding
		3.4.6.2 Contrast Coding
	0 4 7	3.4.6.3 Other Coding
	3.4.7	Sampling Options
		3.4.7.1 Rate
		3.4.7.2 Range
		3.4.7.3 Sampling
		3.4.7.4 Replacement 198 3.4.7.5 Out of bag 198
	3.4.8	
	3.4.0	3.4.8.1 Deterministic Shuffling
		3.4.8.2 Linear Shuffling
		3.4.8.3 Random Shuffling
	349	Creating Linear Regressions with Configured Options
		API Request Preview
3.5		izing Linear Regressions
0.0		Linear Regression Chart
		Coefficient Table
3.6		Regression Predictions
0.0	3.6.1	Introduction
	3.6.2	Creating Linear Regression Predictions
		3.6.2.1 Predict
		3.6.2.2 Batch Predictions
	3.6.3	Configuring Linear Regression Predictions
		3.6.3.1 Default Numeric Value
		3.6.3.2 Excluded Fields
		3.6.3.3 Fields Mapping
		3.6.3.4 Output Settings
		3.6.3.5 Prediction explanation
	3.6.4	Consuming Linear Regression Predictions
		3.6.4.1 Local Predictions
		3.6.4.2 Using Linear Regression Predictions via the BigML API
		3.6.4.3 Using Linear Regression Predictions via BigML Bindings
	3.6.5	Descriptive Information
		3.6.5.1 Name
		3.6.5.2 Description
		3.6.5.3 Category
		3.6.5.4 Tags
	3.6.6	Linear Regression Predictions Privacy
	3.6.7	Moving Linear Regression Predictions to Another Project
	3.6.8	Stopping Linear Regression Predictions
07	3.6.9	Deleting Linear Regression Predictions
3.7		Iming Linear Regressions
	3.7.1 3.7.2	Downloading Linear Regressions
		Using Linear Regressions Via the BigML API
3.8	3.7.3 Linoar	Using Linear Regressions Via the BigML Bindings
5.0	Lineal	3.8.0.1 Field Limits 236

		3.8.1	Chart Li	mits								 	 	 	236
				ent Table Lim											
	3.9			rmation											
	0.9														
				egression N											
		3.9.2		ion											
		3.9.3		у											
			-												
				S											
			<u> </u>	ion Privacy .											
				Regressions											
				^r Regression											
	3.13	Deletir	ng Linear	Regressions								 	 		243
	3.14	Takea	ways									 	 		245
4			egressior												246
	4.1	Introdu	uction									 	 		246
	4.2	Under	standing I	Logistic Reg	ressions	s						 	 		247
		4.2.1		eld Transform											
		4.2.2		Values											
				Regressions											
	4.3			c Regressio											
	4.4			sion Configu											
	7.7			e Field											
				ic Optimizati											
		4.4.2		Training dur											
		4.4.0		Logistic reg											
		4.4.3	•												
			4.4.3.1												
				Objective W											
			4.4.3.3	0											
		4.4.4		Numeric Valu											
		4.4.5		Numerics .											
		4.4.6	Eps									 	 		260
		4.4.7	Stats									 	 		261
		4.4.8	Bias and	Auto-Scalin	g							 	 		263
		4.4.9	Regulari	zation								 	 		264
		4.4.10	Field Co	dings								 	 		264
			4.4.10.1	One-hot Co	ding .							 	 		264
			4.4.10.2	Dummy Co	ding .							 	 	 	265
				Contrast Co											
				Other Codir											
		4.4.11		g Options											
				Rate											
				Range											
				Sampling .											
				Replaceme											
				Out of bag											
		1 / 12		Logistic Reg											
	4 5			uest Preview											
	4.5			stic Regress											
		4.5.1	-	Regression (
		4.5.2		ent Table											
				Coefficient											
	4.6	-	-	sion Predicti											
		4.6.1		tion											
		4.6.2	-	Logistic Reg	-										
			4.6.2.1	Predict											
			4.6.2.1	1.1 Logistic	Regre	ssion F	redict	ion v	vith I	mag	es	 	 		305

5

		4.6.2.2 Batch Predictions
		4.6.2.2.1 Batch Prediction with Images
	4.6.3	Configuring Logistic Regression Predictions
		4.6.3.1Probability threshold3164.6.3.2Default Numeric Value318
		4.6.3.3 Fields Mapping
		4.6.3.4 Output Settings
	4.6.4	Visualizing Logistic Regression Predictions
		4.6.4.1 Single Predictions
		4.6.4.1.1 Local Predictions
		4.6.4.1.2 Prediction explanation
		4.6.4.2 Batch Prediction
		4.6.4.2.1 Output CSV file
		4.6.4.2.2 Output Dataset
	4.6.5	Consuming Logistic Regression Predictions
		4.6.5.1 Using Logistic Regression Predictions via the BigML API
	466	4.6.5.2 Using Logistic Regression Predictions via BigML Bindings
	4.6.6	Descriptive Information 329 4.6.6.1 Name 329
		4.6.6.2 Description
		4.6.6.3 Category
		4.6.6.4 Tags
	4.6.7	Logistic Regression Predictions Privacy
	4.6.8	Moving Logistic Regression Predictions to Another Project
	4.6.9	Stopping Logistic Regression Predictions
	4.6.10	Deleting Logistic Regression Predictions
4.7		ming Logistic Regressions
	4.7.1	Downloading Logistic Regressions
	4.7.2	Using Logistic Regressions Via the BigML API
4.0	4.7.3	Using Logistic Regressions Via the BigML Bindings
4.8	Logisti	c Regression Limits
	4.8.1	Chart Limits
	4.8.2	Coefficient Table Limits
4.9		ptive Information
	4.9.1	Logistic Regression Name
	4.9.2	
	4.9.3	Category
	4.9.4	Tags
	4.9.5	Counters
		c Regression Privacy
		Logistic Regressions to Another Project
		ng Logistic Regressions
		ng Logistic Regressions
4.14	Takea	vays
Dee	pnets	346
		lction
5.2	Under	standing Deepnets
	5.2.1	Convolutional Neural Network
		Automatic Parameter Optimization
	5.2.3	Deepnets Use Cases
	5.2.4	Missing Values
5.3		ng Deepnets with 1-Click
5.4	Deepn 5.4.1	et Configuration Options
	5.4.1	
		Default Numeric Value

	5.4.4	Missing Numerics
	5.4.5	Training Duration
	5.4.6	Maximum Iterations
	5.4.7	Network Architecture
		5.4.7.1 Hidden Layers
		5.4.7.2 Learn Residuals
		5.4.7.3 Batch Normalization
		5.4.7.4 Tree Embedding
	5.4.8	Algorithm
		5.4.8.1 Gradient Descent Algorithm
		5.4.8.2 Learning Rate
		5.4.8.3 Dropout Rate
		5.4.8.4 Seed
	5.4.9	Weights
		5.4.9.1 Balance Objective
		5.4.9.2 Objective Weights
		5.4.9.3 Weight field
	5410	Sampling Options
	5.4.10	5.4.10.1 Rate
		5.4.10.2 Range
		5.4.10.3 Sampling
		5.4.10.4 Replacement
	E / 11	5.4.10.5 Out of bag
		Creating Deepnets with Configured Options
		API Request Preview
5.5		zing Deepnets
	5.5.1	Partial Dependence Plot
	5.5.2	Image Deepnet Page
		5.5.2.1 Image Deepnet Page - Classification
		5.5.2.2 Image Deepnet Page - Regression
	5.5.3	Summary Report
		5.5.3.1 Field Importances
		5.5.3.2 Summary
5.6		et Predictions
	5.6.1	Introduction
	5.6.2	5 1
		5.6.2.1 Predict
		5.6.2.2 Deepnet Prediction with Images
		5.6.2.3 Batch Predictions
		5.6.2.3.1 Batch Predictions with Images
	5.6.3	Configuring Deepnet Predictions
		5.6.3.1 Probability threshold
		5.6.3.2 Default Numeric Value
		5.6.3.3 Fields Mapping
		5.6.3.4 Output Settings
	5.6.4	
		5.6.4.1 Single Predictions
		5.6.4.1.1 Prediction explanation
		5.6.4.2 Batch Prediction
		5.6.4.2.1 Output CSV file
		5.6.4.2.2 Output Dataset
	5.6.5	
	5.5.5	5.6.5.1 Using Deepnet Predictions via the BigML API
		5.6.5.2 Using Deepnet Predictions via BigML Bindings
	5.6.6	Descriptive Information
	5.0.0	5.6.6.1 Name
		5.6.6.2 Description
		5.6.6.3 Category

 5.6.7 Deepnet Predictions Privacy 5.6.8 Moving Deepnet Predictions to Another Project 5.6.9 Stopping Deepnet Predictions 5.10 Deleting Deepnet Predictions 5.7 Consuming Deepnet Nate BigML API 5.7.3 Using Deepnets Via the BigML Bindings 5.8 Deepnet Imits 5.8.0.1 Field Limits 5.8.1 PDP Limits 5.8.1 PDP Limits 5.9 Descriptive Information 5.9.1 Deepnet Category 5.9.1 Deepnet Category 5.9.2 Deepnet Category 5.9.3 Deepnet Conters 5.10 Deepnet Conters 5.11 Deepnet Name 5.12 Stopping Deepnets 5.13 Delepnet Conters 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.1 Fusion Stopping 6.3.1 Moving Deepnets 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.2 Objective Field 6.4.3 Weights 6.5.4 Truston Aptions 6.5.5 A Deepnet Conters 6.5.6 Limons 6.6.1 Introduction 6.6.2 Fusion Stopping Deepnets 6.5 Stopping Deepnets 6.5 Stopping Deepnets 6.6.4 Creating Fusions 6.6.1 Introduction 6.6.2 Fusion Configuration Options 6.6.3 Creating Fusions 6.5.1 Fusion Reprivation Options 6.6.1 Introduction 6.6.2 Preview 6.5 Visualizing Fusions with Configured Options 6.6.3 Predictions 6.6.1 Introduction 6.6.2.1 Predictions 6.6.3.2 Probability threshold 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.4 Excluded Fiels 6.6.3.4 Excluded Fiels 6.6.3.5 Helds Mapping 6.6.3.6 Output Settings 6.6.3.1 Introduction 6.6.3.2 Batch Predictions 6.6.3.2 Floation medictions 6.6.3.2 Floation medictions 6.6.3.4 Excluded Fiels 6.6.3.4 Excluded Fiels 6.6.3.4 Excluded Fie			
5.6.8 Moning Deepnet Predictions 5.6.10 Deleting Deepnet Predictions 5.7 Consuming Deepnet 5.7.1 Downloading Deepnet 5.7.2 Using Deepnets Via the BigML API 5.7.3 Using Deepnets Via the BigML API 5.7.4 Using Deepnets Via the BigML API 5.7.5 Descriptive Information 5.8.1 Deepnet Limits 5.8.1 Deepnet Description 5.9.2 Deepnet Description 5.9.3 Deepnet Description 5.9.4 Deepnet Counters 5.10 Deepnet So Another Project 5.13 Deleting Deepnets 5.14 Deepnet So Another Project 5.15 Deleting Deepnets 5.16 Deepnet Field 6.2 Understanding Fusions 6.2.1 Introduction 6.2 Understanding Fusions 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.4 Creating Fusions 6.5.1 Fusion Configuration Options 6.4.2 Colective Field			5.6.6.4 Tags
5.6.9 Stopping Deepnet Predictions 5.7 Consuming Deepnet Predictions 5.7.1 Downloading Deepnet 5.7.2 Using Deepnets Via the BigML API 5.7.3 Using Deepnets Via the BigML Bindings 5.8.0.1 Field Limits 5.8.0.1 Field Limits 5.8.1 PDP Limits 5.8.1 PDP Limits 5.9 Descriptive Information 5.9.1 Deepnet Pascription 5.9.2 Deepnet Description 5.9.3 Deepnet Tags 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 5.15 Deepnet Privacy 5.14 Takeaways 5.15 Deepnet Privacy 5.16 Lintroduction 6.2 Understanding Fusions 6.3 Catting Fusions 6.4 Lintroduction 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights <td></td> <td></td> <td>5.6.7 Deepnet Predictions Privacy 41</td>			5.6.7 Deepnet Predictions Privacy 41
5.6.9 Stopping Deepnet Predictions 5.7 Consuming Deepnet Predictions 5.7.1 Downloading Deepnet 5.7.2 Using Deepnets Via the BigML API 5.7.3 Using Deepnets Via the BigML Bindings 5.8.0.1 Field Limits 5.8.0.1 Field Limits 5.8.1 PDP Limits 5.8.1 PDP Limits 5.9 Descriptive Information 5.9.1 Deepnet Pascription 5.9.2 Deepnet Description 5.9.3 Deepnet Tags 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 5.15 Deepnet Privacy 5.14 Takeaways 5.15 Deepnet Privacy 5.16 Lintroduction 6.2 Understanding Fusions 6.3 Catting Fusions 6.4 Lintroduction 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights <td></td> <td></td> <td>5.6.8 Moving Deepnet Predictions to Another Project</td>			5.6.8 Moving Deepnet Predictions to Another Project
5.6.10 Deleting Deepnet Predictions 5.7.1 Downloading Deepnet Via the BigML API 5.7.2 Using Deepnets Via the BigML Bindings 5.8 Deepnet Limits 5.8 Deepnet Via the BigML Bindings 5.8 Deepnet Umits 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Name 5.9.3 Deepnet Counters 5.9.4 Deepnet Counters 5.9.5 Deepnet Paracy 5.10 Deepnet Vouters 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 6.1 Introduction 6.2 Fusions 6.3 Creating Fusions 6.4.1 Models 6.4.2 Cbjective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.6.2 Predictions 6.6.3 Configuration Options 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.6.2 Predictions 6.6.2 Predictions			
 5.7 Consuming Deepnet 5.7.1 Downloading Deepnet Name 5.7.2 Using Deepnets Via the BigML API. 5.7.3 Using Deepnets Via the BigML Bindings 5.8 Deepnet Limits 5.8.0 Field Limits 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Description 5.9.3 Deepnet Category 5.9.4 Deepnet Tags 5.9.5 Deepnet Counters 5.10 Deepnets Vanther Project 5.12 Stopping Deepnets 5.13 Deleting to Another Project 5.14 Takeaways 5.14 Takeaways 5.15 Through Tags 5.15 Deepnet Counters 5.10 Deepnet Solution 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.1 Models 6.5.2 Fusion Model List 6.5.2 Fusion Partial Dependence Plot 6.5.2 Fusion Predictions 6.6.3 Creating Fusions Predictions 6.6.3 Creating Fusion Predictions 6.6.3 Creating Fusion Predictions 6.6.3 Creating Fusion Predictions 6.6.4.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.3 Creating Fusion Predictions 6.6.3 Definition Predictions 6.6.3 Creating Fusion Predictions 6.6.3 Creating Fusion Predictions 6.6.3 Definition Predictions 6.6.3 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Disein Predictions 6.6.4.1 Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Couptur Dataset 			
5.7.1 Downloading Deepnets Via the BigML API. 5.7.3 Using Deepnets Via the BigML Bindings 5.8 Deepnet Limits 5.8.1 PDP Limits 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Name 5.9.3 Deepnet Name 5.9.4 Deepnet Regory 5.9.5 Deepnet Counters 5.10 Deepnet Counters 5.11 Moving Deepnets 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Fuderstanding Fusions 6.2.1 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions with Configured Options 6.4.4 Creating Fusion Predictions		- -	
5.7.2 Using Deepnets Via the BigML API. 5.7.3 Using Deepnets Via the BigML Bindings 5.8 Deepnet Limits 5.9 Descriptive Information 5.9 Descriptive Information 5.9 Descriptive Information 5.9 Deepnet Name 5.9.1 Deepnet Description 5.9.3 Deepnet Description 5.9.4 Deepnet Description 5.9.5 Deepnet Counters 5.10 Deepnet Triags 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.2 Fusion Objective Field 6.2.2 Fusion Objective Field 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5.1 Fusion Predictions 6.6.2 Freating Fusion Predictions		5.7	
5.7.3 Using Deepnets Via the BigML Bindings 5.8 Deepnet Limits 5.8.1 PDP Limits 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Description 5.9.3 Deepnet Category 5.9.4 Deepnet Tags 5.9.5 Deepnet Category 5.10 Deepnet Tags 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deepnet Privacy 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Fusion Objective Field 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.4 Creating Fusions with Configured Options 6.4.4 Creating Fusions 6.5.1 Fusion Predictions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Predictions 6.6.3			
5.8 Deepnet Limits 5.9 Descriptive Information 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Category 5.9.3 Deepnet Category 5.9.4 Deepnet Counters 5.10 Deepnet Tags 5.15 Deepnet Sto Another Project 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Inderstanding Fusions 6.3.1 Trusion Configuration Options 6.4.1 Models 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions 6.5.4 Invision Swith Configured Options 6.4.1 Models 6.5.2 Fusion Preview 6.5.1 Fusion Preview 6.5.1 Fusion Predictions <td< td=""><td></td><td></td><td>5.7.2 Using Deepnets Via the BigML API</td></td<>			5.7.2 Using Deepnets Via the BigML API
5.8 Deepnet Limits 5.9 Descriptive Information 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Category 5.9.3 Deepnet Category 5.9.4 Deepnet Counters 5.10 Deepnet Tags 5.15 Deepnet Sto Another Project 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Inderstanding Fusions 6.3.1 Trusion Configuration Options 6.4.1 Models 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions 6.5.4 Invision Swith Configured Options 6.4.1 Models 6.5.2 Fusion Preview 6.5.1 Fusion Preview 6.5.1 Fusion Predictions <td< td=""><td></td><td></td><td>5.7.3 Using Deepnets Via the BigML Bindings</td></td<>			5.7.3 Using Deepnets Via the BigML Bindings
5.8.1 Field Limits 5.9.1 Deepnet Information 5.9.2 Deepnet Name 5.9.3 Deepnet Category 5.9.4 Deepnet Counters 5.9.5 Deepnet Counters 5.10 Deepnet Counters 5.10 Deepnet Counters 5.10 Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.2 Fusion Objective Field 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 Weights 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Partial Dependence Plot 6.5.1 Fusion Partial Dependence Plot 6.5.2 Eusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Predictions 6.6.3 Defiguri		58	
5.8.1 PDP Limits 5.9.1 Deepnet Name 5.9.2 Deepnet Description 5.9.3 Deepnet Category 5.9.4 Deepnet Category 5.9.5 Deepnet Counters 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.2 Fusion Objective Field 6.2.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions 6.5 Visualizing Fusions 6.6.1 Introduction 6.6.2 <t< td=""><td></td><td>0.0</td><td></td></t<>		0.0	
 5.9 Descriptive Information 5.9.1 Deepnet Name 5.9.2 Deepnet Description 5.9.3 Deepnet Category 5.9.4 Deepnet Tags 5.9.5 Deepnet Counters 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.1 Fusion Collective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.6.1 Introduction 6.6.2 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2 Creating Fusion Predictions 6.6.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.3 Excluded Fiels 6.6.3.6 Contigured Constanting Constantions 6.6.3.1 Missing Strategies 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.4 Excluded Fiels 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Predictions 6.6.4.1 Predictions 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Organity Predictions 6.6.4.1 Predictions 6.6.4.2 Cuptur Dataset 			
5.9.1 Deepnet Name 5.9.2 Deepnet Description 5.9.3 Deepnet Category 5.9.4 Deepnet Tags 5.9.5 Deepnet Counters 5.10 Deepnet Yracy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.1 Fusion Objective Field 6.2.2 Fusion S 6.4.1 Models 6.4.2 Coljective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2.1 Predict 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value <td></td> <td></td> <td></td>			
5.9.2 Deepnet Description 5.9.3 Deepnet Tags 5.9.5 Deepnet Toyacy 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.3.1 Greating Fusions 6.4.1 Models 6.2.2 Fusion Dejective Field 6.2.2 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5.4 Prosendence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.2 Probability threshold 6.6.3.2 Probability threshold 6.6.3.4 Exoluded Fiels <		5.9	Descriptive Information
5.9.2 Deepnet Description 5.9.3 Deepnet Tags 5.9.5 Deepnet Toyacy 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.3.1 Greating Fusions 6.4.1 Models 6.2.2 Fusion Dejective Field 6.2.2 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5.4 Prosendence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.2 Probability threshold 6.6.3.2 Probability threshold 6.6.3.4 Exoluded Fiels <			5.9.1 Deepnet Name
5.9.3 Deepnet Category 5.9.4 Deepnet Tags 5.9.5 Deepnet Counters 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2.1 Fusion Objective Field 6.2.2 Fusion Objective Field 6.2.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 Weights 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.3 Definiting fusion Predictions 6.6.3 Probability threshold 6.6.3 Definiting Pusion Predictions 6.6.3.4 Evoluced Fiels 6.6.3.5			
5.9.4 Deepnet Tags 5.9.5 Deepnet Counters 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Default Numeric Value 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6			
5.9.5 Deepnet Counters 5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Objective Field 6.3. Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Partial Dependence Plot 6.5.3 Creating Fusion Predictions 6.6.4 Introduction 6.6.2 Predictions 6.6.3.1 Introductions 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value <			
5.10 Deepnet Privacy 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 6.6.1 Introduction 6.6.2 Creating Fusions Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions			
 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.4 Creating Fusions 6.5 Visualizing Fusions 6.5 Visualizing Fusions 6.6 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2 Detauth Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Muser Value 6.6.4.2 Batch Predictions 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.2 Batch Predictions 6.6.4.1 Misualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Output CSV file 6.6.4.2.2 Output Dataset 			
 5.11 Moving Deepnets to Another Project 5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.4 Creating Fusions 6.5 Visualizing Fusions 6.5 Visualizing Fusions 6.6 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2 Detauth Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Muser Value 6.6.4.2 Batch Predictions 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.2 Batch Predictions 6.6.4.1 Misualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Output CSV file 6.6.4.2.2 Output Dataset 		5.10	Deepnet Privacy
5.12 Stopping Deepnets 5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 6.6 Fusion Predictions 6.6.2 Creating Fusions Predictions 6.6.2.1 Predict 6.6.2 Creating Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Predictions 6.6.4.1 Single Predictions 6.6.4.1 Output CS			
5.13 Deleting Deepnets 5.14 Takeaways 6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Rediction explanation 6.6.4.1 Single Predictions 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions			
5.14 Takeaways 6 Fusions 6.1 6.2 Understanding Fusions 6.2.1 6.2.2 Fusion Objective Field 6.2.3 Greating Fusions 6.4 Fusion Configuration Options 6.4.1 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.3 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 1 6.5.1 Fusion Partial Dependence Plot 6.5.2 6.5.3 Fusion Predictions 6.6.4 6.6.2 Creating Fusion Predictions 6.6.3 6.6.4 1 Mitodution 6.6.2 Creating Fusion Predictions 6.6.3.1 6.6.3.2 Probability threshold			
6 Fusions 6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Field Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 <			
6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.1 Predictions 6.6.2.1 Predictions 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Field Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1		5.14	Takeaways
6.1 Introduction 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.1 Predictions 6.6.2.1 Predictions 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Field Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1	~	<u> </u>	
 6.2 Understanding Fusions 6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Directions 6.6.4.1 Directions 6.6.4.1 Directions 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Directions 6.6.4.2 Dutput CSV file 6.6.4.2.2 Output Dataset 	6		
6.2.1 Fusion Objective Field 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 API Request Preview 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Output CSV file 6.6.4.2.1 Output CSV file		••••	
 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2 Creating Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.2 Probability threshold 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Ningle Predictions 6.6.4.1 Predictions 6.6.4.1 Predictions 6.6.4.1 Output CSV file 6.6.4.2.2 Output Dataset 		6.2	Understanding Fusions
 6.2.2 Fusion Field Importances 6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2 Creating Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.2 Probability threshold 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Ningle Predictions 6.6.4.1 Predictions 6.6.4.1 Predictions 6.6.4.1 Output CSV file 6.6.4.2.2 Output Dataset 			6.2.1 Fusion Objective Field
6.3 Creating Fusions 6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Prediction s 6.6.4.2 Batch Predictions 6.6.4.1 Prediction s 6.6.4.2 Output CSV file 6.6.4.2.2 Output Dataset			
6.4 Fusion Configuration Options 6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 API Request Preview 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.2 Batch Prediction 6.6.4.2.2 Output CSV file		63	
6.4.1 Models 6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Partial Dependence Plot 6.5.2 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2 Batch Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Prediction <			
6.4.2 Objective Field 6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset		0.4	
6.4.3 Weights 6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.2.3 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.1 Predictions 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.4.4 Creating Fusions with Configured Options 6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.1 Introduction 6.6.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.1.1 Predictions 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2 Creating Fusion Predictions 6.6.2 Batch Predict 6.6.3 Configuring Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Probability threshold 6.6.3 Default Numeric Value 6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Output CSV file 6.6.4.2.2 Output Dataset			6.4.3 Weights
6.4.5 API Request Preview 6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2 Creating Fusion Predictions 6.6.2 Batch Predict 6.6.3 Configuring Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Probability threshold 6.6.3 Default Numeric Value 6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Predictions 6.6.4.1 Output CSV file 6.6.4.2.2 Output Dataset			6.4.4 Creating Fusions with Configured Options
6.5 Visualizing Fusions 6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Predict 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.5.1 Fusion Partial Dependence Plot 6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3.1 Predict 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.1 Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Predictions		65	
6.5.2 Fusion Model List 6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Predictions 6.6.4.2 Batch Predictions 6.6.4.2 Batch Prediction 6.6.4.2 Dutput CSV file 6.6.4.2.2 Output Dataset		0.0	
6.6 Fusion Predictions 6.6.1 Introduction 6.6.2 Creating Fusion Predictions 6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
 6.6.1 Introduction			
 6.6.2 Creating Fusion Predictions		6.6	
6.6.2.1 Predict 6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3 Missing Strategies 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			6.6.1 Introduction
6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			6.6.2 Creating Fusion Predictions
6.6.2.2 Batch Predictions 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			•
 6.6.3 Configuring Fusion Predictions 6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset 			
6.6.3.1 Missing Strategies 6.6.3.2 Probability threshold 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4.1 Single Predictions 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.3.2 Probability threshold . 6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels . 6.6.3.5 Fields Mapping . 6.6.3.6 Output Settings . 6.6.4 Visualizing Fusion Predictions . 6.6.4.1 Single Predictions . 6.6.4.2 Batch Prediction . 6.6.4.2.1 Output CSV file . 6.6.4.2.2 Output Dataset .			
6.6.3.3 Default Numeric Value 6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.3.4 Excluded Fiels 6.6.3.5 Fields Mapping 6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.3.5 Fields Mapping . 6.6.3.6 Output Settings . 6.6.4 Visualizing Fusion Predictions . 6.6.4.1 Single Predictions . 6.6.4.1.1 Prediction explanation . 6.6.4.2 Batch Prediction . 6.6.4.2.1 Output CSV file . 6.6.4.2.2 Output Dataset .			6.6.3.3 Default Numeric Value
6.6.3.5 Fields Mapping . 6.6.3.6 Output Settings . 6.6.4 Visualizing Fusion Predictions . 6.6.4.1 Single Predictions . 6.6.4.1.1 Prediction explanation . 6.6.4.2 Batch Prediction . 6.6.4.2.1 Output CSV file . 6.6.4.2.2 Output Dataset .			6.6.3.4 Excluded Fiels
6.6.3.6 Output Settings 6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			6.6.3.5 Fields Mapping
6.6.4 Visualizing Fusion Predictions 6.6.4.1 Single Predictions 6.6.4.1.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.4.1 Single Predictions 6.6.4.1.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.4.1.1 Prediction explanation 6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			0
6.6.4.2 Batch Prediction 6.6.4.2.1 Output CSV file 6.6.4.2.2 Output Dataset			
6.6.4.2.1Output CSV file6.6.4.2.2Output Dataset			
6.6.4.2.2 Output Dataset			
6.6.4.2.2 Output Dataset			6.6.4.2.1 Output CSV file

			6.6.5.1 L	sing Fusion Predictions via the Big	ML API	475
			6.6.5.2 L	sing Fusion Predictions via BigML F	Bindings	475
		6.6.6			· · · · · · · · · · · · · · · · · · ·	
		0.0.0				
			6.6.6.3 C	ategory		477
		6.6.7		-		
		6.6.8				
		6.6.9				
		6.6.10	Deleting F	sion Predictions		479
	6.7					
	••••	6.7.1				
				-		
		6.7.2				
		6.7.3	Using Fusi	Ins Via the BigML Bindings		481
	6.8	Fusior	Limits			482
	6.9	Descri	ptive Inform	ation		483
	0.0	6.9.1				
		6.9.2				
		6.9.3				
		6.9.4	Tags			486
		6.9.5				
	6 10					
	6.12	Stoppi	ng Fusions			488
	6.13	Deletir	na Fusions			490
			•			
	0.11	ranou	ayo			102
7	Eval	luation	c			194
	7.1	Introdu	uction			494
	7.1	Introdu	uction			494
	7.1	Introdu	uction standing Ev	luations		494 497
	7.1	Introdu Under	uction standing Ev Classificat	Iluations	· · · · · · · · · · · · · · · · · · ·	494 497 498
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 P	Iluations on Measures	· · · · · · · · · · · · · · · · · · ·	494 497 498 498
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C	Iluations	· · · · · · · · · · · · · · · · · · ·	494 497 498 498 498 498
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C	Iluations		494 497 498 498 498 498 499
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C	Iluations	· · · · · · · · · · · · · · · · · · ·	494 497 498 498 498 498 499
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C	Iluations		494 497 498 498 498 498 499 500
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.1 7.2.1.3.2	aluations		494 497 498 498 498 498 499 500 500
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.3	aluations		494 497 498 498 498 498 499 500 500 500
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4	aluations		494 497 498 498 498 499 500 500 500 500
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4 7.2.1.3.5	aluations		494 497 498 498 498 499 500 500 500 500 501 501
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4	aluations		494 497 498 498 498 499 500 500 500 500 501 501 502
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4 7.2.1.3.5	aluations		494 497 498 498 498 499 500 500 500 500 501 501 502
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C 7.2.1.3 C	aluations		494 497 498 498 498 498 499 500 500 500 500 501 501 502 502
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C	aluations		494 497 498 498 498 498 499 500 500 500 500 501 501 502 502 502 503
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C	aluations	resholds	494 497 498 498 498 499 500 500 500 500 501 501 502 502 503 504
	7.1	Introdu Under	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4 7.2.1.3.5 7.2.1.3.5 7.2.1.3.5 7.2.1.3 E	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 501 501 502 502 502 503 504 508
	7.1	Introdu Under	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 501 501 502 502 502 502 504 508 508
	7.1	Introdu Under	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.2 7.2.1.3.4 7.2.1.3.5 7.2.1.3.5 7.2.1.3.5 7.2.1.3 E	aluations	resholds	494 497 498 498 498 498 500 500 500 500 501 501 502 502 502 502 504 508 508
	7.1	Introdu Under	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 E 7.2.1.5 C	aluations	resholds	494 497 498 498 498 499 500 500 500 500 500 501 501 502 502 502 502 503 504 508 509 510
	7.1	Introdu Under	Letion standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C	aluations	resholds	494 497 498 498 499 500 500 500 500 500 501 502 502 502 502 502 502 503 504 508 509 510 513
	7.1	Introdu Under 7.2.1	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 E 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C	aluations on Measures ositive and Negative Classes onfusion Matrix assification Measures assification Measures Accuracy Precision Recall F-measure Phi Coefficient Macro-averages Kendall's Tau Spearman's Rho onfidence, Probability and Vote Thr valuation curves Precision-Recall Curve ROC Curve Gain Curve & K-S statistic Lift Curve	resholds	494 497 498 498 498 499 500 500 500 500 500 501 502 502 502 503 504 508 509 510 513 514
	7.1	Introdu Under	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 501 502 502 502 502 502 502 502 502 502 503 504 508 509 510 513 514 5515
	7.1	Introdu Under 7.2.1	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 500 500 502 502
	7.1	Introdu Under 7.2.1	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 500 500 502 502
	7.1	Introdu Under 7.2.1	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.2.5 C 7.2.5 C 7.2.5 C 7.2.5 C 7.5	aluations	resholds	494 497 498 498 498 500 500 500 500 500 500 500 500 500 50
	7.1	Introdu Under 7.2.1	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C 7.2.1.5 C 7.2.2.1 C 7.2.2.1 C 7.2.2.1 C 7.2.2.2 C 7.2.2.2 C 7.2.2.3 F	aluations	resholds	494 497 498 498 498 499 500 500 500 500 500 500 500 500 500 5
	7.1 7.2	Introdu Under 7.2.1	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 E 7.2.1.5 C 7.2.1.5 C 7.2.2.5 C 7.2.5 C 7.2.5 C 7.5 C 7	Aluations	resholds	494 497 498 498 498 499 500 500 500 500 500 500 500 500 500 5
	7.1	Introdu Under 7.2.1 7.2.2 7.2.2 7.2.3 Creatin	uction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.2.1 C 7.2.2.1 C 7.2.2.2 C 7.2.2.2 C 7.2.2.2 C 7.2.2.3 C Cross-Valin c Evaluation	aluations on Measures psitive and Negative Classes onfusion Matrix assification Measures Accuracy Precision Recall F-measure Phi Coefficient Macro-averages Kendall's Tau Spearman's Rho onfidence, Probability and Vote Thr valuation curves Precision-Recall Curve ROC Curve Gain Curve & K-S statistic Lift Curve Measures ean Absolute Error ean Absolute Error squared Absolute Error Squared Absolute Error Squared	resholds	494 497 498 498 498 499 500 500 500 500 500 500 500 500 500 5
	7.1 7.2	Introdu Under 7.2.1 7.2.2 7.2.2 7.2.3 Creatin 7.3.1	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.2.5 C 7.2.5 C 7.5 C 7.	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 500 500 500 50
	7.1 7.2	Introdu Under 7.2.1 7.2.2 7.2.2 7.2.3 Creatin	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.2.5 C 7.2.5 C 7.5 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 500 500 500 50
	7.1 7.2	Introdu Under 7.2.1 7.2.2 7.2.2 7.2.3 Creatin 7.3.1	Liction standing Ev Classificat 7.2.1.1 F 7.2.1.2 C 7.2.1.3 C 7.2.1.5 C 7.2.2.5 C 7.2.5 C 7.5 C	aluations	resholds	494 497 498 498 498 498 500 500 500 500 500 500 500 500 500 50

7.4.2Select the Voting Strategy for Decision Forests537.4.3Fields Mapping537.4.4Sampling Your Dataset547.4.4.1Rate547.4.4.2Range547.4.4.3Sampling547.4.4.4Replacement547.4.4.5Out of bag547.4.5Cross-Validation Configuration Options54	25 28 36 37 38 39 40 40 40 40 40 40 40 41 41
7.4.5.3 Ensemble's k-fold cross-validation 54 7.5 Visualizing Evaluations 54 7.5.1 Original Resources Information 54 7.5.2 Classification Evaluations 54 7.5.2.1 General Evaluation 54	43 44 45 46 46
7.5.2.2 Evaluation curves 55 7.5.3 Regression Evaluations 55 7.5.4 Cross-Validation Visualization 55 7.6 Evaluation Comparison 55 7.6.1 Compare Evaluations Side By Side 55 7.6.2 Compare Multiple Evaluations 55	51 52 54 54
7.6.2.1 Exporting Evaluation Comparison 56 7.7 Consuming Evaluations 56 7.7.1 Using Evaluations Via the BigML API 56 7.7.2 Using Evaluations Via the BigML Bindings 56	63 64 64 65
7.8Evaluation Limits567.9Descriptive Information567.9.1Evaluation Name567.9.2Evaluation Description567.9.3Evaluation Category567.9.4Evaluation Tags567.10Evaluation Privacy567.11Moving Evaluations to Another Project567.12Stopping Evaluation567.13Deleting Evaluation567.14Takeaways57	65 66 66 67 67 68 69 70
List of Figures 57	74
List of Tables 58	89
Glossary 59	90
References 59	94



Models

1.1 Introduction

There are multiple Machine Learning problems that require a model to predict an output variable (objective field in BigML parlance) given a number of input variables (input fields in BigML). These problems can be divided into classification and regression problems, depending on the data type of the objective field:

- **Classification**: when the objective field is categorical. For these problems, a Machine Learning algorithm is used to build a model that predicts a category (label or class) for a new example (instance). That is, it "classifies" new instances into a given set of categories (or discrete values). For example, "true or false", "fraud or not fraud", "high risk, low risk or medium risk", etc. There can be hundreds of different categories.
- **Regression**: when the objective field is numeric. For these problems, a Machine Learning algorithm is used to build a model that predicts a continuous value. That is, given the fields that define a new instance the model predicts a real number. For example, "the price of a house", "the number of units sold for a product", "the potential revenue of a lead", "the number of hours until next system failure", etc.

Both classification and regression problems can be solved using supervised Machine Learning techniques. They are called supervised in the sense that the values of the output variable has either been provided by a human expert (e.g., the patient had been diagnosed with diabetes or not) or by a deterministic automated process (e.g., customers who did not pay their fees in the last three months are labeled as "delinquent"). The objective field values along with the input fields need to be collected for each instance in a structured dataset that is used to train the model. The algorithms learn a predictive model that maps your input data to a predicted objective field value.

A BigML Model uses a proprietary decision tree algorithm based on the Classification and Regression Trees (CART) algorithm proposed by Leo Breiman. Section 1.2 explains in detail BigML models implementation and interpretation.

This chapter contains comprehensive description of BigML models including how they can be created with 1-click (Section 1.3), all configuration options available (Section 1.4), and the different visualizations provided by BigML (Section 1.5). Once you create a model, you can get a report for each field importance (Subsection 1.6.1). See Section 1.7 for an explanation of how models can be used to make predictions. You can also export your models in different formats to make local predictions faster at no cost (Section 1.8). The process to evaluate your model's predictive performance in BigML is explained in a different chapter (Chapter 7).

In BigML, the third tab of the main menu of your Dashboard allows you to list all your available models. In the model list view (Figure 1.2), you can see, for each model, the **dataset** it was created from, the model's **Name**, **Type** (either classification or regression), **Objective**, **Age** (time elapsed since it was created), **Size**, and number of **predictions**, **batch predictions**, or **evaluations** that have been created

Sources	Datasets Models Clusters	Anomalies	Associations	Predictions	Tasks			Scrip	NE •ts ▼
			Models						Q
II	Name	🗢 Туре	Objective	\$	🗄 🔶 🛛	a \$	\$	\$	
ıl <mark>e</mark>	Churn telecom April-2016 dataset - sample (8.		weight		5m 1w	8.0 KB			
6 . 2	Stroke Prediction	00	stroke		5m 1w	2.5 MB	1		
к <mark>Р</mark>	LinkedIn profiles	×	connections		5m 1w 5	51.9 MB			
к <mark>.</mark>	Predicting The Oscars - clone	۵۵	Best Picture_	0	5m 1w	77.9 KB			
K P	Children's interests	۵۵	Goals		5m 1w	23.6 KB			
ıl <mark>î</mark>	Churn Telecom dataset Training (80%)'s mode	el aO	churn		5m 1w 2	42.2 KB			1
ıl <mark>î</mark>	Loan risk - sample (80.00%)'s model	۵۵	class		5m 1w	92.0 KB			1
ılî	Air Carrier Flight Delays' dataset - sample (80.	×	Year		5m1w 8	30.1 MB			1
ılî	Accidents 2012 UK - sample (80.00%)'s model	۵۵	Did Police Off	icer Attend	5m1w 3	36.6 MB			1
ıl <mark>e</mark>	Churn telecom April-2016 dataset's model	×	weight		5m 1w	10.0 KB			
how 10	♦ models		21 to 30 of 74 model	S		K	《 1 2	3 4 5	> >

using that model. The SEARCH menu option in the top right corner of the ensemble list view allows you to search your models by name.

Figure 1.1: Models list view

When you first create an account at BigML, or every time that you start a new project, your list view for models will be empty. (See Figure 1.2.)

Sources	Datasets	Models -	Clusters	Anomalies	Associations	Predictions	Tasks	_	_	Scri	pts 👻
					Models						Q,
III N	ame			Type	Objective	\$	# \$	â :	\$	\$	○ 8 80 ≎
					No models						
Show 10	models				No models found					I< <	> >

Figure 1.2: Empty Dashboard models view

Finally, in Figure 1.3 you can see the icon used to represent a model in BigML.



Figure 1.3: Models icon

1.2 Understanding BigML Models

This section describes internal details about BigML models. Besides being useful to better understand how BigML implements them and the optimizations it applies to improve performance, it will also provide the foundations to understand BigML models' configuration options. You can safely skip this section on first read.

Models are built by splitting the data into partitions so each of them maximizes the information gain¹ for classification models or minimizes the mean squared error² for regression models. Each partition, i.e., split, has an associated predicate that defines it, e.g., balance < 1,000. The process of partitioning the instances in a dataset is recursive, i.e., partitions are themselves split into smaller ones, thus forming a hierarchy of partitions with their associated predicates. At each step of this process, a number of summary statistics is also computed, such as how many instances belong partition, its confidence, etc.

BigML uses a proprietary algorithm for models that produces Classification And Regression Tree³ (CART) style decision trees. BigML algorithm adapts itself to the dataset you are learning from to provide optimal performance. Indeed, for datasets that have a large number of instances, BigML will use "streaming", i.e., it will process the data instances in chunks to reduce the memory footprint it requires. Alternatively, for datasets that have a large number of fields, BigML tends to load the whole dataset into memory.

The key points of BigML streaming algorithm (stree) are listed below:

- Support for large datasets
- · Multi-core parallelism or multi-machine distribution
- Anytime algorithm⁴ with frequent updates

The key points of BigML in-memory algorithm (mtree) are listed below:

- Support for datasets with a large number of fields
- · Faster than stree, albeit at the cost of needing more memory
- Anytime algorithm⁵ with frequent updates

1.2.1 Streaming Model

Inspired by algorithms by Ben-Haim⁶ and Tyree, BigML models are grown breadth-first⁷. To grow the next generation of an stree, training instances are run through the tree and summary statistics are collected in each leaf. The summary statistics are then used to choose new splits for each leaf.

The main advantage of growing in this fashion is that an stree does not need to keep the training data in memory. Instead, the memory footprint for an stree is dependent on the size of the tree and the number of input features in the training data.

The summaries collected by an stree over a set of training instances may be merged with summaries from a separate block of training instances. This feature allows the stree summary collection phase to be easily parallelized or distributed.

Traditionally, CART trees consider every training instance that reaches a leaf when determining how to split. While an stree may be built similarly, it can also generate a split early using only a sample of the training data. stree attempts to detect when an early split is safe by calculating when the summary statistics have "settled". Early splitting requires that the training instances are shuffled beforehand. (See Subsection 1.4.8.)

¹https://en.wikipedia.org/wiki/Information_gain_in_decision_trees

²https://en.wikipedia.org/wiki/Mean_squared_error

³https://en.wikipedia.org/wiki/Decision_tree_learning#Types

⁴http://en.wikipedia.org/wiki/Anytime_algorithm

⁵http://en.wikipedia.org/wiki/Anytime_algorithm

⁶http://jmlr.csail.mit.edu/papers/v11/ben-haim10a.html

⁷https://en.wikipedia.org/wiki/Breadth-first_search

1.2.2 In-Memory Model

Unlike stree, which grows an entire new tree layer on every iteration (a breadth first⁸ expansion), mtree grows the tree one split at a time. At each iteration, an mtree chooses the split that looks to improve the tree the most. So the tree grows more like an A^{*} search⁹ than a breadth-first search.

mtree makes it possible to cap the overall size of BigML trees, so this approach helps prevent defining splits on relatively unimportant parts of the tree.

When choosing the best candidate split, mtree uses the reduction in squared error¹⁰ for regression trees. For classification trees, we use a split's information gain¹¹ multiplied by its population. These scores proved better than reusing pruning error estimates. (See also Subsection 1.4.3.)

As expected, this model growth technique outperforms the breadth-first approach for similarly sized trees.

With stree, the **node threshold** option (see Subsection 1.4.5) sets the minimum number of nodes in the tree, but that threshold could be exceeded. A threshold of 255 nodes could result in a tree sized anywhere between 255 and 509 nodes. However, asking for an mtree with 255 nodes will produce a tree with 255 nodes.

1.2.3 Scoring and Splitting

To choose a split for a leaf node, stree picks the best split for each input field and then selects the best of those candidates.

For categorical fields a candidate split is scored for every category, with a predicate in the form fieldX == "some_category". Along with the binary splits, stree scores an exploded split where every category is given its own child node.

Numeric splits are always binary with a predicate of the form fieldX >= 42. In traditional CART implementations, a split is considered between every two adjacent data points. Since stree does not keep the training data in memory, we use the summary histogram [11] instead. The median between every pair of adjacent histogram bins becomes a candidate split.

1.2.3.1 Early Splitting

A node in an stree can grow for one of two reasons. First, when the node collects summary information over the entire dataset it grows, as there is no more information to help inform the split. Secondly, a node can grow after collecting summary information over just a portion of the training data. This is the early split.

The core assumption for early splitting is that a sample of the dataset can often provide enough information to pick a good split. This is commonly the case for splits near the top of the tree. stree detects when enough information has been collected for an early split.

To find the similarity, stree calculates split scores for each field. The scores are normalized across all fields in the field summary set. At this point stree has two normalized score distributions, one for each field summary set. The L1 distance¹² between the distributions produces the similarity metric. A score of 0 means an exact match while a score of 1 means complete disagreement.

1.2.4 Pruning

As mentioned, models are trained by recursively partitioning the data, i.e., splitting it in two parts and then splitting each of them again in two and so on. This process goes on and on unless a stopping criteria is used. If a model grows too much, it ends up generating rules on minutiæ and losing the ability

⁸https://en.wikipedia.org/wiki/Breadth-first_search

⁹https://en.wikipedia.org/wiki/A*_search_algorithm

¹⁰https://en.wikipedia.org/wiki/Mean_squared_error

¹¹ https://en.wikipedia.org/wiki/Decision_tree_learning#Information_gain

¹²https://en.wikipedia.org/wiki/Taxicab_geometry

to generalize. This phenomenon is called overfitting. Pruning strategies play an essential role in avoiding it.

If pruning is enabled, then at each split BigML tries to determine whether that split increases the confidence (for classification models) or decreases the expected error (for regression models). If it does not, then it is pruned away. The pruning uses pessimistic error estimates as given by either the Wilson score interval or a standard confidence interval for regression trees.

Note: a node needs at least two instances to have a split, so models built on datasets with less than 200 instances will not be pruned.

1.2.5 Field Importance

The field importance is the relative contribution of each field to the objective field predictions. So, a field with higher importance will have a greater impact on predictions. You can find a visual histogram containing the importance for all fields in the Model Summary Report. (See Subsection 1.6.1.)

BigML calculates the field importances by estimating the prediction error each field helps to reduce for each split in a tree (these same estimates are used when pruning). Then, BigML sums the error reduction for every split grouped by field. Then, those sums are normalized, so that they total to 1. Conceptually, BigML measure of importance is based on the popular Random Forest measure of Breiman's Gini importance [1], except that BigML does not use Gini as the error metric.

Fields with an importance of zero may still be correlated with the objective field. It just means the model preferred other fields when choosing splits.

It is worth noting that the field importance values from Random Decision Forests will often be more meaningful than they are from a single tree.

Note: The concept of field importance is also used in prediction explanation for single predictions (See Subsection 1.7.4.1.1). But they are calculated differently. A field can be very important for the model but insignificant for a given prediction.

1.2.6 Confidence and Probability

Classification models in BigML include two different measures of the model's certainty when predicting a class at a node: **confidences** and **probabilities**. Both metrics can take values between 0%, which means the prediction is no better than randomly selecting a class, and 100% which indicates absolute certainty. The main difference between both metrics is that the confidence penalizes more heavily a low number of instances of the predicted class at the node. The confidence is considered a pesimistic measure of the model certainty because it is usually lower than the probability unless the number of instances at the node is very high in which case the confidence and probability take similar values. A detailed explanation of each one can be found in the following subsections.

1.2.6.1 Confidence

The confidence is calculated in BigML using the Wilson score formula. You can find a detailed explanation of the **confidence formulation** and its **interpretation** below.

1.2.6.1.1 Formulation

BigML computes confidence based on the lower bound of the Wilson score interval¹³, which means it provides a pessimistic confidence:

$$\frac{\hat{p} + \frac{1}{2n}z^2 - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{1}{n}z^2}$$

¹³https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval#Wilson_score_interval

Where \hat{p} is the proportion of instances for the predicted class, n is the number of instances at that node and z = 1.96, calculated as the $\frac{1-\alpha}{2}$ quantile of the standard normal distribution for a level of error of $\alpha = 5\%$.

1.2.6.1.2 Interpretation

The main goal is to balance the proportion of the predicted class with the uncertainty of a small number of instances. This formula takes into account two factors:

- The **class distribution** of the node: assuming equal number of instances, the more homogeneous the node, the higher the confidence. In other words, the node containing more instances for the predicted class will have higher confidence.
- The **number of instances** in the node: a node containing 1,000 instances with the same class distribution vs. another one containing five instances deserves higher confidence as uncertainty about the predicted class is higher whenever fewer instances are involved.

For example, if we are trying to predict if a person will have diabetes, Figure 1.4 shows both example nodes have the same predicted class distribution (93.75%), but the node with 64 instances has more confidence than the node with 16 instances because of the adjustment the Wilson interval formula makes to penalize the lower number of instances.

The idea is to provide a confidence measure underestimating the performance of your model, so you can be sure that predictions will not do it worse than the confidence.

BigML shows the confidence in several places: in a model's top panel, at each node, in the prediction form (see Subsection 1.7.2.2), and in the BigML API.



Figure 1.4: Example of confidences in two different nodes of the same tree

1.2.6.2 Probability

The **probability** of a class at a certain node is the **percentage of instances for that class at the node**. BigML also adds Laplace smoothing also known as additive smoothing¹⁴ to this probability. This means the class probabilities will tend to be pulled towards to their overall proportion in the dataset. If there are only a few instances in the leaf leading to the prediction, this smoothing can have a significant impact. If there are a high number of instances, the smoothing will have little effect.

For the probability calculation BigML takes into account the total number of instances for the class of interest at a given node ("node_class_count"), the total number of instances in the node ("node_total_count"), and the percentage of instances for that class over the total instances in the dataset ("class_prior"):

 $Probability_class = \frac{node_class_count + class_prior}{node_total_count + 1}$

To **illustrate** the probability calculation **with an example**, imagine a dataset with a total of 100 instances and three classes: class_a (30 instances), class_b (50 instances), and class_c (20 instances).The "class_prior" for each of them is:

¹⁴https://en.wikipedia.org/wiki/Additive_smoothing

class_a_prior = 0.3 class_b_prior = 0.5 class_c_prior = 0.2

If we were calculating the probabilities at a given node containing a total of 20 instances where 10 instances belong to class_a and other 10 instances to class_c, the probabilities for each class will be calculated as:

probability_class_a = $\frac{10 + 0.3}{20 + 1} = 0.49$ probability_class_b = $\frac{0 + 0.5}{20 + 1} = 0.024$ probability_class_c = $\frac{10 + 0.2}{20 + 1} = 0.486$

1.2.7 Expected Error

The measure of the predictions certainty at a node for **regression trees** is the expected error.

1.2.7.1 Formulation

The formula below is the standard formula to estimate the average squared error for the prediction at a given node:

$$\bar{e} = \frac{\sum_{i} (v_i - x)^2}{n}$$

Here, x is the predicted output at the node, n is the total number of instances, and $v_1...v_n$ are the rest of the objective values at that node. Note that if we use n - 1 instead of n, this formula becomes the sample variance (S^2) of the node. To calculate the expected error for the predictions, we need to include the error that could arise from our sample variance not representing the true population variance. We solve this problem by constructing a confidence derived from the true variance, which we will call σ^2 .

$$\left[\frac{S^2(n-1)}{\chi^2_{\alpha/2,n-1}},\frac{S^2(n-1)}{\chi^2_{1-\alpha/2,n-1}}\right]$$

The notation $\chi^2_{p,f}$ indicates the critical value from the χ^2 , distribution at probability p with f degrees of freedom. Using the upper bound, we can calculate the standard error of our estimate of the mean. The estimated error using the bounds is then our upper bound on the standard deviation $\hat{\sigma}$, plus the possible error in the mean above, squared:

$$\hat{e} = (z\frac{\hat{\sigma}}{\sqrt{n}} + \hat{\sigma})^2 \tag{1.1}$$

$$=\frac{z^2\hat{\sigma}^2}{n} + \frac{2z\hat{\sigma}^2\sqrt{n}}{n} + \frac{\hat{\sigma}^2n}{n}$$
(1.2)

$$=\frac{\hat{\sigma}^{2}(z^{2}+2z\sqrt{n}+n)}{n}$$
(1.3)

$$=\frac{\hat{\sigma}^{2}(z+\sqrt{n})^{2}}{n}$$
 (1.4)

If the variance (and thus error \hat{e}_c) is zero, we use the parent error \hat{e}_p to construct the following estimate of the child error given the parent prior $\hat{e}_{c|p}$:

$$\hat{e}_{c|p} = \frac{\hat{e}_p + n\hat{e}_c}{n+1} = \frac{\hat{e}_p}{n+1}$$

1.2.7.2 Interpretation

The expected error is an average of the instance errors at a given node. In Figure 1.5, the node predicts a value of 70.60 with an error of 28.59, meaning the prediction is, on average, likely to be within 28.59 of the target (between 42.01 and 99.19 in this example). This is an average, so while a single prediction may be more than 28.59 away from the true target, on average it is expected to do better than that. As in the confidence calculations for classification models, it is a pessimistic measure of your model performance, so that you can be sure your predictions will not do worse than the predicted value +/- the expected error.



Figure 1.5: Example of the expected error at a node

Along with the prediction and the expected error at a node, BigML also provides the histogram with the instances and their values. This allows you to find out the predicted value distributions and gives you information such as the complete possible range for the objective variable or whether the error is more likely to be skewed to one side of the prediction. If you need to further explore the data for a specific branch, you can also create a dataset from that branch (see Figure 1.26).

1.2.8 Models with Images

BigML models do not take images as input directly, however, they can use image features as those fields are numeric.

BigML extracts image features at the source level. Image features are sets of numeric fields for each image. They can capture parts or patterns of an image, such as edges, colors and textures. For information about the image features, please refer to section Image Analysis of the Sources with the BigML Dashboard¹⁵[11].

¹⁵https://static.bigml.com/pdf/BigML_Sources.pdf

Sources Datase	ts Supervised ▼ Unsup	ervised Predictions	Tasks	W	hizzML 👻
	••••	grape-strawb	erry	ılề₊ 📚 - 🦽 (≡)*- 🛈
	• label	ABC × V	E	Search by name	×
Name	1, Туре	\uparrow_{\downarrow} Count \uparrow_{\downarrow} Miss	ing t _i Errors	1 Histogram	
image_id	image	70	0	۵ 🍝 🎑 🐛 🔍	
filename	path	70	0	0	
label	AB C 234 image features	70	0	0	
Show 10 V fields	✓! 236 234	1 to 3 of 3 fields		I< <	1 > >

Figure 1.6: A dataset with images and image features

As shown in Figure 1.6, the example dataset has an image field *image_id*. It also has image features extracted from the images referenced by *image_id*. Image feature fields are hidden by default to reduce clutter. To show them, click on the icon "Click to show image features", which is next to the "Search by name" box. In Figure 1.7, the example dataset has 234 image feature fields, called *Histogram of gradients*.

Sources Datasets Supe	ervised 🗸 Unsuper	vised - Predictions -	Tasks		WhizzML 🔫
8 ¥. 8		grape-strawber	Click to hide ima	age features	(≡) ⁵ , (i
	əl	ABC × v			×
Name	1, Туре	t↓ Count t↓ Missing	t. Errors	t Histogram	
image_id	image	70	0	• 👅 🍓 🍯 🔤	*
ïlename	path	70	0	0	
abel	ABC	70	0	0	
mage_id.full image: 0° edge stre	🕞 gradients	70	0	• I II	
mage_id.full image: 20° edge str	🕞 gradients	70	0	°	
mage_id.full image: 40° edge str	🕞 gradients	70	0	•	
mage_id.full image: 60° edge str	🕞 gradients	70	0	°	al
mage_id.full image: 80° edge str	🕞 gradients	70	0	°	
mage_id.full image: 100° edge st	🕞 gradients	70	0	•	dh
mage_id.full image: 120° edge st	🕞 gradients	70	0	°	
Show 10 🗸 fields 🖌 🖋 🛚 236	⊠≣ 234	1 to 10 of 237 fields		K < 1 2	3 4 5 > >

Figure 1.7: A dataset with image feature fields shown

From image datasets like this, models can be created and configured using the steps described in the following sections. All other operations including prediction, evaluation applies too.

1.3 Creating Models with 1-Click

To create a model in BigML you have two options: either the 1-click option which uses the default values for all available configuration options, or you can tune the parameters in advanced by using the configuration options explained in Section 1.4. You can also create a BigML model either from a dataset or from a cluster. Creating BigML models from clusters will be explained with more detail in the **Cluster Analysis with the BigML Dashboard document** [8]. This section will guide you through the process of creating BigML models from a dataset with just 1-click.

The BigML unique **1-click** feature provides a convenient way to quickly train a model from a dataset. The 1-CLICK MODEL menu option is available in the **1-click action menu**. (See Figure 1.8.)

	PRO	DJECT: All		- / 🖬 📽	PRO	D C De
Sources Datasets M	odels - Clusters	Anomalies As:	sociations Predi	ctions Tasks	_	Scripts 🐳
a s ^a 67,	•	iris	s.csv dataset		°a- ∽	(≡)∕∵ (j
<u>a</u>				A 1-CLICK MODEL		×
Name	🗢 Туре	Count	Missing	1-CLICK ENSEMBLE		
and the state	(122	150	0	1-CLICK CLUSTER		
sepal length	123	150	0	1-CLICK ANOMALY		
sepal width	123	150	0	1-CLICK ASSOCIATION		
ooparman		100	Ū	1-CLICK TRAINING TEST		

Figure 1.8: 1-click model

Alternatively, you can select the 1-CLICK MODEL option in the 1-click action menu from the dataset list view. (See Figure 1.9.)

			PROJECT: All			•	e 🛛 🖬	~ 2	Ϊ¢	PROD C	DEV
Source	es Datasets Mo	<u><u> </u></u>	1-CLICK MODEL		iations P	redictions	Tasks			Scr	ipts + 💖
			1-CLICK ENSEMBLE		Datasets						Q
9	Name		1-CLICK CLUSTER 1-CLICK ANOMALY		# \$	Å \$	£1 ≎	£ំណ ÷	¥. ¢	*****	₽ ≎
ılı	iris.csv dataset - sample	-	1-CLICK ASSOCIATION		1m 3w	922 bytes					
ılî	iris.csv dataset - sample	_	1-CLICK TRAINING TEST		1m 3w	3.6 KB					
ılı	iris.csv dataset - sample		VIEW DETAILS DELETE DATASET		1m 3w	922 bytes					
ılî	iris.csv dataset - sample			>	1m 3w	3.6 KB					
54	iris.csv dataset			۲	1m 3w	4.5 KB	1	0	0	0	0

Figure 1.9: 1-click model from the dataset list view

Either option builds a new model using default values for all available configuration options (see Section 1.4). This can give you a quick starting point to understand how your model behaves and how it can be improved.

Creating a model may take a variable time, depending on how big your dataset is, your subscription plan, etc. Once your BigML model is ready, it will be automatically displayed on your BigML Dashboard and you will be able to explore it through BigML sophisticated visualizations (see Section 1.5), and using it for evaluations (see Chapter 7) or predictions (see Section 1.7.)

1.4 Model Configuration Options

While the 1-click creation menu option (see Section 1.3) provides a convenient and easy way to create a BigML model from a dataset, there are cases when you want more control. This section will focus on

the options that BigML offers to configure its internal algorithms for BigML models.

You can set a number of parameters that affect the way BigML creates models from a dataset. Such parameters can be grouped in two categories:

- Parameters that are permanently associated to the dataset, such as its objective field and preferred fields. Once you provide a value for a dataset's permanent parameters, they will be used as a default value for the creation of models from that dataset.
- Parameters that only affect the model that is currently being created and that you are expected to set each time. Those include the objective field, included/excluded fields, and a number of configuration options that are described below.

Set a dataset's permanent parameters by clicking on the edit button that is displayed when you hover on the dataset's fields. This opens a modal dialog where you can set some of the field properties (See Figure 1.10).

	Sources Datasets Models +	Clusters Anomalies Associations Pr	edictions Task	s Scripts -
	≙ 5° 60° ••••	iris dataset		°‰- ∽ (≡)-
	,iff	Update field 000002 details		Q X
	Name		Errors	Histogram
	sepal length	Name	0	ndindindindina.aa
Edit field	sepal width	petal length Label	0	
	potal longth		0	J
Preferred	petal width	Description	0	
field	species		0	
Objective	Show 10 0 fields			
field -		Save Cancel		

Figure 1.10: Configure permanent parameter modal

Click on the preferred field button to make that field non-preferred.

Click on the objective field button to make that field the new objective field.

To access the configuration panel, select the CONFIGURE MODEL menu option located in the **configuration menu** of your dataset's detail view. (See Figure 1.11.)

Sources Datasets Model	s - Clusters Anomalies	Associations Predictions Tasks	Scripts 🗸
≜ 5° 6 0,		iris dataset	\$\$- \$\$- (≡)-
K		CONFIGURE MODEL	×
Name sepal length	Type Count 123 150		
sepal width	123 150	CONFIGURE ANOMALY	1111
petal length	123 150	CONFIGURE ASSOCIATION	ит
petal width	123 150	SAMPLE DATASET	llinna
species	ABC 150	add fields to dataset	
Show 10 ᅌ fields	100	1 to 5 of 5 fields	

Figure 1.11: Configure model

When the configuration panel is displayed, you can:

- Select or deselect individual fields for them to be included in or excluded from the model computation.
- · Change the objective field used for the model to be created.
- Manually configure a number of configuration options or automatically optimize these options.

Note: when the configuration panel is displayed, the edit is not visible, so you cannot set the dataset's permanent properties.

See below for a detailed explanation of the configuration options that are available as well as the corresponding default values.

1.4.1 Objective Field

Also known as "target field", the objective field is the output variable you want to predict.

Select your objective field in BigML in either of two ways. Specify the objective field each time you create a model from the configuration panel or set a field as the default objective for all models by clicking the edit button and then the objective field button.

By default, BigML will use the last valid field in your dataset as objective, with the exemption fields of type text and items that cannot be used as objective.

1.4.2 Automatic Optimization

You can turn on the **Automatic optimization** option so BigML will automatically tune the parameters of your model (see Figure 1.12).

Sources Datasets Supervised -	Unsupervised ▼ Predictions ▼ Tasks	WhizzML 👻
≙ ₽, 80, ····	Sentiment	🎭 ఁమ- (≡)⁵- ③
MODEL CONFIGURATION		100
Objective field: sentiment ABC v	Automatic optimization Max. training time: Mode 0:30:00	I candidates:
Advanced configuration		\odot
Model name:		
Sentiment	Reset	😚 Create model

Figure 1.12: Automatic optimization

The high number of possible combinations for parameter values makes it difficult to find the optimum configuration since the combinations that lead to a poor result outnumber the ones that result in a satisfying performance. Hand-tuning different configurations is a time-consuming process that requires a high level of expertise and intuition. To combat this problem, BigML offers first-class support for automatic model parameter optimization.

Behind the scenes, BigML uses the same technology for model parameter optimization as the one used for OptiML. If you want to know more about the technical details, please read the Chapter 2 of the document OptiML with the BigML Dashboard [10].

When you turn on the **Automatic optimization** option, all the model parameters will be disabled (because they will be automatically optimized), except the **Missing splits** and the **Weights** parameters which you can manually configure (see Subsection 1.4.4 and Subsection 1.4.6).

ources Datasets Supervised	 Unsupervised Predictions Insupervised I	· Tasks	WhizzMl
	Diabetes diagnosis_weight	_field Training (80%)	* ☆ ▲ <i>(</i> 5 - (≡) [*] ~
ODEL CONFIGURATION			
bjective field: Diabetes ABC	Automatic optimization	Max. training time: 00:30:00	Model candidates:
Advanced configuration			
Tree:		The parame	eters will be automatically optimized $igodot$
Statistical pruning:			STATISTICAL PRUNING 2 AUTO
Missing splits:			MISSING SPLITS NO
Node threshold:			
-0			512
Weights:			۲
Select a weight option Weight field			WEIGHTS No weights
Objective weights			

Figure 1.13: Configure the missing splits and the weights for your model

Since the optimization process can take some time, BigML offers two configurable parameters to limit the time to create the optimized model: a training duration (see Subsection 1.4.2.1) and the model candidates (see Subsection 1.4.2.2).

1.4.2.1 Training duration

The scale parameter to regulate the model runtime. It's set as an integer from 1 to 10. It indicates the user preference for the amount of time they wish the optimization to take. The higher the number, the more time that users are willing to wait for possibly better model performance. The lower the number, the faster that users wish the model training to finish. The default value is set to 5.

The training duration is set in a scale. The actual training time depends on the dataset size, among other factors.

Sources Datasets Supervised -	Unsupervised Predictions	Tasks	WhizzML 👻
	diabetes		lề- 🗞 (≡)⁵ 🛈
MODEL CONFIGURATION			Pot
Objective field: diabetes ABC v	Automatic optimization	Training duration:	Model candidates:
Advanced configuration			\odot
Model name:			.+
diabetes		Reset	📅 Create model

Figure 1.14: Training duration

1.4.2.2 Model candidates

The maximum number of different models (i.e., models using a unique configuration) to be trained and evaluated during the optimization process. The default is 128 models which is usually enough to find the best model, but you can set it from 4 up to 200. The top-performing model model will be returned. If the training duration is very low (see Subsection 1.4.2.1) given the dataset size, it is possible that not all the model candidates will be tried out.

Sources Datasets Supervised - I	Unsupervised - Predictions -	Tasks	WhizzML 🗸
A 17, 67,	diabetes		ılề₂ 🎭 🦽₂ (≡)⁵₂ (〕
MODEL CONFIGURATION			204
Objective field: diabetes ABC *	Automatic optimization	Training duration:	Model candidates:
Advanced configuration			$\overline{\mathbf{O}}$
Model name: diabetes		Reset	Greate model

Figure 1.15: Model candidates

1.4.3 Pruning

As explained in Subsection 1.2.4, pruning strategies are essential to avoid overfitting, a phenomenon that reduces a model's ability to generalize.

In BigML you can choose three different strategies for pruning (Figure 1.16):

- Smart Pruning: considers pruning the nodes with less than 1% of the instances.
- Statistical Pruning: considers every node for pruning.
- No Statistical Pruning: deactivates pruning altogether.

By default, BigML uses Smart Pruning to create your models.

1.4.4 Missing Splits

When training a model, BigML may encounter missing values, which can be either considered or ignored for the definition of splitting rules.

To include missing splits in your model, enable the **missing splits** option. (See Figure 1.16.) If missing values are included in your model, you may find rules with predicates of the following kind: field x = "is missing" or field x = "y or is missing".

BigML includes missing values following the MIA approach¹⁶ [14].

By default, BigML does not include missing splits.

1.4.5 Node Threshold

Set the **node threshold** to set a limit to a BigML model's growth. (See Figure 1.16.) A lower threashold simplifies the model while helping to avoid overfitting. However, it may also have reduce the model's predictive power compared to deeper models. The ideal number of nodes may depend on the dataset size and the number of features. Larger datasets with many important features may require more complex models. Reducing the number of nodes can also be useful to get an initial unerstanding of the basic data patterns. Then you can start growing the model from there.

By default, BigML sets a **512 node threshold**. Since nodes are computed in batches, on occasion the final number of nodes can be greater than the node threshold. (See Subsection 1.2.2.)

Objective field	Sources Datasets Ensembles + Clusters Anomales Associations Predictions Tasks Image: Comparison of the second s	Sorps 💭	Pruning - strategy
Missing	ବୁରୁ Configure	۲	
values strategy	Missing splits:	MISSING SPLITS NO	
Node threshold —	Node threshold:	NODE THRESHOLD	
	Weights:	⊚	
	Sampling:	768 instances 💿	
	Advanced sampling:	Default settings 📀	
	Advanced ordering:	Default settings 💿	



1.4.6 Weight Options

It is not unusual for a dataset to have an unbalanced objective field, where some categories are common and others very rare. For example, in datasets used to predict fraud, usually fraudulent transactions are very scarce compared to regular ones. When this happens, models tend to predict the most frequent values simply because the overall model's performance metric improves with that approach. However,

¹⁶http://oro.open.ac.uk/22531/1/decision_trees.pdf

in cases such as fraud prediction, you may be more interested in predicting rare values rather than successfully predicting frequent ones. In that case, you may want to assign more **weight** to the scarce instances so they are equivalent to the abundant ones.

BigML provides three different options to assign specific weight to your instances.

1.4.6.1 Balance Objective

When you set the **balance objective** weight, BigML automatically balances the classes of the objective field by assigning a higher weight to the less frequent classes, with the most frequent class always having a weight of 1. This option is only available for classification models. For example, take the following frequencies for each class:

[False, 2000; True, 50]

By enabling the **balance objective** option, BigML will automatically apply the following weights:

[False, 1; True, 40]

In this example, the class "True" is getting forty times more weight as it is forty times less frequent than the most abundant class.

1.4.6.2 Objective Weights

The **objective weights** option allows you to manually set a specific weight for each class of the objective field. BigML oversamples your weighted instances replicating them as many times as the weight stablishes. If you do not list a class, it is assumed to have a weight of 1. Weights of 0 are also valid. This option is only available for classification models.

This option can be combined with the **Weight field** (see Subsection 1.4.6.3). When combining it with the **Weight field**, both weights are multiplied. For example if you assign a weight of 3 for the "True" class and the weight field assigns a weight of 2 for a given instance labeled as "True", that instance will have a total weight of 6.

1.4.6.3 Weight Field

The **Weight Field** option allows you to assign individual weights to each instance by choosing a special weight field. It can be used for both regression and classification models. The selected field must be numeric and it must not contain any missing values. The weight field will be excluded from the input fields when building the model. You can select an existing field in your dataset or you may create a new one in order to assign customized weights.

For example, below is a dataset for which we included a field called "Weight" that assign a ten time higher weight to fraudulent transactions in comparison to non-fraudulent ones. BigML provides a powerful tool, the BigML Flatline editor, to add new fields to your dataset, such as a weight field. As an additional example, we could also take into account the transaction "Amount" to calculate the weights, so transactions with higher amounts will have higher weights.

Trans. ID	Products	Online	Amount \$	Fraud	Weight
xxxxxx098	XYZGH	yes	3,218	FALSE	1
xxxxxx345	VBHGF	no	1,200	FALSE	1
xxxxxx123	UYFHJ	yes	5,000	FALSE	1
xxxxxx567	HSNKI	no	390	FALSE	1
xxxxx789	SHSYA	yes	500	TRUE	10
xxxxxx093	DFSTU	yes	423	FALSE	1
xxxxxx012	TYISJ	yes	60,000	FALSE	1
xxxxxx342	SJSOP	no	789	FALSE	1
xxxxxx908	IOPKJ	no	9,450	FALSE	1
xxxxx334	HIOPN	yes	50,678	TRUE	10

Table Titt Holght Flora example for transactional databet	Table 1.1:	Weight Field	d example for	r transactional	dataset
---	------------	--------------	---------------	-----------------	---------



Figure 1.17: Weighting arguments for models

1.4.7 Sampling Options

Sometimes you do not need all the instances contained in your testing dataset to build your model. If you have a very large dataset, sampling may be a good way of getting faster results. (See Figure 1.18.)

The same sampling options described in the **Datasets with the BigML Dashboard document** [9] to sample datasets, are also available when building BigML models. They are divided in two groups: sampling and advanced sampling options.

1.4.7.1 Rate

The sampling **rate** is the frequency of instances being extracted from the dataset and included in your sample. A sampling rate of 100% means that all instances are included; a rate of 10% means 10% of the instances are included. This option may take any value between 0% and 100%. You can easily configure the **rate** by moving the slider in the **configuration panel for sampling**, or by typing the percentage in the tiny input box, both highlighted in Figure 1.18.

By default, BigML uses a 100% rate.

1.4.7.2 Range

The sampling **range** is the subset of the dataset instances from which to sample, e.g., from instance 5 to instance 1,000. The **rate** will be applied over the range configured.

By default, all instances are included, i.e., the range is (1, num. rows in dataset).

1.4.7.3 Sampling

The **sampling** option represents the type of the sampling process, which can be either random or deterministic.

When using deterministic sampling the random-number generator will always use the same seed, producing repeatable results.

By default, BigML uses random sampling.

1.4.7.4 Replacement

The **replacement** option controls whether a single instance can be selected multiple times or not. Sampling without replacement ensures that each instance cannot be selected more than once.

By default, BigML generates samples without replacement.

1.4.7.5 Out of Bag

The **out of bag** option allows you to include in your sample only those instances that were not selected in the first place, thus effectively inverting the sampling outcome. It is only selectable when a sample is deterministic and the sample rate is less than 100%. The total percentage of instances included in your sample will be one minus the **rate** (when replacement is not allowed). This can be useful for splitting a dataset into training and testing subsets.

By default, BigML will not use out of bag instances.



Figure 1.18: Sampling arguments for models

1.4.8 Advanced Ordering

Ordering options are relevant to ensure that BigML can correctly determine whether it can take an early split of your dataset to accelerate the training process. In particular, early splitting can only be safely used if the training instances have been previosuly shuffled. (See Subsection 1.2.3.1.)

If your instances are already shuffled, BigML allows you to choose the **linear** option. This will make the process of building the model much faster, since it will not required to reshuffle the dataset. If you need to shuffle your instances, BigML provides two options to that aim, **deterministic shuffling** and **random shuffling**, which are described below.

Ordering options have no influence on datasets of less than 34GB, since the whole dataset is used to build the model.

By default, BigML uses **deterministic shuffling** to ensure the same (deterministic) sample of the instances is used and the built model is thus repeatable.

1.4.8.1 Deterministic Shuffling

The **deterministic shuffling** option ensures that the row shuffling of a dataset is always the same, so that retraining a BigML model from the same dataset yields the same results.

By default, this option is true.

1.4.8.2 Linear Shuffling

The **linear shuffling** option is useful when you know that your instances are already in random order. Using linear shuffling, the BigML model will be constructed faster.

By default, this option is false.

1.4.8.3 Random Shuffling

The **random shuffling** option will ensure that a different shuffling will be tried each time you train your model.

By default, this option is false.

	Objective field:	Diabetes Diagnosis' da	80 Co- 🛈
	Objective field:		
	Diabetes	Statistical pruning:	
	ିର୍ଦ୍ଧୁ Configure		۲
	Missing splits:		MISSING SPLITS 0 NO
	Node threshold:		NODE THRESHOLD
	Weights:		⊚
	Sampling:		768 instances 🕤
	Advanced sampling:		Custom settings 💿
eterministic	Advanced ordering:		Custom settings
Linear —	Ordering:		ORDERING Deterministic

Figure 1.19: Ordering argument for models

1.4.9 Creating Models with Configured Options

After finishing the configuration of your options, you can change the default model name in the editable text box. Then you can click on the Create model button to create the new model, or reset the

ective field: ality ABC	Automatic optimization			
	_			
Advanced configuration				
Tree:				谢
Weights:				⊽
First split:				♥
Sampling:			7,388 instances	◙
Dataset advanced sampling:			Default settings	$\overline{\mathbf{O}}$
Advanced ordering:			Default settings	◙
del name:				
quality		Reset	សិ [‡] Create	model

configuration by clicking on the Reset button.

Figure 1.20: Create model after configuration

1.4.10 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to

the Reset button (See (Figure 1.20)). This is to show how to create the model programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 1.21) below:

ective field:	ABC V	
a Advanced c		×
Tree:	API request preview API endpoint: /model	E 🗣 💿
Weights:	1~ { 2	$\overline{\bigcirc}$
First split:	3 "node_threshold"; 512, 4 "name": "lc_quality",	\odot
Sampling:	<pre>5 "optimize": false, 6 "dataset": "dataset/5cb28499de2d4d0d760000e6",</pre>	s 💿
Dataset adv	7 - "excluded_fields": [8 "000000", 9 "000000",	s 💿
Advanced or	10 "000004", 11 "000006",	s 💿
	Check the full list of arguments in the AP	Idocumentation
del name:		

Figure 1.21: Model API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for models, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the chosen configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

1.5 Visualizing BigML Models

Being able to effectively visualize a model is paramount to exploring it, interpreting it, and explaining why it produces certain outcomes. To this aim, BigML provides two powerful interactive visualizations for BigML models, the compact tree and the sunburst views, which will be introduced below.

1.5.1 BigML Compact Tree Visualization

BigML compact tree is a powerful, patented, visual tree representation of your BigML model. Thanks to their being interactive, BigML compact trees bring to a new level your ability to explore your BigML models, by allowing you to see at a glance the most relevant information encoded in the model, by allowing you to collapse less important parts, and by enabling advanced filtering options.

1.5.1.1 BigML Compact Tree Visual Representation

BigML compact trees rely on the almost natural representation of BigML models as trees to convey a wealth of information in a compact format:

- The color of each node in the compact tree identifies the input field associated to it, i.e. the field that was identified as providing the best split. see Subsection 1.2.3 for more information.
- The first node at the top, called the tree root, corresponds to the split that best divides your data into two segments according to your objective variable. However, one must not confuse the best split with the most significant field since a root field may not always be as important as other predictors after growing the full tree. See Subsection 1.2.5 for more information.
- The branches connect the root to its child nodes at the next level down. Each child node may recursively have a number of children, to which it is connected through the afferent branches. This goes forth until we get to the terminal nodes, called leaves, that have no further children and are used to generate predictions.
- The width of a node represents the number of instances that the branch rooting at that node includes.



Figure 1.22: Tree visualization

1.5.1.2 Interacting with BigML Compact Trees

You can explore even more information that is encoded in your BigML model interacting with it. If you hover over the tree nodes, additional information about those nodes will be displayed. Additionally, you can set interactive filters to display only a subset of nodes based on a given criteria. Finally, you can create a dataset from a node to further investigate the data instances that support it. See below a description for all these interaction modes.

If you mouse over a node in the tree, BigML compact tree will display:

- The **prediction** at that node along with a measure of the **prediction's certainty** on the top navigation bar above the tree visual:
 - For classification models you can display the confidence or the probability by clicking in the switcher shown in Figure 1.23. See a detailed explanation of both measures in Subsection 1.2.6
 - For regression models you will get the expected error. See a detailed explanation in Subsection 1.2.7).



Figure 1.23: Node prediction

- The decision **prediction path** for the node, i.e. the series of rules that lead to it, on the right side of the tree.
- The number of **instances** that follow this decision path as well as their **distribution** in terms of the objective field values in the tooltip appearing on top of the hovered node. In the same tooltip you will see again the prediction at that node along with the confidence or probability (for classification models) or the expected error (for regression models).


Figure 1.24: Prediction path

Because models often grow too wide to comfortably fit the display area, BigML makes it possible to collapse some of the less important parts:

- To show all the nodes for a particular path, just click on that node and the whole branch will be displayed.
- To return to the default view, click on the root of the tree.

1.5.1.2.1 Interactive Filters

BigML compact trees provide a set of filters that let you interact with your model and control what part of it is displayed or hidden. Filters are interactive, meaning that you will see in real-time how applying a given filter changes the visualization. Filters aim to make it easier for you to analyze the information the model conveys in a number of ways, as explained below.

All available filters are shown in the top bar just above the tree, as depicted in Figure 1.25.

- Filter by support: this filter displays or hides branches based on how well they are supported by all data instances. To apply it, you specify the minimum and maximum percentage of instances that you require to provide support to visible branches. This is useful to identify branches with a minimum threshold of supporting instances.
- Filter by prediction: this filter displays or hides branches based on the predictions they provide. To apply it, you choose what predictions you are interesting in: for regression models, a slider allows you to set a range of values for your predictions; for classification models, select the predicted class.
- Filter by confidence, probability or expected error: this filter displays or hides nodes based on the certainty of the predictions they provide. For classification models you will be able to filter the model by the confidence or the probability depending on the option you select (see Figure 1.23). For regression models you will be able to filter your model by the expected error. This is useful to identify the most likely predictions your model provides.
- Rare interesting patterns: this filter displays only those branches that are supported by few instances, i.e. represent not so common cases, while still providing high confidence. Low support is defined as being supported by less than 1% of instances. High confidence is defined at least 50% confidence for classification models, or being in the lowest expected error decile for regression models.
- Frequent interesting patterns: this filter displays only those branches that have high support, i.e. represent frequent cases, and also provide high confidence. High support is defined as being



supported by more than 1% of instances. High confidence is defined at least 50% confidence for classification models, or being in the lowest expected error decile for regression models.

Figure 1.25: Interactive filters

1.5.1.2.2 Creating a Dataset from a Node

As mentioned, you can create a dataset collecting all the instances that support a given node. This can be useful if you want to apply other Machine Learning algorithms to it to gain further insight into the set of rules that belong to that branch.

Create a dataset from any node of your model by following these steps:

- · Click on a node to select a specific view of the model.
- Press the SHIFT key on your keyboard to freeze the current view.
- In the frozen view, click the Create dataset button. (See Figure 1.26.)
- · Release the frozen view by pressing the ESC key on your keyboard.

By executing the above listed steps, a new dataset will be created containing only the instances that support the selected node.



Figure 1.26: Create dataset from a branch

1.5.2 Partial Dependence Plot

Appart from the main compact tree view, BigML also offers an alternative view for models: the Partial Dependence Plot (PDP). The main goal is to represent the marginal effect of a set of variables (input fields) on the model predictions disregarding the rest of the variables. It is a common method for visualizing and interpreting the impact of the variables on the predictions, and it can be used for classification and regression models.

Sources Data	sets Supervised -	Unsupervised - Predictions -	Tasks	WhizzML 🗸
PDP	a* ••••	Diabetes diagnosis dataset	Training (70%) v1 🛛 📋 🗘	ૐ- (≡)⁵- ③
2	0	Diabetes: ?	?	? %
			8	â 🛛 🐺
		Ya		
Y Age	123 •	- (}	PREDICTION	
70 -				Confidence
65 -			min.	max.
60 -			False	
55 -			True	
50 -				
45 -			All classes	٣
40			INPUT FIELDS	All fields:
35 -			Pregnancies	2.50%
30 -				3
25 -				
			Blood pressure	6.66%
0. 50	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	9, 9, 4, 4, 9, 9,	0	68
x	Glucose	-	Skinfold	5.85%
			0	78

Figure 1.27: Model PDP view

Note: the model PDP is not a representation of the dataset values; it is a representation of the model results and their dependence from a set of variables used as inputs.

The PDP allows you to visualize the model predictions in the heatmap chart. In the case of **classi-fication** models, the different classes of the objective field are represented by **different colors**. The different color shadings for each class represent the different **confidences** or **probabilities** depending on the option you select from the switcher highlighted in Figure 1.28.

Sources	Datasets St	upervised -	Unsupervised 🔻	Predictions -	Tasks	WhizzML 🔻
- ¥.	60 - ••••	<u> </u>	Diabetes dia	gnosis dataset 1	Fraining (70%) v1	〕
<u> </u>	i@)		Diab	etes: True	76.37%	23
						<u>.</u>
Y ВМІ	_	23 🔻	48.2281	* Q		esc Releases v
- Divin	_		40.2201	Cx	PRI	EDICTION
					True	Confidence
60 -					44.8%	76.4%
					E-1-2	
50					False	•6.19%
					True	76.37%
40 -						
					All classes	v
30 -						UTFIELDS All fields:
20 -						_
					Pregnancies	21
10 -					°	3
0.0					Blood pressure	142
0.	N N 6	°, %	28 1.NO	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	·	68
	X Glucose	123	•	167.9063 🖋	Skinfold	5.85%
					0	78

Figure 1.28: Classification model PDP

For **regression** models, the different prediction values for the objective field are represented by differences in the **color scale**.

a 🖅 📴 🚥 🖇	Fictional Wine Sales dataset	- sample (80.00%.	- 📑 🗘	(≡) [*] -
	Total Sales: 48.87		24.01	
			80	8 0
Y Grape	Direct Origin			esc Releases
Y Grape ABC V	Pinot Grigio		PREDICTION	
		Total Sales		
Reisling		28.8		122.5
Prosecco			48.8654	
Pinot Noir			1010001	
Pinot Grigio				
Merlot		17	INPUT FIELDS	All fields:
		Country	-	5.62%
Malbec		Argentina		٣
Chardonnay				
Cava		Rating		8.36%
		88		92 89
Cabernet Sauvignon	\$ \$ \$ \$ \$ \$ \$ \$ \$			-
8° ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	17° 12° 12° 13° 13° 13° 13°	From Oregon?		1.40%

Figure 1.29: Regression model PDP

You can select any categorical or numeric field for each axis. You can also switch the axis by clicking on the option on top of the chart area. In the grey area next to the axis selectors you can see the axis values. You can freeze the view by pressing Shift and release it again by pressing Escape from your keyboard. When the view is frozen, an edition icon will appear and you can edit the axis values to obtain a prediction for that value. (See Figure 1.30.)



Figure 1.30: Model PDP axis options

The prediction legend next to the chart allows you to visualize the **objective field classes** (classification models) or the predicted **value** (regression models). In the case of classification models you will also obtain the **confidence** or the **proabbility** depending on your selected option (see <u>Subsection 1.2.6</u>). By default, the color tones and shadings are set according to the range of values shown in the chart area. This is the default because for some configurations of the chart the predictions may vary a small amount relative to the global range. For example, imagine the chart is showing temperature predictions based on location, time-of-year, and time-of-day. San Diego's daily range (13° C to 18° C) could be tiny compared to the Earth's global range (-62° C to 48° C). You can change this behavior and see the color scales and shading according to the total range of possible predicted values by clicking on the icon next to the probability bar Total. For classification models, this option allows you to see the **color shading** for the total range of potential confidence or probability values (from 0% to 100%). For regression models, the Total colors option allows you to see the **color scale** for the total range of predictions. For classification models you can also select to see only one of the classes using the class selector at the bottom of the legend. (See Figure 1.31.)



Figure 1.31: Prediction legend options

Below the chart legend, you can find the input fields form. (See Figure 1.32.) You can configure the values for any **numeric**, **categorical**, **text** or **items** field. By changing their values, you can see the predictions changing in real-time. You can sort the fields by their importance, select or disable them. If you disable an input field, it will be ignored to calculate the final prediction. The strategy used to calculate predictions when some fields are disabled is the **proportional missing strategy** (see Subsection 1.7.3.1).

ources Da	atasets	Supervised 🗸	Unsupervised -	Predictions -	Tasks		WhizzML 🔻
1 1	¶n, •••	• 5	Diabetes dia	Ignosis dataset	Training (70%) v1	â 🗘	ૐ- (≡)⁵- (
s: 🔳	i)		Diab	etes: True	76.3	7%	{? %
						80	8 0 9
Y ВМІ	_	123 🔻	48.2281	<i>•</i> {}			esc Releases vi
	_		10.2201	CX.	True	PREDICTION	Confidence
60 -					44.8%		76.4%
50 -					False		6.19%
					True	_	76.37%
40 -							
30 -					All classes		*
					1E	INPUT FIELDS	All fields: 🔲
20 -					Pregnancies		2.50%
10 -					-0		- 3
0.0					Blood pressure		6.66%
0.0 40	10	S S 2	120 1.40	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	0		142 68
х	Glucose	12	3 V	167.9063 🖋	Skinfold		5.85%
					0		78

Figure 1.32: Input fields from in the model PDP

If you want to know more about how to interpret the PDP view, please refer to Subsection 2.5.1.2.

1.5.3 Sunburst Visualization

In addition to compact trees and the PDP, BigML provides an additional, BigML-unique visualization for BigML models: the Sunburst, which allows you to visualize the entire model in a single view, without collapsing or expanding branches.

The Sunburst diagram can be seen as a representation of a BigML tree as if you were looking at a real pine tree directly down from its tip. So, the center circle in Figure 1.33 represents the root of the tree and its children wrap around it. Similarly, moving from the center further out, you will see the first split of the tree followed by the others.

For the uninitiated, BigML Sunburst diagram may not appear as immediately intuitive as our regular tree view. Nonetheless, it can help advanced users gain more insight into their models. In particular:

- More layers indicate deeper (i.e., more complex) prediction paths.
- The arc length of each child ring corresponds to the percentage of the training set covered by that child.
- · Smaller arcs indicate lower support for that child.
- When you mouse over the chart, the prediction rules will be shown to the right of the diagram as in the tree visualization view. However, the Sunburst diagram summarizes the prediction rules whenever possible. For example, two rules applicable to a leaf node may read: x <= 2 and 2 < x < 7. The sunburst view would combine these conditions and restate it as x < 7, which makes it easier to read and understand.

• When you click on a child ring, you zoom in the diagram, and all layers above the clicked one are collapsed into the diagram center. In the pine tree analogy, clicking on a ring would be equivalent to moving the view point down along the tree, so you can see more detail from the bottom layers. Click the center of the Sunburst to move up one level up to the tip.



Figure 1.33: Sunburst visualization

As a final note, you can color the chart by field, by prediction, or by confidence (or expected error in case of regression trees).

	Sources Datasets Supervised	I ■ Unsupervised ■ Predictions ■ Tasks		WhizzML 🔻
	≙ ₽ ₽↓ <mark>6</mark> 8° •••••	Diabetes diagnosis dataset Training (70%) v1	â 🗘	(≡) ⁵ , (Ξ)
Channe		Diabetes: ?	D	? %
Change colors				• 3
	0%	CONFIDENCE 100%		

Figure 1.34: Change Sunburst colors



Figure 1.35: Sunburst colors

1.6 Model Summary Report

From a model's detail view, you can access the model summary report by clicking on the icon highlighted in Figure 1.36.

Sources	Datasets	Models -	Clusters	Anomalies	Associations	Predictions	Tasks			Executions -
	'ılı	ଣ୍ଡ _ୁ ••••		wine	s dataset - sa	mple (80.00%	, D	Ê	¢	

Figure 1.36: Button to display a model's summary report

The model summary report has two tabs: field importance and summary, explained in the following subsections.

1.6.1 Field Importance

The field importance provides you a measure of how important a field is relative to the other fields. See Subsection 1.2.5 for more details.



Figure 1.37: Field importance

1.6.2 Summary

This summarized view of your model, includes:

- **Data distribution**: for classification problems, percentage of the dataset instances that belong to each of the objective field classes; for regressions, this is the objective field distribution.
- **Predicted distribution**: for classification problems, percentage of predicted instances for each of the objective field classes; for regressions.
- Field importance: a measure of how important a field is relative to the other fields (see also Subsection 1.2.5).
- **Rules summary**: a summary of the rules that the model learned from the dataset This means that for each possible prediction outcome, you can find here a summary of the paths that would produce that prediction.

Model Summary Report	×
Field importance Summary	
Data distribution: Iris-setosa: 33.33% (50 instances) Iris-versicolor: 33.33% (50 instances) Iris-virginica: 33.33% (50 instances) Predicted distribution: Iris-setosa: 33.33% (50 instances) Iris-versicolor: 33.33% (50 instances) Iris-virginica: 33.33% (50 instances) Field importance: 1. petal length: 70.19%	
DATA DISTRIBUTION	PREDICTED DISTRIBUTION
	Close

Figure 1.38: Summary Report

1.7 BigML Model Predictions

1.7.1 Introduction

The ultimate goal in building a BigML model is being able to make predictions for previously unseen instances with an unknown label. In BigML, you can make predictions for **single instances** or for **many instances in a batch**. Each **prediction** comes with a measure indicating its **reliability**, expressed either as a **confidence**, **probability** or as an **expected error**.

The predictions tab in the main menu of your BigML Dashboard is where all your saved predictions are listed. (See Figure 1.39.)

Sources	Datasets	Supervised -	Unsupervised 🔻	Predict	tions 🔻 Tasks	_	WhizzML	- •
ŧ	e				Predictions		Ô	Q,
ជា	Name			\$	Objective	\$ Prediction	\$ ÷	\$
				N	o predictions			
Show 10	predictions			No p	redictions found		< >	×

Figure 1.39: Predictions list empty view

Model predictions are saved under the CLASSIFICATION & REGRESSION option in the menu. (See Figure 1.40.)

CLASSIFICATION & REGRESSION		
CLUSTER ANALYSIS		â Q
ANOMALY DETECTION	Prediction	÷ 🛗
ASSOCIATION DISCOVERY		
TOPIC MODELING		
	CLUSTER ANALYSIS ions ANOMALY DETECTION ASSOCIATION DISCOVERY	CLUSTER ANALYSIS ions ANOMALY DETECTION Image: Cluster of the second se

Figure 1.40: Menu options of the predictions list view

From this view you can select to view the list for your **single instances predictions** or your **batch predictions** by clicking in the corresponding icons. (See Figure 1.41 and Figure 1.42.)



Figure 1.41: Single predictions icon



Figure 1.42: Batch predictions icon

In the predictions list view, you can see, for each prediction, the **Model or Ensemble** icon used for the prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created). (See Figure 1.43.)

Sources	Datasets	Supervised -	Unsupervised -	Predict	tions 🔻 Tasks		Whizz	ML 🔻
=	Ē			2	Predictions		Ô	Q
សា	Name			\$	Objective	\$ Prediction	\$ e H	🗄 🗘
				N	o predictions			
Show 10	predictions			No p	redictions found		< <	> >1

Figure 1.43: Predictions list view

You can also search your predictions by name by clicking the search button on the top right menu.

1.7.2 Creating Model Predictions

As shown in Figure 1.44, BigML provides three options to make predictions from your models:

- PREDICT BY QUESTION: to predict a single instance answering just the relevant questions required by the model.
- PREDICT: to predict a single instance using the prediction form.
- BATCH PREDICTION: to predict multiple instances simultaneously.

							Models	;								Q,
II	Name			\$	Туре	\$	Objective	\$	÷	\$	Ê	\$	\$		\$	08
ul <mark>î</mark>	iris datase	t's mo	odel	\odot	00		species		20	16h	4.5	KB	0	0		0
how 10	ᅌ moc	\$ [?]	PREDICT BY QUESTION			1	to 1 of 1 models							I< <	1)	>
		\$	PREDICT													
		۲	CREATE BATCH PREDICTION													
			EVALUATE													
		۲ <mark>۹</mark>	VIEW DETAILS													
		ស	DELETE MODEL													
		2	MOVE TO		>											

Figure 1.44: Menu options to create predictions

1.7.2.1 Predict by Question

This option will ask for a series of input field values, one question at a time, required to make a single instance prediction. In the general case, only the values of a subset of all input fields are needed to reach a prediction. So although your model may have seven data fields as potential predictors, if it is able to make a prediction with only two answered questions, then you will not have to specify the remaining seven fields by way of answering more questions. Follow these steps:

1. Start the questions by clicking Predict :

Sources Datasets Me	odels • Clusters Anomalies Associations Predictions Tasks	Scripts 👻
Se 11° 25°	Predict using Diabetes' dataset Trai	45 - (i)
	Diabetes: ?	?
Click to g	get the first question Answe	rs:
	Predict &	Start guestionnaire

Figure 1.45: Predict Question by Question first step

2. Set the values for each question and click Next question :

Sources Datasets Models	Clusters Anomalies Associations Predi	ictions Tasks Scripts	NEW
🗟 ៧৮ 🖏	Predict using Diabetes' dat	taset Trai 🤣 🤇	0
	Diabetes: False	60.96%	
Question: Glucose?		Answers:	
0	248 124		Set a value
Start over	Next question		Next

Figure 1.46: Predict Question by Question second step

3. After the model gives you a final answer, you will be able to optionally Save the prediction:



Figure 1.47: Predict Question by Question third step

1.7.2.2 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the model, so you can easily set the input data and get an immediate response. This option is only available from the BigML Dashboard for models with less than 100 fields. If you want to perform single instance predictions for models with a higher number of fields, you can use the BigML API¹⁷.

Follow the steps detailed below to create a single prediction:

1. Choose the PREDICT option under the model 1-click menu. (See Figure 1.48.)



Figure 1.48: Predict option from model 1-click menu

Alternatively, you can choose the PREDICT option in the **pop up menu** in the list view as shown in Figure 1.49.

¹⁷https://bigml.com/api/predictions

Source	es Datasets Models - Clusters	Anoma	lies	Associations	Predictions	Tasks			Scri	ipts 🗸
	Models								Q,	
ıll	Name	\$	Туре	Objective	\$	⊞ ≎	ê	\$	😫 ÷	○ ● ○ ≎
ılî	wines dataset - sample (80.00%)'s model	8	$\mathbf{\Lambda}$	Total Sales		2d 19h	564 bytes	0	0	0
ılî	wines da PREDICT BY QUESTION		₼	Total Sales		2d 19h	724 bytes			
ılî	iris datas		*	distance		2d 22h	7.1 KB			
ıl ^Q	wines da		$\mathbf{\Lambda}$	Total Sales		1 w 3d	724 bytes	1	1	
ılî	wines da		*	Total Sales		1w3d	413 bytes			
ılî	diab data MOVE TO	3		Diabetes		1w 5d	637 bytes			
II.	iris dataset - sample (100.00%)'s model		00	species		1 w 6d	4.5 KB		1	

Figure 1.49: Predict option from model pop-up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the model as predictors ordered by field importance. The importance percentage is found next to the field name as shown in Figure 1.50. You may not find all the fields from your original dataset because the model may find them irrelevant or redundant in terms of their predictive impact.

Sources Datase	ts Models -	Clusters Ano	malies Associat	ions Predictions	Tasks	Script	is 🕌
≣ີໃ ເມື່ ເລື	L	Predict	t using wines da	itaset - sample (80	.00%	∽.	()
			Total Sales:	61.43	16.6*	D	
		Missing	strategy:	8		All input fields:	€
Price 2.44	Field importance: 5	41.50	58.66%	Grape Cabernet Sauvign	-	27.41%	• • • • • • • • • • • • • • • • • • •
Country Argentina		•	13.80%	Rating		91	1 90
New prediction nat						Save	

Figure 1.50: Single predictions form

3. Select the fields you want to be taken into account for your prediction as shown in Figure 1.51. Non-selected fields will be considered as missing values during the prediction. If your model was trained with Missing splits (see Subsection 1.4.4), then missing values are considered by the model as any other valid value. If your model was built without missing values then any of the Missing strategies may apply during your prediction (see Subsection 1.7.3.1.)

Sources Datasets Mode	ls	ons Predictions	Tasks	Executions -
	Predict using wir	ies dataset's mode	ł	5 - i
	Total Sales:	61.43	16.61	
	Missing strategy:	8	(All input fields: 🕑
Price 2.44	41.50 21.41	Grape Cabernet Sauvign		24.87%
Country	●5.66% ▼	Rating 88	9.	0.09% 🕑
New prediction name Prediction for Total Sales				Save

Figure 1.51: Select fields in the prediction form

- 4. Set input values for your selected fields. Depending on the field type, you will need to input the values differently:
 - Numeric fields: move the slider or input a specific value in the edition box.
 - · Categorical fields: select one class from the selector.
 - Text fields: write one or several terms in the free text box.
 - Date-time fields: select the appropriate values from the selector.
 - Items fields: when you write the first three characters of an item name, several items matching those characters will appear, so you can select the right one. You can input more than one item for a field.
- 5. Get the prediction along with the confidence, the probability or expected error on the top of the form. BigML predictions are synchronous, i.e., when you send the input data you get an immediate response. Read more about local predictions in Subsection 1.7.2.2.1.

S II A	Prediction f	or Total Sales		Ğ-	0
	Total Sales: 0	61.43	16.61		
	Missing strategy:	8		All input fields:	V
2.44	41.50 21.41	Grape Cabernet Sauvign	-	24.87%	•
Country Argentina	5.56% v	Rating	9		9 0
New prediction name					

Figure 1.52: Single predictions view

6. Optionally Save the prediction so you can access them afterwards from the model predictions list view.

Sources Datasets Models -	Clusters Anomalies Associations Predictions	Tasks Executions
ື ຟ [ີ] ສີ	Predict using wines dataset's mode	نة بالمانية (i)
	Total Sales: 61.43	16.61
	Missing strategy:	All input fields: 🗹
2.44	69.38% C Grape 41.50 21.41 Cabernet Sauvign	24.67%
Country Argentina	• 5.66% Kating	91 91 90
New prediction name Prediction for Total Sales		Save

Figure 1.53: Single predictions list view

1.7.2.2.1 Local Predictions

BigML provides local predictions from the BigML Dashboard for single instance predictions. Local predictions allow you to get a real-time prediction without consuming any credits or requiring an internet connection. This is possible because your model is **saved in the browser's memory** so when the input values change, BigML immediately navigates your model to obtain their predictions in a matter of microseconds.

1.7.2.2.2 Predictions with Images

BigML models can be trained from images using extracted image features (Subsection 1.2.8). Because

image features are automatically generated numeric fields, creating model predictions with images is the same as creating other models. The only thing different is input fields of images.

Note: When the input fields contain images, in order to create the single prediction, BigML will extract image features automatically to match what were used in the dataset to train the model.

Sources	Datasets	Supervised -	Unsupervise	d - Predictions	 Tasks 			WhizzML 🔻
T .			F	Predict using grap	pe-strawberr	у		(≡) ⁵ √ (Ξ)
				label: ?		?		۲
Missing strat	tegy: 💦	&	% {?	Select a positive clas	s v Proba	bility threshold:	6	50%
								All input fields: 🗹
image_id				= 100.00% <				
Se	elect image			^ 🗋				
New pred	1659.jpg closed, ii) mage, 2 fields (1 pat	h, 1 image)					
	696969.						Θ.	Predict
		images.zip mage, 6 sources, 23	6 fields (234	E				

Figure 1.54: Select a single image source in the image input field

The model in Figure 1.54, "grape-strawberry", was created from a dataset containing image features *Histogram of gradients*. This set of image features are extracted by default for all image composite sources. Creating a prediction using the model will be directed to the **prediction form** which presents all input fields used by the model. One of them is the image field. Because this is a single prediction, an image is input by using a single image source. Clicking on the input field box, single image sources available will be in the dropdown list. There is also a search box which can be used to locate specific ones.

Sources	Datas	ets Supervised -	Unsupervised	Predictions	Tasks		WhizzML 🗸
** *			Pro	edict using grape-	strawberry		(≡) ⁵ √ (Ξ)
				label: ?		2	۲
Missing st	rategy:	8	% {?	Select a positive class	Probability thresho	ld:	50% All input fields:
image_id				100.00%			
	Select ima	ige		^ D			
		96969.jpeg pen, image, 2 fields (1 patł	n, 1 image)				
New pree grape-s		redict-images.zip losed, image, 6 sources, 2	List components			۵,	Predict
		rape-strawberry.zip losed, image, 70 sources, 2	237 fields (1 c	=			

Figure 1.55: List the components of a composite source

Oftentimes single image sources were used for creating a composite source, they become component sources of the composite source. Or an image was uploaded as a part of an archive file (zip/tar) which created a composite source. In those cases, the composite source will be shown in the dropdown list, along with an icon "List components". In the example in Figure 1.55, predict-images.zip is a composite source, click on the icon to show its component sources.

Sources	Datasets	Supervised -	Unsupervised -	Predictions -	Tasks		WhizzML 👻
¥.			Prec	lict using grape-s	strawberry		ॐ (≡) [*] , (≣)
				label: ?		?	۲
Missing str	rategy: 💦	&	% ? S	elect a positive class	 Probability thres 	hold:	50%
							All input fields: 🕑
image_id				00.00%			
	Select image			^ D			
	🗧 🚺 prec	lict-images.zip					
New prec				۹			
grape-s			, 1 image)			e.	Predict
	1659.jp					<u> </u>	
	328320. open, im	.jpg age, 2 fields (1 path,	, 1 image)				

Figure 1.56: Select a component of a composite source

After the component sources of the composite are listed, scroll the dropdown list to find the desired one, then click to select it, as shown in Figure 1.56. There is also a search box to locate specific component sources.

Sources Datasets	Supervised - Unsupervi	ised - Predictions -	Tasks	WhizzML 🔫
₩.	2	Predict using bluebird-g	ps-tagged	ॐ ⊷ (≡) [*] ∕⊷ (i
		label: ?	?	۲
Missing strategy:	8 7	Select a positive class	Probability threshold:	50%
				All input fields: 🗹
gpslongitude		91.51% 🕑 image_i	id	• 8.49%
-158.32	45.18	-97.10325	Select image	*
New prediction name				
bluebird-gps-tagged				Predict

Figure 1.57: Model prediction form, image field and more

In addition to images, models may use other fields, which will be in the prediction form too. As shown in Figure 1.57, all the fields can be selected, and their input values be set by dragging the knobs on the sliders or by entering precise values in their input boxes.

Once all fields are selected, click on the green button	Predict	to create a prediction.
---	---------	-------------------------

Sources Datasets Supervis	sed - Unsupervised - Predict	ions 👻 Tasks	WhizzML 👻
A 77.	grape-	strawberry	
	label: grap	e 97.92%	۲
<u>In</u>	97.92%		
		2.08%	
	State.	strantart	
Missing strategy:	Select a positive	Probability threshold:	50%
image_id	· 100.00%		All input fields: 🗹
1659.jpg	× • D		
New prediction name			
grape-strawberry			Predict

Figure 1.58: Model image single prediction

After a new prediction is created, as shown in Figure 1.58, the predicted class is at the top of the form along with its probability. The prediction interface is the same as ones created by non-image models. Everything described earlier in this section (Subsection 1.7.2.2) applies.

1.7.2.3 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the model you want to use to make predictions and a dataset containing the instances for which you want to obtain prediction. BigML will create a prediction for each instance in the dataset. Follow the steps detailed below to create a batch prediction:

1. Select the BATCH PREDICTION option under the model 1-click menu (see Figure 1.59) or the CRE-ATE BATCH PREDICTION option in the pop up menu of the list view (see Figure 1.60.)

Sources Datasets Models	Clusters Anomalies	Associations Pre	dictions Tasks		Scripts
	••• 🔊 wines da	taset - sample (80.00	%)'s model	🏮 🗘 🦽	(=) ^{/-}
<u> (0) (1) </u>		Total Sales: ?	PREDICT QUESTION	BYQUESTION	e
			PREDICT		
SUPPORT 12.5 % 25		TPUT 112.07			
			EVALUATE		
			CALETE MODEL		
	1		MOVE TO		

Figure 1.59: Batch predictions option from model 1-click menu

Source	s Datasets Mo	dels 🔻 C	lusters	Anomal	ies ,	Associations	Predictions	Tasks			Scri	pts -
						Models	6					Q,
III	Name			\$	Туре 🗧	Objective	\$	# \$	₿ ≎	\$	\$	08 ÷
ılî	wines dataset - sample	e (80.00%)'s m	nodel		$\mathbf{\Lambda}$	Total Sales		2d 19h	564 bytes			
ılî	wines dataset's model				入	Total Sales		2d 19h	724 bytes	0	0	0
ılî	in s datas	CT BY QUESTI	ON		$\mathbf{\Lambda}$	distance		2d 22h	7.1 KB			
ılî	wines da	CT	DICTION		杰	Total Sales		1w3d	724 bytes	1	1	
ıl	wines da				⋆	Total Sales		1w3d	413 bytes			

Figure 1.60: Batch predictions option from model pop up menu

- Select the dataset containing all the instances you want to create a prediction for. The instances should contain the input values for the fields used by the model as predictors. You can also select a subset of the model fields to be taken into account by configuring your prediction (see Subsection 1.7.3.4.) BigML batch predictions can handle missing data in your prediction dataset (see Subsection 1.7.3.1.)
- 3. **Optionally, select the model** you want to use for the prediction. BigML pre-selects the model you created the batch prediction from at step 1, but you can change it at any time in the batch prediction view by selecting another model from the model selector displayed in the right pane. You can even switch to an ensemble or logistic regression by selecting the corresponding icon in the top left menu.

1° 5°	New Batch Pr	ediction	∽-
81 Fr 1			
wines dataset - sample (80.00	%)'s model × v	Search dataset win wines dataset - batchprediction - ba	A Q
0.6 KB 5 size fields	16 instances	wines dataset - batchprediction - batchpredictio	atchprediction raining (80%)
Description:	$\overline{\mathbf{O}}$	wines dataset - sample (80.00%) T wines dataset - sample (20.00%) wines dataset - sample (80.00%)	est (20%)
Prediction name:		wines dataset - batchprediction	

Figure 1.61: Select dataset for batch predictions

4. After you have selected the model and the dataset, the batch prediction configuration options (see Subsection 1.7.3) will appear along with a preview of the prediction output, which is formatted as a comma-separated list of values (CSV format). (See Figure 1.62.) The default output includes all the fields in your prediction's dataset plus a last column containing the calculated predictions.

Note: BigML does not include the predictions' confidence, probability or expected error by default so you will have to configure your output file to include that information as explained in Subsection 1.7.3.5.

Sources Datasets Models - Clusters	Anomalies Associatio	ns Predictions	Tasks	Scripts 👻
	New Batc	h Prediction		5 - (i)
81 Fr 🔟				
wines dataset - sample (80.00%)'s mod	el x v	wines data	set - batchprediction	× v
Total Sal	Fri, 20 May 2016 15:55:05	I		Fri, 06 May 2016 17:20:26
0.6 KB 5 size fields	16 instances	1.0 KB size		20 instances
Description:	$\overline{f O}$	Description:		⊙
Configure				◙
Preview of the prediction file (using the type of eac	h field)			
Country, Grape, Rating, From Oregon?, Pric ABC, ABC, 123, ABC, 123, 123, text, 123, 123 ABC, ABC, 123, ABC, 123, 123, text, 123, 123	e,Total Sales,error,Tot	al Sales,Total Sales		
Prediction name:				
Batch Prediction of wines dataset - sample	(80.00%)'s with	i.	Reset	Predict

Figure 1.62: Configuration options displayed and output preview

5. By default, BigML generates an output dataset containing the batch prediction results. You can find in BigML Dashboard's dataset list view and can use it as any other dataset to analyze the batch prediction output afterwards. If you do not want a dataset with all the prediction results to be created, you can deselect the button highlighted in Figure 1.63.



Figure 1.63: Create dataset from batch predictions

6. Once you are done configuring your batch prediction, click the predict green button to generate it. This process may take some time depending on the size of the input dataset. (See Figure 1.64.)

ources Datasets Models - Clusters			(S	Scripts 🗸
ilî Kî	New Bato	ch Prediction		\$ - (
63 F6 1				
wines dataset - sample (80.00%)'s mode		wines dataset	- batchprediction	× v
rt 🕂 🗘 Total Sal	Fri, 20 May 2016 15:55:05	ılî		Fri, 06 May 2016 17:20:26
0.6 KB 5 size fields	16 instances	1.0 KB size	8 fields	20 instances
Description:	$\overline{\mathbf{O}}$	Description:		\odot
Sonfigure				\odot
Preview of the prediction file (using the type of each Country, Grape, Rating, From Oregon?, Price ABC, ABC, 123, ABC, 123, 123, text, 123, 123 ABC, ABC, 123, ABC, 123, 123, text, 123, 123 ABC, ABC, 123, ABC, 123, 123, text, 123, 123		al Sales,Total Sales		
ABC,ABC,123,ABC,123,123,text,123,123 ABC,ABC,123,ABC,123,123,text,123,123				

Figure 1.64: Create batch predictions

 After the batch prediction has been created, you will be able to download a CSV file with all the instances found in your input dataset along with the prediction corresponding to each one of them. (See Figure 1.65.)

Sources Datasets Models - Clusters And	nalies Associations Prediction	s Tasks	Scripts 👻
	Batch Prediction of wines data	ase	% - ()
Wines Dataset - Sample (80.00%)'S	Model 🔊	es Dataset - Batchprediction	©
Output preview			
Country,Grape,Rating,From Oregon?,Price,Tot Austraila,Reisling,,No,14.95,95.73333,,73.6 Argentina,Chardonnay,89,No,19.99,61.42857,, Austraila,Reisling,,No,18.95,25.4375,,41.44 Argentina,Pinot Grigio,,No,8.95,112.06667,, France,Malbec,,No,17.99,54.73913,,58.356126 Argentina,Cava,No,17.95,52.93694,,53.11715 Spain,Cava,90,No,28.95,54.15789,,61.481952, France.Pinot GriaioNo.14.99.90.9090990	1844,25.4375 0.851683,61.42857 092,25.4375 07.16666800000003,112.06667 76.91072 ,52.93694 4.15789	Sales	

Figure 1.65: Download batch prediction output CSV file

8. If you did not disable the option to create a dataset, as explained above (see step 4), an Output dataset button will also be available to allow you to directly jump to the output dataset. (See Figure 1.66.)



Figure 1.66: View batch predictions output dataset

1.7.2.3.1 Batch Prediction with Images

BigML models can be trained from images using extracted image features (Subsection 1.2.8). The input of a batch prediction is a dataset. So when creating a batch prediction with images, the dataset has to have the same image features used to train the model. The image features are in the dataset used to create the model.

I Y		New Batch Pre	diction	۵. (
ର 🐔 🏄	∑ ×			
bluebird-gps-ta	gged	× •	Search dataset	
	6 insta	predict-images ances, 236 fields (234 numeric, 1 pat/ image), 1 non-preferred	predict predict-images 6 instances, 236 fields (234 nume	م eric, 1 path, 1 image), 1 n
36.6 MB size	5 fields	instances	iris dataset - extended - samp 60 instances, 142 fields	le (100.00%) - extende
		\odot	iris dataset - extended - samp 60 instances, 139 fields	le (100.00%) - extende
Description:			iris dataset - extended - samp	le (100.00%) - extende

Figure 1.67: Batch prediction using an image dataset

As shown in Figure 1.67, the input for the model batch prediction is selected as predict-images, which is a dataset consisting of six images and contains the default set of extracted image features, *Histogram* of gradients.

Image features are configured at the source level. For more information about the image features and how to configure them, please refer to section Image Analysis of the Sources with the BigML Dashboard¹⁸[11].

¹⁸https://static.bigml.com/pdf/BigML_Sources.pdf

For the rest of batch predictions with images, including batch prediction configuration options and output datasets, everything stated earlier in current section (Subsection 1.7.2.3) applies.

1.7.3 Configuring Model Predictions

BigML provides several options to change its default behavior when calculating predictions. For single predictions as well as for batch predictions you can configure the strategy used for handling **missing values** (see Subsection 1.7.3.1.) and set a **probability or confidence threshold** for a given class only for classification models (see Subsection 1.7.3.2.). For batch predictions, you can also set a **default numeric value** for missing values (Subsection 1.7.3.3), the automatic **fields mapping** performed by BigML (Subsection 1.7.3.4), and define the **output file settings** (Subsection 1.7.3.5.)

1.7.3.1 Missing Strategies

When you create a new prediction, BigML will automatically navigate through the corresponding model to find the leaf node that best classifies the new instance.

However, it may just so happen that your new data (the instances you want to predict) does not have populated values for all the fields used in building the original model. For example, imagine that you are trying to predict diabetes and you have the patient's glucose level and BMI (Body Mass Index) but not his blood pressure. If the model arrives at a node where the blood pressure level is required, BigML can handle this missing value by using one of these two strategies:

- Last prediction: it returns the prediction value and confidence or probability of the parent node.
- **Proportional**: it combines all subtrees predictions beneath the current node based on the data distribution of their child nodes in order to compute the prediction value and confidence or probability.

For single predictions you can select any of both Missing strategies by clicking in the icons shown in Figure 1.68.

Predict using wines dataset's mode Total Sales: 61.43 sing strategy:	el 65- 3 16.61 All input fields: 1 24.87% 1
sing strategy: 💦 🚴	All input fields: 🗹
69.38% 🗹 Grape	
	 24.87% 🕑
21.41 Cabernet Sauvign	v
5.66% C Rating	91 90
	5.66% C Rating

Figure 1.68: Missing strategies for single predictions

For batch predictions you can find both options under the configuration panel as shown in Figure 1.69.

Sources Datasets Supervised -	Unsupervised - Predic	tions 👻 Tasks		WhizzML 🔻
	New Bat	ch Prediction		5 - i
£1 F£1 ∠ ≤				
Diabetes diagnosis dataset Tr	aining (70%) v1 × ×	Diabetis dia	agnosis dataset Test (30%)	×
	Fri, 01 Dec 2017 19:20:59	ılî		Wed, 08 Jun 2016 15:39:39
18.9 KB 9 size fields	537 instances	8.1 KB size	10 fields	231 instances
Description:	\odot	Description:		$\overline{\mathbf{O}}$
ំ _{ភ្លំ} Configure				۲
	elect a positive class Proba	ability threshold:	50%	
Default numeric value: ? Select a d	lefault value			
Excluded fields:				$\overline{\mathbf{O}}$
1 Make sure that the fields in the M	odel and Dataset match			
Fields mapping:			Defaul	t fields 📀

Figure 1.69: Missing strategies for batch predictions

1.7.3.2 Confidence and Probability Threshold

Confidence and probability thresholds are only available for **classification** models, and they usually make sense when you want to minimize false positives at the cost of false negatives. The positive class will be predicted if its confidence or probability is greater than the given threshold, otherwise the following class with greater confidence or probability will be predicted instead.

To configure a threshold for your single predictions follow these steps:

1. Select the **probability** or the **confidence** measure using the buttons shown in Figure 1.70. To learn more about the differences between model confidences and probabilities refer to Subsection 1.2.6.



Figure 1.70: Select probability or confidence

2. Select the **positive class**, i.e. the class for which you want to apply the confidence or probability threshold:



Figure 1.71: Select the positive class

3. Set a value for the **threshold** using the slider. The positive class will only be predicted when the confidence or probability of the prediction is above the established threshold, otherwise the

Sources	Datasets S	upervised -	Unsupervis	ed - Predict	ons 🗸 Tasks		Wh	izzML 🔻
***			Predi	ct using Diabe	tes diagnosis da	taset Trai		e)∱_ (i)
			[Diabetes: True		95.56%)	۲
					95.56%			
			4.44%					
			False		True			
Missing strat	egy: 💦	8	1	True	× ▼ Probabi	lity threshold: 🗨	All input fie	
Glucose			-	38.56%	Age			.15% 🗹
0	G)	248	124	8	0	84	46
BMI			-	16.73%	Diabetes pedigree	9		51.96 💽
0.0	6		83.88	40.74	0.0	-0	3.01	1.46
Blood press	ure		140	6.66%	Skinfold			35%
0	6)	142	71	0	0	78	39

following class with higher probability or confidence will be predicted instead.

Figure 1.72: Set a threshold

For batch predictions, you will find the same options under the CONFIGURE panel. (See Figure 1.73.)

Sources Datasets Supervised - Unsupervised - Prediction	ons 🗸 Tasks	WhizzML 👻
New Batc	h Prediction	5 - (i)
ŝi fa ∠ ≤		
Diabetes diagnosis dataset Training (70%) v1 × ×	Diabetis diagnosis dataset Test (30%)	×
Diabetes Fri, 01 Dec 2017 19:20:59	Wed,	08 Jun 2016 15:39:39
18.9 KB 9 537 size fields instances	8.1 KB 10 size fields	231 instances
Description:	Description:	$\overline{\mathbf{O}}$
the Configure		۲
Select a positive class V Probab	bility threshold: 50%	
Default numeric value: Select a default value		
Excluded fields:		$\overline{\mathbf{O}}$
1 Make sure that the fields in the Model and Dataset match		
Fields mapping:	Default fiel	lds 💿

Figure 1.73: Configure a threshold for batchpredictions

1.7.3.3 Default Numeric Value

If the dataset used to make the batch prediction contains instances with **missing** values for the numeric fields you can easily replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero** using the **Default numeric value** before creating your batch prediction, (See Figure 1.74.)

ources Datasets Supervised - Unsup	ervised - Predictio	ns 👻 Tasks		WhizzML 👻
	New Batch	Prediction		نى 👶
£1 €£ £				
Diabetes diagnosis dataset Training (70%)	vi × v	Diabetis d	iagnosis dataset Test (30%)) × •
Diabetes Fr	i, 01 Dec 2017 19:20:59	ılî		Wed, 08 Jun 2016 15:39:39
18.9 KB 9 size fields	537 instances	8.1 KB size		231 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
්යු Configure				۲
Select a positiv	e class Probab	ility threshold:	50%	
Default numeric value: Select a default value				
Excluded fields:				⊚
1 Make sure that the fields in the Model and D	ataset match			
Fields mapping:			Defa	ult fields 💿

Figure 1.74: Default numeric value for batch predictions

1.7.3.4 Field Mapping

By default, BigML maps fields based on their **names**. If there is a mismatch between the field names in your model and those in the input dataset you selected for the batch prediction, you can specify the right correspondence between the two sets of fields by explicitly assigning to each field appearing in the "Model fields" column its associated input field in the "Dataset fields" column. (See Figure 1.75.)

If the dataset's and model's field names do not match but their IDs do, which happens when corresponding fields appear in the same order, you can tell BigML to use the **field ID** instead of the field name to map the fields. To this aim, click the green switcher shown in Figure 1.75.

If you do not want some of the fields to be considered during the evaluation, you can also **manually** search for those fields and remove them from the "Dataset fields" column.

ា <mark>ខ្ល</mark> ស្ន	New Ba	atch Prediction		
សិ 🕅				
wines dataset's model	×	wines d	lataset - batchprediction	× v
Total Sal	Fri, 06 May 2016 10:32:27	ıl		Fri, 06 May 2016 17:20:26
0.7 KB 5 size fields	20 instances	1.0 KB size		
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
Configure				۲
8. 8				
Make sure that the fields in the M	odel and Dataset match			
	odel and Dataset match		Appl	y this map 📀
1 Make sure that the fields in the M	odel and Dataset match	Dataset fields		y this map 🙆
1 Make sure that the fields in the M Fields mapping:	odel and Dataset match	Dataset fields		
1 Make sure that the fields in the M Fields mapping: Model fields			Auto-mapping by: NAM	
1 Make sure that the fields in the M Fields mapping: Model fields 1. Country	ABC	1. Country	Auto-mapping by: NAM	

Figure 1.75: Fields Mapping for batch predictions

The fields mapping from the BigML Dashboard has a **limit of 200 fields**. For batch predictions with a higher number of fields, use the argument field_map from BigML API¹⁹ if you need to map your fields.

1.7.3.5 Output Settings

As mentioned, batch predictions can create a CSV file containing all input instances along with the predictions BigML calculated for each of them. Define the following settings to customize your prediction file:

- **Separator**: this option allows you to choose a separator for your output file values. The default separator is the comma. You can also select the semicolon, the tab, or the space.
- New line: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".
- **Output fields**: this option allows you to include or exclude any of your dataset fields from the output file from the preview shown in Figure 1.76.

Note: a maximum of 100 fields are displayed in the preview, but all your dataset fields are included in the output file by default unless you exclude them.

- **Headers**: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability and/or confidence column name, field importances column names, etc.). By default, BigML activates the headers.
- **Prediction column name**: this option allows you to customize the name for your predictions column. By default BigML uses the name of the model's objective field.

 $^{^{19} \}tt https://bigml.com/api/batchpredictions\#bp_batch_prediction_arguments$

- **Confidence or expected error**: this option allows you to include an additional column in the output file with the confidence or expected error per instance. By default, neither the confidence nor the expected error are included.
- Confidence column name: this option allows you to customize the name for the confidence (or expected error) column in case you include it in the output file. By default BigML uses "confidence".
- **Probability**: this option allows you to include an additional column in the output file with the predicted class probability for each instance. By default, it is not included.
- **Probability column name**: this option allows you to customize the name for the probabilities column in case you include it in the output file. By default BigML uses "probability".
- All class confidences: this option allows you to include the confidences for each class in the objective field. There is a column per class, named "<class_name> confidence".
- All class probabilities: this option allows you to include the probabilities for each class in the objective field. There is a column per class, named "<class_name> probability".
- **Importances**: this option allows you to include a column for each of the field relative importances for the model predictions. There is a column per field, named "<field_name> importance".

Separator:	New line:			
, (comma)	Unix, Linux or OS X (LF)	<i>≋</i> ≡		
Prediction column name:	Confidence column name:	Probability column name:		
a,b,c -,-,-	2	{ %		
Output Fields:				
* Pregnancies	123 x Glucose	123	× Blood pressure 123	
* Skinfold	123 y Insulin		* BMI 123	
	123 x Insulin	123		
x Diabetes pedigree	123 × Insulin	123 123	x Diabetes ABC	
view of the prediction file (using th	123 x Age etype of each field) ssure, Skinfold, Insulin, BMI, Diabete 23, ABC, ABC, ABC 23, ABC, ABC, ABC 23, ABC, ABC, ABC 23, ABC, ABC, ABC	123	× Diabetes ABC	

Figure 1.76: Output settings for batch predictions

1.7.4 Visualizing Model Predictions

Model predictions visualization changes depending on whether you are predicting one **single** instance or you are predicting multiple instances using the **batch predictions** option. (See Subsection 1.7.4.1.)

1.7.4.1 Single Predictions

For single predictions you can find the prediction for your objective field at the top of the form along with the performance measure.

• For classification models you will find the objective field class predicted along with the **probability** or the **confidence** depending on which measure you select. You will also get all the class distribution histogram according to the measure selected, i.e., all class probabilities or all class confidences. (See Figure 1.77.)



Figure 1.77: Single predictions view for classification

• For **regression models** you will get a numeric prediction and the expected error for that prediction as shown in Figure 1.78.

	Prediction f	or Total Sales		(5 - (
	Total Sales:	61.43	16.61	
	Missing strategy:	8		All input fields:
2.44	41.50 21.41	Grape Cabernet Sauvign	-	24.87%
Country Argentina	• 5.86% V	Rating 88	91	0.09%
New prediction name				

Figure 1.78: Single predictions view for regression models

In either case, you can change any time the value of the displayed input fields to have your prediction recalculated in real-time.

If you have saved your prediction, you can go back to it and visualize it.

Read a detailed explanation of confidence, the probability, and expected error calculations in Subsection 1.2.6 and Subsection 1.2.7 respectively.
1.7.4.1.1 Prediction explanation

Prediction explanation helps understand why a model makes a certain prediction. This is very useful in many applications, and the reasons behind a model's prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

For any classification or regression model, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Save (see Figure 1.79).



Figure 1.79: Explain prediction

The prediction explanation represents the most important factors considered by the model in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 1.80. The importances across all input fields should sum 100%.

Sources [Datasets Superv	ised Unsupervised Predictions	- Tasks	WhizzML 🗸
8 ¥.		Titanic Su	rvival	ૐ- (≡)⁵- ③
		Survived: TRUE	77.69%	
% ?				ిం
PREDICTION EXP	PLANATION			
Input data		Importance		
Class/Dept	1st Class			81.98%
Fare today	24,719.00			11.90% +
Age	44.93			6.12% +
Joined	Belfast			0.00%

Figure 1.80: Input field importances

For some input fields you will see a "+" icon next to the importance. This is because the importance may not be directly associated with the input value, i.e., it can be explained by other reasons. In the Figure 1.81 below, the importance of 6.12% for the field "Age" is not explained by this field being equal to 44.93. Rather, it is because this field value is higher than 30.5 and lower than 45.47.

Sources Da	atasets Supen	vised 👻 Unsup	Predictions -	Tasks		WhizzML 🔻
₽ ₽ -			Titanic Surv	vival	چ.	(≡)⁵,
			Survived: TRUE	77.69%		
{%						ð
PREDICTION EXPL	ANATION					
Input data		Importance				
Class/Dept	1st Class					81.98%
Fare today	24,719.00					11.90% +
Age	44.93					6.12% +
Joined	Belfast				Detailed Explana	ation _{0.00%}
					<= 45.47 > 30.50	3.06% 3.06%

Figure 1.81: See the detailed explanation

The prediction explanation for models is calculated using the prediction path of the decision tree.

Note: the input field importances in the prediction explanation are different from the overall field importances of the model. A field can be very important for the model but insignificant for a given prediction.

1.7.4.2 Batch Predictions

For batch predictions, you always get a CSV file and an optional output dataset.

1.7.4.2.1 Output CSV File

From the batch prediction view, you can access the CSV file containing your **predictions** for each of your dataset instances in the last column. (See Figure 1.82.) You can configure several options to **customize your CSV file** including the separator for the columns, the name of your prediction column, the dataset

fields you want to include, whether you want to include a first row with the names of your columns. You can find a detailed explanation of those options in Subsection 1.7.3.5.

Note: by default BigML does not include the predictions confidence, probability or expected error in your output file. Again you will need to click that option from the output settings panel if you want to include it.

Sources Datasets Models - Clusters Anomalies Associatio	ns Predictions Tasks	Scripts -
a Statch Prediction	n of wines datase	5 - (i)
Wines Dataset - Sample (80.00%)'S Model 🛛 🕤	Wines Dataset - Batchprediction	⊚
Output preview		
Country, Grape, Rating, From Oregon?, Price, Total Sales, error, Tot Austraila, Reisling, , No, 14. 95, 95. 73333, , 73. 661844, 25. 4375 Argentina, Chardonnay, 89, No, 19. 99, 61. 42857, , 60. 851683, 61. 42857 Austraila, Reisling, No, 18. 95, 25. 4375, , 41. 444092, 25. 4375 Argentina, Pinot Grigio, , No, 8. 95, 112. 06667, , 107. 16666800000003 France, Malbec, , No, 17. 99, 54. 73913, , 58. 356126, 76. 91072 Argentina, Cava, No, 17. 95, 52. 93694, , 53. 117159, 52. 93694 Spain, Cava, 90, No, 28. 95, 54. 15789, , 61. 481952, 54. 15789 France, Pinot Griaio. No. 14. 99. 90. 90909. 90. 37270000000001.76.	, 3,112.06667	
Download batch prediction	Output dataset	

Figure 1.82: Download batch prediction output CSV file

See an output CSV file example in Figure 1.83 where the two last columns contain the prediction and the confidence for each instance.

```
Pregnancies, Glucose, Blood pressure, Skinfold, Insulin, BMI, Diabetes, Confidence
8,183,64,0,0,23.3, True,0.6574
5,116,74,0,0,25.6, False,0.845
10,115,0,0,0,35.3, True,0.6469
8,125,96,0,0,0.0, False,0.9356
1,189,60,23,846,30.1, True,0.7574
1,103,30,38,83,43.3, False,0.675
7,103,66,32,0,39.1, False,0.7682
1,101,50,15,36,24.2, False,0.948
0,100,88,60,110,46.8, False,0.5413
```

Figure 1.83: An example of a batch prediction CSV file

1.7.4.2.2 Output Dataset

By default, BigML creates a dataset out of your batch prediction. (See Subsection 1.7.3.5.) You can access your output dataset from the batch prediction view by clicking the Output dataset button shown in Figure 1.85.



Figure 1.84: View batch predictions output dataset

In the output dataset you can find an additional **field** (named by default as per your model's objective field) containing the **predictions** for each one of your instances. If you configured your batch prediction to include the **confidence**, the **probability** or **expected error** you will be able to find it in the last field of your output dataset as shown in Figure 1.85.

Sources Datasets Models	 Clusters 	Anomalies A	ssociations Pre	edictions Ta	isks	Scripts 🗸
a (∰° <mark>1</mark> 10, •••••	wir	nes dataset - bat	chprediction -	batchpredic	ction 🏾 🗞 - 🔇), (≡), (≡), (Ξ)
					٩	×
Name	🗢 Туре	Count	Missing	Errors	Histogram	
Country	ABC	19	1	0		
Grape	ABC	18	2	0		
Rating	123	6	14	0		-
From Oregon?	АВС	18	2	0		
Price	123	18	2	0		
Total Sales	123	11	9	0		
error	text	0	20	0		
Total Sales	123	20	0	0		
Total Sales	123	20	0	0		
Confidence	123	20	0	0		
Show 10 ᅌ fields		1	to 10 of 10 fields			к < 1 > >

Figure 1.85: Batch predictions output dataset

1.7.4.2.3 Batch Prediction 1-Click Actions

From the batch prediction view you can perform the following actions (see Figure 1.86):

- BATCH PREDICTION AGAIN: this option will redirect you to the batch prediction creation view, with the same model and prediction dataset already selected. This option allows you to rapidly recreate the batch prediction using a different configuration.
- BATCH PREDICTION WITH ANOTHER DATASET: this option allow you to easily create a batch prediction using the same model and a different dataset.
- BATCH PREDICTION USING ANOTHER MODEL: this option allows you easily create a batch prediction using the same dataset and a different model.
- NEW BATCH PREDICTION: this option redirects you to the batch prediction creation view where you can select a prediction dataset and a model to create your prediction.

Sources Datasets	Models - Cluste	rs Anomalies	Associatio	ns Predictions	a Tasks	Script	S
	б <u>т</u> ••••	Bato	h Predictio	on of iris datase	ət	G ₇	()
					CE BATCH PREDICT		
			-			ION WITH ANOTHER DATAS	
Iris Datase	et - Sample (100.0	0%)'S Model	♥	lris C	BATCH PREDICT	ION USING ANOTHER MOD	2
Output preview					I NEW BATCH PRE	DICTION	F .
					DELETE BATCH F	PREDICTION	
5.1,3.5,1.4,0.2,1 4.9,3.0,1.4,0.2,1 4.7,3.2,1.3,0.2,1 4.6,3.1,1.5,0.2,1 5.0,3.6,1.4,0.2,1 5.4,3.9,1.7,0.4,1 4.6,3.4,1.4,0.3,1	Il width,petal len; iris-setosa,Iris-s; iris-setosa,Iris-s; iris-setosa,Iris-s; iris-setosa,Iris-s; iris-setosa,Iris-s; iris-setosa,Iris-s; iris-setosa,Iris-s;	itosa itosa itosa itosa itosa itosa itosa	species, spe	ctes			

Figure 1.86: Batch prediction 1-click actions

1.7.5 Consuming Model Predictions

BigML provides plenty of means for developers to integrate BigML model predictions within their apps. In the following sections, we will describe how you can use the BigML REST API and the BigML Python bindings to work with model predictions.

1.7.5.1 Using Model Predictions via the BigML API

Model predictions have full citizenship in the BigML API. This means you can programmatically create, update, list, delete, and use them for predictions. For example, this is how you can create a single prediction using the command line from a given model and defining the input data. This will require properly setting the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"model": "model/50650bdf3c19201b64000020",
                          "input_data": {"000001": 3, "000002":4.5, "000003"}}'
```

For more information on using model predictions through the BigML API, please refer to prediction REST API documentation²⁰.

1.7.5.2 Using Model Predictions via the BigML Bindings

BigML bindings provide a convenient way to access BigML REST API from your language of choice. They offer a higher-level view of BigML Machine Learning resources and algorithms in a number of

²⁰https://bigml.com/api/predictions

languages, including Python, Node.js, Java, Swift, and Objective-C. For example, this is how you can create a model prediction in Python using BigML bindings:

BigML bindings also provide the means to carry through predictions locally, without ever hitting the network, which can greatly improve the latency of predicting from your apps. This is made possible by BigML models being white-box, meaning you can download them and use them independently from BigML. For example, the following code snippet shows how you can download a model and use it for making a local prediction using the BigML bindings for Python:

For more information on using models through the BigML bindings, please refer to BigML bindings documentation.

1.7.6 Descriptive Information

Descriptive information is what allows you to describe a prediction so you can find it later and easily recognize it among other predictions.

Each prediction has an associated **name**, **description**, **category**, and **tags**. You can find a brief description for each concept in the following sections. In Figure 1.87, you can see the options that the **More info** panel gives to edit them.



Figure 1.87: Edit predictions

1.7.6.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: the name always follows the structure "Prediction for <objective field name>"
- **Batch predictions**: BigML combines your prediction dataset name and the model name: "Batch prediction of <model name> with <dataset name>".

Predictions names are displayed on the list view and also on the top bar of a prediction view. Predictions names are indexed to be used in searches. You can rename your predictions at any time from the **More info** panel.

The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, since they are automatically assigned unique internal identifiers.

1.7.6.2 Description

Each model prediction also has a **description** that it is very useful for documenting your Machine Learning projects. Predictions take the description from the models used to create them.

Descriptions can be written using plain text and also markdown²¹. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 1.88.)

Edit description	>
You can add formatting and links using a simple markdown langua B $I \models = \infty$	ge:
Write your **description** here	
Description:	
Write your description here	
Can	cel Update

Figure 1.88: Markdown editor for evaluations descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

1.7.6.3 Category

Each prediction has associated a **category** taken from model used to create it. Categories are useful to classify predictions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 1.2.

²¹https://en.wikipedia.org/wiki/Markdown

Category	
Aerospace and Defense	
Automotive, Engineering and Manufacturing	
Banking and Finance	
Chemical and Pharmaceutical	
Consumer and Retail	
Demographics and Surveys	
Energy, Oil and Gas	
Fraud and Crime	
Healthcare	
Higher Education and Scientific Research	
Human Resources and Psychology	
Insurance	
Law and Order	
Media, Marketing and Advertising	
Miscellaneous	
Physical, Earth and Life Sciences	
Professional Services	
Public Sector and Nonprofit	
Sports and Games	
Technology and Communications	
Transportation and Logistics	
Travel and Leisure	
Uncategorized	
Utilities	

Table 1.2: Categories used to classify predictions by BigML

1.7.6.4 Tags

A prediction can also have a number of **tags** associated with it that can help to retrieve it via BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the model use to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

1.7.7 Model Predictions Privacy

The link displayed in the **privacy panel** is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared from your BigML Dashboard by sharing a link, as you can do with other resources.

ources	Datasets	Mode	els 🕶 🛛	Clusters	Anomalies	Associations	Predictions	Tasks	Scrip	ts 👻
	ılî	• • Q 	••••		Bate	ch Centroid of	iris dataset		G -	6
DETAILS										ਂ
INFO										ਂ
PRIVACY										۲
Private lin	k Only yo	u can view	this batch	ncentroid log	gged into your a	ccount.				
				6	https://tropo.	dev.bigml.com/da	shboard/batchcen	troid/5739fe37b95b3903dc000037		

Figure 1.89: Private link of a prediction

1.7.8 Moving Model Predictions to Another Project

When you create a prediction it will be assigned to the same **project** where the original model is located. You cannot move predictions between projects as you do with other resources.

1.7.9 Stopping Models Predictions

Single predictions are synchronous resources, so you cannot cancel them during the creation since you get the result immediately.

Batch predictions are asynchronous resources, so you can stop the creation before the task is finished. You can use the DELETE option from the 1-click action menu (Figure 1.90) or from the pop up menu on the prediction list view. (See Figure 1.91.) You can see in Figure 1.91 that the objective field column has the label PROCESSING to indicate the batch prediction is still in progress. If you stop the prediction during its creation, you will not be able to resume the same task again, so if you want to create the same prediction, you will have to re-start a new task.



Figure 1.90: Stop prediction from the 1-click menu

Sources	Datasets Models ▼ Clusters Anor	malies Ass	sociations Predictions	Fasks	So	cripts 👻
ŝ	X X E E		Predictions			Q,
ជា	Name	\$	Objective	Prediction	٥	# <
к <mark>о</mark>	Prediction for Species	8	Species	Iris-versicolor		5m
68	Prediction for Surv		Survived	1		5m 1w
6 0	Prediction for ratins		rating	3.57752		6m

Figure 1.91: Stop prediction from the predictions list view

1.7.10 Deleting Model Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the 1-click action menu (see Figure 1.92) or using the pop up menu on the predictions list view (see Figure 1.93.)

a S° in ≴	Prediction	for Diabe	etes	5 -	0
	Diabetes:	FALCE	PREDICT QUESTION BY QUESTION		
	Diabetes:	FALSE	DELETE PREDICTION		
	Missing strategy:	8		All input fields:	V
Glucose	29.21%	вмі	-	22.96%	۲
0	248	0	83	.88	9.2

Figure 1.92: Delete prediction from the 1-click menu

Sources	Datasets M	lodels 🔻	Clusters	Anomalies	Ass	ociations	Predictions	Та	sks	Scripts	(NEW)
វា	X.	Ŧ	Ē			Predictio	ns				0
វា	Name				\$	Objective		٥	Prediction	\$ ÷	\$
Кî.	Prediction for Spec	cies			8	Species			Iris-versicolor		5m
é 🖧	Prediction for Survi		EW DETAILS			Survived			1	51	m 1w
5 ⁰	Prediction for ratin		LETE PREDICTI	ON	_	rating			3.57752		6m

Figure 1.93: Delete prediction from popu up menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted and there is no way you (or even the IT folks at BigML) can retrieve it.



Figure 1.94: Delete prediction confirmation

1.8 Consuming Models

In the previous sections, we have described how you can create BigML models, configure them, use them to make predictions, and more. This section will introduce a number of BigML features that enable interesting ways of taking advantages of BigML models: exporting them locally, and using them programmatically via the BigML REST API and Bindings.

1.8.1 Exporting and Downloading Models

You can export your model in a variety of programming languages, in PMML or in Excel format. Just click on the download icon in the top menu and select your preferred option.

The main goal of downloading your model in a programming language is to make local predictions faster and at no cost. (See Subsection 1.7.5.2.)



Figure 1.95: Download your model

If you are interested in exporting only the rules of a particular branch, you click on the branch leaf and press SHIFT from your keyboard. An icon to export the branch rules will appear below the prediction path. (See Figure 1.96.) Release the frozen view by pressing ESCAPE.



Figure 1.96: Download branch rules

1.8.2 Using Models Via the BigML API

Models have full citizenship in the BigML API. This means you can programmatically create, update, list, delete, and use them for predictions. For example, this is how you can create a model from the

command line with custom values for a few available arguments. This will require you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

For more information on using models through the BigML API, please refer to model REST API documentation.

1.8.3 Using Models Via the BigML Bindings

BigML bindings provide a convenient way to access BigML REST API from your language of choice. They offer a higher-level view of BigML Machine Learning resources and algorithms in a number of languages, including Python, Node.js, Java, Swift, and Objective-C. For example, this is how you can create a model in Python using BigML bindings:

For more information on using models through the BigML API, please refer to BigML bindings documentation.

1.9 Model Limits

BigML imposes a few limits on the characteristics of a model that it can handle:

- Fields: there is no enforced limit to the number of fields that can be present in a model.
- Instances: there is no enforced limit to the number of instances that can be handled.
- Classes: a maximum number of 1,000 distinct classes per field is allowed.
- **Term-tokens**: BigML can handle up to 1,000 tokens total. In case multiple text fields are defined, then the token limit per field is divided by the number of text fields.
- Term-full terms: BigML can handle up to 256 characters total.
- Items: a maximum of 10,000 items per field is allowed.
- Node threshold: BigML supports a value between 3 and 2,000.

1.10 Descriptive Information

Each model has an associated **name**, **description**, **category**, and **tags**. A brief description follows for each concept. In Figure 1.97, you can see the possibilities that the MORE INFO menu option gives to edit them.



Figure 1.97: Panel to edit a model's name, category, description and tags

1.10.1 Model Name

Each model has an associated **name** that is displayed on the list view and also on the top bar of a model view. Model names are indexed to be used in searches. When you create a model, by default, it gets the name of the dataset used to create it. The name of a model cannot be longer than **256** characters. There is no restriction on the characters that can be used in a model name. More than one model can have the same name even within the same project, since they are automatically assigned unique internal identifiers.

1.10.2 Description

Each model also takes the description from the dataset used to create it. Having a **description** can be very useful for documenting your Machine Learning projects. Descriptions can be written using plain text and also markdown²². BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 1.98.)

²²https://en.wikipedia.org/wiki/Markdown

Edit description	×
You can add formatting and links using a simple markdown language: B $I \coloneqq 60$	
Write your **description** here	
Description:	
Write your description here	
Cancel	Update

Figure 1.98: Markdown editor for model descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

1.10.3 Category

Each model has an associated category taken from the dataset used to create it. Categories are useful to classify models according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A model category must be one of the 24 categories listed on table Table 1.3.

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 1.3: Categories used to classify models by BigML

1.10.4 Tags

A model can also have a number of **tags** associated with it that can help to retrieve it via BigML API or to provide models with some extra information. Models inherit the tags from the dataset used to create them.

Each tag is limited to a maximum of 128 characters. Each model can have up to 32 different tags.

1.10.5 Counters

For each model, BigML also stores a number of counters to track the number of other resources that have been created using the model. In the model view, you can see a menu option that displays these counters. It also allows you to quickly jump to all the resources of one type that have been created with this model as shown in Figure 1.99.



Figure 1.99: Menu option to quickly access to resources created with a model

1.11 Models Privacy

In this section, you will learn about the privacy options that BigML provides for models, including how to share a resource privately, or make it public through the BigML BigML Gallery.

Privacy options for a BigML model can be defined in the MORE INFO menu option, displayed in Figure 1.100. There are three levels of privacy for BigML models:

- **Private**: only accessible by authorized users (the owner and those who have been granted access).
- **Shared**: by enabling the **secret link** you will get two different links to share your model. The first one is a sharing link that you can copy and send to others so they can visualize and interact with your model. The second one is a link to embed your model directly on your web page.
- **Public**: accessible and clonable as private resources by any user. Public resources are listed in BigML Gallery. (See Section 1.12).

	Sources [Datasets	Models -	Clusters	Anomalies	Associations	Predictions	Tasks				Scripts -	In
	A 54	ılî	617)	•	S .w	vines dataset'	s model		Ê	Ŷ	₲.	(≡) [/] → () ●	
	DETAILS											♥	
	INFO											\odot	
Private	PRIVACY											۲	
	- Private link	Only you	can view this mo	odel logged int	to your account.								
_					https://tropo.d	ev.bigml.com/das	hboard/model/5	72789a4b95b3	9287a0008	135			
hared	Secret link	Only peop	le with the secr	et link can view	v this model.								
		-	3	P	https://bigml.	com/shared/mod	el/1M9ERVbxPCI	Pi1qaSZqQT1	/wGwZE				
				;</td <td><iframe fra<="" src="</td><td>https://bigml.com</td><td>/embedded/mod</td><th>del/1M9ERVbx</th><th>PCPi1qaS</th><td>ZqQT1Mv</td><td>'GwZE" td=""><td>L</td><td></td></iframe></td>	<iframe fra<="" src="</td><td>https://bigml.com</td><td>/embedded/mod</td><th>del/1M9ERVbx</th><th>PCPi1qaS</th><td>ZqQT1Mv</td><td>'GwZE" td=""><td>L</td><td></td></iframe>	L							
	- Gallery link	Everybod	y will see your n	nodel if you sh	are it in BigML's g	jallery.							
Public	•	2		2	Not shared								
FUDIC			1										

Figure 1.100: Models privacy options within the More Info menu option

1.12 Models in the BigML Gallery

This section will introduce the BigML BigML Gallery, which provides a marketplace for Machine Learning models and workflows running on the BigML platform. Models and workflows published in the BigML Gallery may have a price or be free.

Using the BigML Gallery, you can **publish** your models so they are available to other users, or you can **clone** models that other users made public.

1.12.1 Publishing Models in the Gallery

You can make your model public for other BigML users. To accomplish this, publish your model in the BigML Gallery following these steps:

- 1. Provide a **description** for your model, which is mandatory for models in the BigML Gallery. We recommend you also assign a proper name, category, and tags to the model. (See Section 1.10 to learn how to update your model's descriptive information.)
- 2. Choose how to publish your model, either as a Black Box or a White Box model:
 - Black Box models allow BigML users to make predictions with it, but they are not available to purchase or clone. You can publish a Black Box model by clicking the black lock icon as shown in Figure 1.101.

ources Datasets Models - Clu	isters Anomalies Associations Predictions Tasks	Scripts 🗸
	Arrythmia diagnosis dataset's model	ૐ- (≡)∕- 0
DETAILS		$\overline{\mathbf{S}}$
INFO		$\overline{\mathbf{v}}$
PRIVACY		۲
Private link Only you can view this model log	ged into your account. https://bigml.com/dashboard/model/5746e49628eb3e03b50002ab	
Secret link Only people with the secret link of	an view this model.	
	Not shared	
Gallery link Everybody will see your model if	you share it in BigML's gallery.	
Black Box	Not shared	

Figure 1.101: Black Box models let other BigML users make predictions

A modal window will automatically appear asking for confirmation. Decide whether to share your model for predictions for free, or sell it and obtain benefits per prediction. Set the price you consider appropriate by just moving the price slider and click Update. (See Figure 1.102.)

Are you sure that you want to Other BigML users will be able to u not be able to visualize or downloa	se your model to make predictions , but they wil
Credits per prediction:	Earnings:
	\$1 per 57 predictions
This is the number of credits that other users will consume to make a prediction with your model.	Please email us at gallery@bigml.com if you want to price predictions with your model higher.

Figure 1.102: Set a price for other users to make predictions with your model

• White Box models allow users to clone or purchase the full model. Click the white lock icon to share your model as White Box. (See Figure 1.103.)

Sources Dataset	s Models -	Clusters Anomalie:	s Associations	Predictions	Tasks			Scripts -
a s² ın²	6in) - ••••	Arrythr	nia diagnosis da	ataset's mode	ı [i 🗘	∽.	(≡)∕ <u>~</u> () _
DETAILS								◙
INFO								$\overline{\mathbf{O}}$
PRIVACY								۲
Private link Only y	you can view this mode	l logged into your account						
		https://bigm	.com/dashboard/mo	del/5746e49628eb	3e03b50002ab			
Secret link Only p	eople with the secret li	nk can view this model.						
G	<u> </u>	Not shared						
Gallery link Every	body will see your mod	lel if you share it in BigML's	s gallery.					
A 2		Not shared						
	White Box							

Figure 1.103: White Box models let other users to clone or purchase the full model

A modal window will automatically appear asking for confirmation. Decide whether to share your model for free, or charge a price either by cloning the full model or by making predictions with your model. Set the prices you consider appropriate by just moving the price slider and click Update. (See Figure 1.104.)

rivacy and price settings	
	make this Model "White Box"?
they be able to use your model to use your would be your your would be your your would be your your would be your your your your your your your your	risualize, download and clone your model. Also make predictions.
Model price:	Earnings:
This is the price that other BigML users must	\$0.00 per Model Please email us at gallery@bigml.com if you want to
pay to clone your model.	price your model higher.
Credits per prediction:	Earnings:
	\$0
This is the number of credits that other users will consume to make a prediction with your model.	Please email us at gallery@bigml.com if you want to price predictions with your model higher.
	Cancel Update

Figure 1.104: Set a price for other users to clone your model and make predictions with it

3. After publishing your model, the gallery link automatically appears in the privacy panel and the

status changes from "Private" to "Black Box" or "White Box", depending on your choice. You can change the set price anytime by clicking on the edit icon. (See Figure 1.105.)



Figure 1.105: Public status changed to Black Box and the gallery link is available

You can only publish a model when the model is yours. If you are using a model previously cloned from another user, BigML will display a warning stating you cannot share that model or sell it. (See Figure 1.106.)

Sources	Datasets	Models -	Clusters	Anomalies	Associations	Predictions	Tasks	_	S	cripts 🔻
- K	ılî	⁶ 0 - ● • • •			Wine sales m	odel		â 🗘	(≣)	NEW ()
DETAILS										◙
INFO										$\overline{\mathbf{O}}$
PRIVACY										۲
Secret lin	k Only peop	ole with the secret			om/dashboard/mo	del/576030bde00	3ea499a001a1	18		
				This mode	l is cloned and priv t share or sell it.	ate.				
Gallery lin	k Everyboo	dy will see your mo	odel if you sha	e it in BigML's g	allery.					
				Not shared						

Figure 1.106: A cloned model cannot be shared or sold

1.12.2 Cloning Models From Gallery

BigML lets you use models that are public in BigML Gallery. These models are publicly available because other users have shared them. Some of the models available are free of charge and others have a specific cost. The owner of the model decides its cost. (See Subsection 1.12.1 for more details on how to share or sell your models.)

1. To import a model from the Gallery into your Dashboard, first you need to clone it. The link that gives you access to BigML public Gallery is on the very top menu on the left. (See Figure 1.107.)

			💱 🖌 0 tasks	S	Max. Tasks: 2	Max. Size: 64 MB		Log Out
	ES GALLERY -	LABS			ALFRED	WHAT'S NEW	DEVELOPERS	Dashboard
PUB	LIC RED'S	PROJECT: All			• /		PROD (DEV

Figure 1.107: Access to BigML Gallery

2. Select "Models" on the top menu. Then click the model you are interested in. (See Figure 1.108.) Clone it by clicking the Buy label. If the model is free of charge, click the Free label, which changes to Buy when you mouse over it, but actually BigML will not charge you anything.



Figure 1.108: BigML public gallery

 If the dataset that was used to create the model is also available in the gallery, a modal window (see Figure 1.109) will be displayed asking you if you want to add it to your current purchase. Click the Yes button to confirm you want to clone it along with the model; No to just buy the model.

	Do you want add it to your current purchase?
	This model's dataset is also available for purchase for:
etsy.com 58K shops	
Stats from etsy.com shops: items, followers, admirers, feedback, open date, sales,	
Source	
etsy.com import.io shops ecommerce crawler	
8.56 MB 9 fields / 58093 instances	

Figure 1.109: Modal window to confirm you want to buy the model's dataset

4. A modal window (see Figure 1.110) will be displayed asking you for confirmation.

SE PUBLIC - A MODELO		
Clone this mo	You are about to clone this model	Cione
•	ounoch	
Taxt Classification Model	credit-d's dataset model	Loan Risk
Text Classification Model	credit-g's dataset model	Loan Risk bigml
Text Classification Model bernt Model with word counts for about 47,000 text documents, of which roughly 32,000 are novels,	Credit-g's dataset model	Loan Risk Lignt Model that will predict the quality of risk of a loan application. This dataset was initially
Text Classification Model bigml Model with word counts for about 47,000 text	credit-g's dataset model	Loan Risk Lignt Model that will predict the quality of risk of a loan

Figure 1.110: Modal window to confirm you want to clone this model

5. Your new model goes directly to your BigML Dashboard. Notice that any task performed from a cloned model from BigML Gallery is free of charge, no matter the size of the task to be performed.

1.13 Moving Models to Another Project

When you create a model it will be assigned to the same project where the original dataset used to create it belongs to.

Models can only be assigned to a single project. However, you can move models between projects. The menu option to do this can be found in two places:

1. In the model detail view, within the 1-click actions menu (see Figure 1.111).



Figure 1.111: 1-click menu option to move models

2. In the model list view, within the pop up menu (see on Figure 1.112).

Sources	s Datasets	Models -	Clusters	Anoma	lies	Associations	Predictions	Tasks			Scri	pts + 🗞
						Model	s					Q,
III	Name			\$	Туре	Objective	\$	# \$	Ê ≎	* \$	\$	○ e ●○ ≎
ıl <mark>e</mark>	iris.csv dataset's	model		8	00	species		1m3w	4.5 KB	2	1	1
ıl <mark>e</mark>	113.0370	PREDICT BY QU	ESTION			species		1m 3w	4.5 KB			
ıl	iris.csv d	PREDICT CREATE BATCH	PREDICTION		00	species		1m 3w	4.5 KB			
ılî		EVALUATE			00	species		1m 3w	4.5 KB			
ılî	iris.csv d	VIEW DETAILS			00	species		1m 3w	4.5 KB			
ιP	iric cev d	DELETE MODEL				NEW PROJECT		Зw	4.5 KB			
ılî	iris.csv dataset's	model		-	PROJ			3w	4.5 KB			
ιP	diabetes dataset	's model			Р	ROJECT 2		3w	6.7 KB			
ılî	iris.csv dataset's	model			TI DU	EST-1		3w	4.5 KB			

Figure 1.112: Pop up menu option to move models

1.14 Stopping Model Creation

You can also stop a model's creation process before the task is finished from the 1-click menu (Figure 1.113) or from the pop up menu in the models list view (Figure 1.114).

species: ? SUPPORT 0.66 % Predictions for: species Ei PREDICT BATCH PREDICTION EXALUATE EXALUATE MOVE TO	S° II° ⁶ 0- ••••	iris.csv dataset's mode	el 📋 🚯 🚳	(≡)∕- 0
SUPPORT 0.66 % 33.35 % Predictions for: species EVALUATE Structure Support EVALUATE Structure Support EVALUATE		species: ?	PREDICT QUESTION BY QUESTION	G
D.66 % 33.35 % Predictions for: species CALUATE			PREDICT	
EVALUATE CONTRACTOR OF CONTRACTOR OF CONTRACTO		Predictions for: species	BATCH PREDICTION	
			EVALUATE	
🧶 📫 мочето				
	•		MOVETO	

Figure 1.113: 1-click menu option to stop a model's creation

Source	s Datasets	Models -	Clusters	Anoma	alies	Ass	ociations	Prediction	is Tasks			Scri	pts 👻
							Models						0
III	Name			\$	Туре	\$ 0	Objective	\$	÷	₿ ≎	\$	\$	<u>○</u>
ıl <mark>î</mark>	Arrythmia diag	jnosis dataset's r	nodel	۲	00	A	rrhythmia		2w 5d	536.2 KB	0	0	1
ılî	Arrythmi: 🔳		JESTION		.0	A	rrhythmia		2w 5d	536.2 KB			1
ıl <mark>e</mark>	Arrythmi		PREDICTION		00	A	rrhythmia		2w 5d	536.2 KB			1
ılî	Arrythmi:				00	A	rrhythmia		2w 5d	536.2 KB			1
II <mark>Q</mark>	Arrythmi					A	rrhythmia		2w 5d	536.2 KB			1
ιR	Arrythmi	-	-			A	rrhythmia		2w 5d	536.2 KB			1

Figure 1.114: Pop up menu option to stop a model's creation

A modal window (Figure 1.115) will be displayed asking you for confirmation.

vou delete this	nodel, you will no longer have ac	cess to it and you will r	need to recreate it
rom your dataset	.,		

Figure 1.115: Menu option to stop a model's creation

Note: if you stop the model during its creation, you will not be able to resume the same task. If you want to create the same model, you will have to start a new task.

1.15 Deleting Models

You can delete your models using two different options:

• From the model view, using the 1-click action menu (Figure 1.116).

Sources Datasets Models Cluste	rs Anomalies Associations Predic		Scripts
<u></u>	species: ?	PREDICT QUESTION BY QUESTION	e
SUPPORT 0.66 %	Predictions for: species	BATCH PREDICTION	
1	•	MOVE TO	
•			

Figure 1.116: Menu option to delete a model

• Using the pop up menu on the model list view (Figure 1.117).

Source	s Datasets	Models -	Clusters	Anoma	lies	As	sociations	Predictions	Tasks			Scri	ots 🔻
							Models						0
III	Name			\$	Туре	\$	Objective	\$	₩ ≎	₿ ≎	\$	\$	<u>○</u> <u> </u>
ıl <mark>î</mark>	Arrythmia diag	nosis dataset's mo	del		o0		Arrhythmia		2w 5d	536.2 KB	0	0	1
ılî	Arrythmi 😫		STION		00		Arrhythmia		2w 5d	536.2 KB			1
ılî	Arrythmi:	PREDICT CREATE BATCH PI	REDICTION		00		Arrhythmia		2w 5d	536.2 KB			1
ılı	Arrythmi:	EVALUATE			00		Arrhythmia		2w 5d	536.2 KB			1
ılî	Arrythmi:				00		Arrhythmia		2w 5d	536.2 KB			1
ılî	Arrythmi	MOVE TO)	00		Arrhythmia		2w 5d	536.2 KB			1

Figure 1.117: Model deletion pop up menu option

A modal window (see Figure 1.118) will be displayed asking you for confirmation. Once you delete a model, it is deleted permanently and there is no way you (or even the IT folks at BigML) can retrieve it.

Are you	ı sure you wa	int to delet	te this mode	el?	
lf you delet from your c	e this model, you will r atasets.	10 longer have ac	cess to it and you	will need to recr	reate it
			Cance		elete

Figure 1.118: Model deletion modal window

1.16 Takeaways

This chapter explained models in detail. We conclude it with a list of key points:

- BigML models are human friendly as opposed to many other Machine Learning models. They provide a set of rules organized in a tree structure that are easy to understand for non-experts.
- You can use BigML models to solve classification and regression problems.
- BigML models support any type of fields as input fields (categorical, numeric, date and time, text, and items fields).
- BigML Models are not affected by uninformative or redundant fields.
- Normalization is not needed.
- To build a BigML model you just need a dataset. (See Figure 1.119).
- A BigML model can be an input to an evaluation, to a prediction, or to a batch prediction. (See Figure 1.119).
- A BigML model can be the output of a cluster [8]. (See Figure 1.119).
- You can create a BigML model with just 1-click or configure it as you wish. BigML models are easy
 to tune without having to configure difficult parameters.
- You can also create models using BigML REST API or the BigML bindings for your language of choice.
- If you do not specify any objective field, BigML will use the last valid field in your dataset.
- You can choose three different pruning strategies when building your BigML model to avoid overfitting: smart pruning, statistical pruning, or no statistical pruning.
- By default, BigML models do not consider missing values when choosing splitting rules, but you
 can explicitly include them.
- You can set the maximum number of nodes you want for your BigML model. Greater number of nodes will grow more complex trees that will perform better with the training data at the expense of generalization.
- To deal with imbalanced datasets, BigML provides three different options to assign specific weight your instances: balance objective, objective weights, weight field.
- For classification problems, the confidence is a measure of the model's certainty when predicting a class at a certain node.
- For regression problems, the expected error is a measure of the expected error at a node.
- You can visualizeBigML models in an interactive decision tree's structure or with the Sunburst view.
- BigML provides a summarized view of your model, including: the data distribution, prediction distribution, field importance, and the rules summary.
- You can easily see which fields in your dataset have more impact on predictions by clicking in the model's summary report.
- You need to evaluate your model's performance using data that the model has not seen before. Evaluating a model is a key step to understand if that model is satisfactory or needs training adjustments. In the latter case, you can use different creation options until you achieve the desired evaluation results.
- You can download your model to your preferred programming language to use it in your local environment, and make predictions faster at no cost.
- Once you get a satisfactory model, you can make single or batch predictions using your model.

- BigML provides local predictions from the BigML Dashboard for single instance predictions. Local
 predictions allow you to get a real-time prediction without consuming any credits or requiring an
 internet connection.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. For batch predictions, you always get a CSV file and an optional output dataset.
- You can furnish your model with descriptive information (name, description, tags, and category).
- You can clone an existing model from BigML Gallery.
- You can share your model in the BigML Gallery as Black Box or White Box, so other BigML users can clone your model and make predictions with it.
- You can stop the model's creation before the task is finished.
- · You can permanently delete a model.



Figure 1.119: Model Workflows



Ensembles

2.1 Introduction

There are multiple Machine Learning problems that can be solved using supervised Machine Learning techniques. Some of these problems require to predict an output variable (objective field) given a number of input variables (input fields). These problems can be divided into classification and regression depending on whether you need to predict a category (label or class) or a continuous value (a real number), respectively. To learn more about concrete use cases for both problems refer to Section 1.1.

However, most of these problems cannot be solved with a single model. One of the pitfalls of Machine Learning is that the algorithm has the potential to overfit¹ your data, so its performance on your training data is very good, but it does not generalize well to new data, which makes single tree models poorer predictive models. Ensembles avoid this disadvantage.

An ensemble is a collection of multiple decision trees which are combined to create a stronger model with better predictive performance. An ensemble of models built on samples of the data can become a powerful predictor by averaging away the errors of each individual model. Generally, ensembles perform better than a single decision tree because they are less sensitive to outliers in your training data, which helps them mitigate the risk of overfitting and generalize better when applied to new data.

Depending on the nature of your data and the specific values for the ensemble parameters, you can significantly boost the performance over using a single model. For a technical explanation of why ensembles perform better, please see this tutorial paper² from our Chief Scientist, Tom Dietterich [2].

BigML currently provides three types of ensembles:

- **Bagging** (also known as Bootstrap Aggregating): this algorithm builds each single model composing the ensemble from a random subset of the dataset instances. By default the samples are taken using a rate of 100% with replacement (this is explained in Subsection 2.4.9). While this is a simple strategy, it often outperforms more complex strategies. Read more about Bagging³.
- **Random Decision Forests**: similar to Bagging but it adds an additional element of randomness by choosing a random subset of features at each tree split. Read more about Random Decision Forests⁴.
- **Boosted Trees**: (or gradient boosted trees) this algorithm sequentially builds a set of weak learners and then combines their outputs in an additive manner to get a final prediction. In every boosting iteration, each single model tries to correct the errors made in the previous iteration by optimizing a loss function. Read more about Gradient Boosting⁵.

¹https://en.wikipedia.org/wiki/Overfitting

 $^{^{2} \}texttt{http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf}$

³https://en.wikipedia.org/wiki/Bootstrap_aggregating

⁴https://en.wikipedia.org/wiki/Random_forest

⁵https://en.wikipedia.org/wiki/Gradient_tree_boosting

See Subsection 2.4.3 for an explanation of each algorithm.

This chapter contains comprehensive description of BigML's ensembles including how they can be created with 1-click (Section 2.3), all configuration options available (Section 2.4), and the visualization provided by BigML (Section 2.5). Once you create an ensemble, you can get a report for each field importance (see Subsection 2.2.2), and a heatmap chart, known as Partial Dependence Plot (Section 2.5), to visualize the impact of your input fields on predictions. See Section 2.6 for an explanation of how ensembles can be used to make predictions. Moreover, you can also export your ensembles in different formats to make local predictions faster at no cost (Subsection 2.7.1), move your ensembles to another project (Section 2.11), or delete them permanently (Section 2.13). The process to evaluate your ensemble's predictive performance in BigML is explained in a different chapter (Chapter 7).

In BigML, the third tab of the main menu of your Dashboard allows you to list all your available ensembles. In the ensemble list view (Figure 2.1), you can see, for each ensemble, the Dataset it was created from, the ensemble's Name, Type (either classification or regression), Objective (objective field name), Age (time elapsed since it was created), Size, and number of predictions, batch predictions, or evaluations that have been created using that ensemble. The SEARCH menu option in the top right corner of the ensemble list view allows you to search your models by name.

Source	s Datasets	Ensembles -	Clusters	Anomalies	Associations	Predictio	ns Task	3	_	Scri	pts 👻
				Ē.	Ensembles						Q
ıll	Name			🗘 Type 🗘	Objective	٥	# \$	Å •	\$	(
ıl <mark>o</mark>	Loan risk's ense	mble		00	class		0min	919.8 KB	0	0	0
ıl <mark>e</mark>	MQIC Patient Da	ita 's ensemble		\checkmark	BMI		Omin	17.7 MB			
ıl <mark>e</mark>	cars dataset dat	aset's ensemble		\checkmark	Zipcode		1min	28.4 KB			
ıl <mark>e</mark>	train_rossman_s	store_info_open data	set - sam	00	Promo2Running		3min	130.5 MB			
ılî ⁰	Diabetes Diagno	sis dataset's enseml	ble	00	Diabetes		1w 6d	267.5 KB			
ıl <mark>e</mark>	wine-sales datas	set's ensemble		木	Total Sales		4m	519.2 KB			1
ıl <mark>e</mark>	Churn in the Tele	com Industry's datas	set ensem	00	Churn		5m 4w	2.6 MB			
Show 10	ensembles			1 t	o 7 of 7 ensembles					K < 1	> >

Figure 2.1: Ensembles list view

By default, when you first create an account at BigML, or every time that you start a new project, your list view for ensembles will be empty. (See Figure 2.2.)

Sources Data	asets Ensembles -	Clusters	Anomalies	Associations	Predictions	Tasks			Scr	ipts 🔻
			£'s	Ensembles						Q,
Name			≎ Type ≎	Objective	÷ 🛗	\$	Â	\$ \$	\$	
			ľ	lo ensembles						
Show 10 🛊 enser	nbles		No	ensembles found					IC (> >

Figure 2.2: Empty Dashboard ensembles view

Finally, in Figure 2.3 you can see the icon used to represent a model in BigML.



Figure 2.3: Ensemble icon

2.2 Understanding Ensembles

In this section, we are going to describe a few internal details about ensembles and how BigML implements them. Specifically, since ensembles are based on BigML models, all the information provided in Section 1.2 also applies to ensembles, unless overridden here.

BigML grows ensembles in a very similar way to how it grows simple models (see Section 1.2). In particular, BigML does not use dataset streaming (see Subsection 1.2.1) for ensembles, thus requiring the entire dataset to be loaded into memory. This choice is motivated by the intrinsic behavior of growing ensembles, which aggressively uses sampling to be able to create multiple significant models from the same dataset.

2.2.1 Decision Forests Versus Boosted Trees

BigML offers three types of algorithms, **Bagging**, **Random Decision Forests** (which are included under the same ensemble type in BigML called **Decision Forests**), and **Boosted Trees**. The following subsections contain some technical details of the main commonalities and differences for these methods. To read about the situations under which is better to use one method or another please refer to Subsection 2.4.3.

All ensemble methods have in common that they are composed by **several single trees** and their **output is combined** to yield a final prediction. The ensemble always has a better performance than each of the individual learners it is composed of.

The main difference between Decision Forests and Boosted Trees is the way single trees are grown and the way their predictions are combined to get the final ensemble prediction. See the following subsections for a detailed expalanation.

2.2.1.1 Single trees

Each single tree in Decision Forests tries to predict the **objective field** using certain level of randomess, either by selecting a random percentage of the dataset instances (Bagging) and/or by selecting a random subset of the input fields at each split (Random Decision Forests). However, in Boosted Trees, each single learner, instead of predicting the objective field, tries to learn from the mistakes made by the previous model by **fitting a gradient** step towards minimizing the error of the previous classifier. For regression problems, the error is the usual squared error (the squared difference between the true objectives and the current prediction). For **classification** problems, a tree is trained for each class in each iteration. The scores from the trees are normalized using the softmax function⁶ to obtain a probability distribution over classes given a datapoint. The error is the difference between this distribution and the "true distribution" over classes for the datapoints, which is one for the correct class and zero for all others.

Another characteristic for Boosted Trees is that they have a **weight** associated with each tree which measures their importance to calculate the final ensemble prediction. The weights are chosen via a line search⁷, where possible weights are evaluated on a group of test points to find a weight that is near-optimal. Test points for the line search are randomly selected from the training data. Lastly, the weights are multiplied by the **learning rate** (see Subsection 2.4.7.2) to calculate the final weight. Read more about choosing weights using line search here⁸.

⁶https://en.wikipedia.org/wiki/Softmax_function

⁷https://en.wikipedia.org/wiki/Line_search

⁸https://en.wikipedia.org/wiki/Gradient_boosting#Gradient_tree_boosting

2.2.1.2 Ensemble predictions: confidence, probability and expected error

For **Decision Forests**, single tree predictions are **averaged** to get a final prediction. The same quality measures obtained when building a single model are returned for Decision Forest predictions: **confidence** and **probabilities**, for classification problems, and **expected error**, for regression problems. Find the calculations details for each measure in Subsection 1.2.6. For regression ensembles, all the single trees predictions are averaged to get a single prediction. For classification ensembles, all the per-class confidences and probabilities are averaged taking into account all the trees in the ensemble. The class with the highest confidence or probability is returned. There is an additional technique to calculate predictions for classification problems called "votes". It is based on the percentage of trees voting for each class in the ensemble to select the winner class. See Subsection 2.6.3.2 for a detailed explanation about **probabilities**, **confidences** and **votes** to calculate predictions.

For **Boosted Trees**, the single model predictions are **additive** rather than averaged. For boosted trees you only get the class **probabilities** in the case of classification ensembles, neither the confidence or the expected error can be calculated. For regression ensembles the final prediction is generated by calculating the sum of each tree prediction multiplied by its boosting **weight** (see the explanation for boosting weight in Subsection 2.2.1.1). The expected error cannot be calculated for Boosted Trees so there is not quality measure returned for regression problems. Predictions for classification ensembles are similar, but separate weighted sums are found for each objective class. The resulting vector of weighted sums is then transformed into **class probabilities** using the softmax function⁹. Hence, the probability for each of the classes in the objective field is returned to measure the prediction quality for boosting.

2.2.2 Field Importance

As with individual decision trees, the field importance for ensembles provides a measure of how important a data field is relative to the others. It is computed by taking a weighted average of how much each field reduces the predicted error of the tree at each split (more details in <u>Subsection 1.2.5</u>). For individual trees this measure can be misleading as it assumes that the tree structure is correct, but for ensembles it is a more meaningful measure. (See Figure 2.4).



Figure 2.4: Field importance for ensembles

To visualize the marginal contribution of a field in the ensemble predictions, BigML offers Partial Depen-

⁹https://en.wikipedia.org/wiki/Softmax_function

dence Plots. You can find a detailed explanation in Section 2.5.

Note: The concept of field importance is also used in prediction explanation for single predictions (See <u>Subsection 2.6.4.1.1</u>). But they are calculated differently. A field can be very important for the ensemble but insignificant for a given prediction.

2.2.3 Ensembles with Images

BigML ensembles do not take images as input directly, however, they can use image features as those fields are numeric.

BigML extracts image features at the source level. Image features are sets of numeric fields for each image. They can capture parts or patterns of an image, such as edges, colors and textures. For information about the image features, please refer to section Image Analysis of the Sources with the BigML Dashboard¹⁰[11].

Sources Dataset	ts Supervised - Un	supervised - Predictio	ns ▼ Tasks		WhizzML 🔫
8 7. 8.	••••	grape-straw	berry texture	เมือ	*** €** (≡)** ③
X	tabel	ABC × V		Search by name	*
Name	†↓ Туре	\uparrow_{\downarrow} Count \uparrow_{\downarrow}	Missing 1, Erro	ors † Histogram	
image_id	imag	e 70	0	0 🚵 🎑	🌆 🍋 🔛
filename	path	70	0	0	
label	ABC 160 image feature		0	0	
Show 10 V fields	√ ‼ 162 ⊠≣ 160	1 to 3 of 3	fields		

Figure 2.5: A dataset with images and image features

As shown in Figure 2.5, the example dataset has an image field *image_id*. It also has image features extracted from the images referenced by *image_id*. Image feature fields are hidden by default to reduce clutter. To show them, click on the icon "Click to show image features", which is next to the "Search by name" box. In Figure 2.6, the example dataset has 160 image feature fields, called *Wavelet subbands*.

¹⁰https://static.bigml.com/pdf/BigML_Sources.pdf

Sources Datasets Super	vised - Unsupervi	ised 👻 Predictions 👻 1	asks	v	VhizzML 🔻
8 1 .	0	grape-strawberry te	Click to hide image	features 🗤 🗞 - 🦽 -	(≡)⁵, 🚺
abel		ABC x v		Search by name	×
Name	†↓ Туре	the Count the Missing	tj. Errors	Histogram	
image_id	image	70	0 0	🎯 🌌 💟 🎕 🧕	
filename	path	70	0 0		
label	ABC	70	0 0		
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	dillin	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	dilligence and the second second	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	a sellar	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	lilin	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	and the second se	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	ululuu	
image_id.full image: wavelet leve	🖽 wavelet	70	0 0	when we have a second	
Show 10 🗸 fields 🖋 🛚 162	160	1 to 10 of 163 fields		K < 1 2 3	4 5 > >

Figure 2.6: A dataset with image feature fields shown

From image datasets like this, ensembles can be created and configured using the steps described in the following sections. All other operations including prediction, evaluation applies too.

2.3 Creating Ensembles with 1-Click

To create an ensemble in BigML you have two options: either the BigML unique **1-click** feature which provides a convenient way to quickly train an ensemble from a dataset, or **configuring** a number of options that give you fine control on how the ensemble is created. This section will guide you through the process of creating an ensemble with just 1-click.

By far, the easiest and quickest way to create an ensemble is using the 1-CLICK MODEL option that is available in the **1-click action menu**. (See Figure 2.7.)

Sources Datasets Models -				ctions	Tasks	Scripts	NEW)
a ılî 📴- ••••		Diabe	etes Diagnosis'	da		\$* 6 -	•
×					1-CLICK MODEL		× 4 alials
Name	≎ Туре	Count	Missing	Errors	CAN 1-CLICK ENSEMBLE		1-click
Pregnancies	123	614	0	0	1-CLICK CLUSTER		ensemble

Figure 2.7: 1-click ensemble from dataset detail view

Alternatively, you can select the 1-CLICK MODEL option in the 1-click action menu from the dataset list view. (See Figure 2.8.)

Sourc	es Datasets Mor	dels – Clusters Anomalies	Assoc	iations P	redictions	Tasks			Scr	ipts -
				Datasets						Q
9	Name		٥	# \$	₿ ≎	<u>6</u> 1 0	£0. 0	₩. ≎	***** ÷	₽ ≎
54	wines dataset		\otimes	3d 18h	724 bytes	4	0	0	0	0
s.	Unnamed remote sourc	1-CLICK MODEL		1w1d	4.5 KB	2				
54	Unnamed remote sourc	1-CLICK ENSEMBLE		1w1d	4.5 KB	1				
s°.	Unnamed remote sourc			1w1d	4.5 KB	1				
_	Unnamed remote sourc	1-CLICK ASSOCIATION		1w1d	4.5 KB	1				
5	Unnamed remote sourc	1-CLICK TRAINING TEST		1w 4d	4.5 KB	1				
54	Unnamed remote sourc	. *		2w1d	4.5 KB	1				
5	Unnamed remote source	моче то	>	Зw	4.5 KB	2				

Figure 2.8: 1-click ensemble from dataset list view

Both options build a new ensemble using default values for all available configuration options (see Section 2.4). This can give you a quick starting point to understand how your ensembles behaves and how it can be improved.

Creating an ensemble may take a variable time, depending on how big your dataset is, your subscription plan, etc. Once your ensemble is ready, it will be automatically displayed on your BigML Dashboard and you will be able to explore it through BigML visualizations (see Section 2.5), and using it for evaluations (see Chapter 7) or predictions (see Section 2.6.)

2.4 Ensemble Configuration Options

While 1-click creation (see Section 2.3) provides a convenient and easy way to create an ensemble from a dataset, there are cases when you want more control. This section will focus on the options that BigML offers to configure its internal algorithms for BigML ensembles.

You can set a number of parameters that affect the way BigML creates ensembles from a dataset. Such parameters can be grouped in two categories:

- Parameters that are permanently associated to the dataset, such as its objective field and preferred fields. Once you provide a value for a dataset's permanent parameters, they will be used as a default value for the creation of ensembles from that dataset.
- Parameters that only affect the ensemble that is currently being created and that you are expected to set each time, such as included/excluded fields, and a number of configuration options that are described below. The objective field can also be specified on a per-ensemble basis, if you do not want to tie it to the dataset as described above.

Set a dataset's permanent parameters by clicking on the edit button that is displayed when you hover on the dataset's fields. This opens a modal dialog where you can set some of the field properties (See Figure 2.9).

	Sources Datasets Models -	Clusters Anomalies Associations Pr	redictions Tas	sks Scripts -
	≙ 5 [°] 88°, ••••	iris dataset		°‰ - ∽ (≡) [™] ①
	×	Update field 000002 details		Q X
	Name		Errors	Histogram
	sepal length	Name	0	od huldhiddaa ar
Edit field	sepal width	Label	0	
	potal length		0	
Preferred	petal width	Description	0	
field	species	• • 1 •	0	
Objective	Show 10 ᅌ fields			
field		Save X Cancel		

Figure 2.9: Configure permanent parameter modal

Click on the preferred field button to make that field non-preferred.

Click on the objective field button to make that field the new objective field.

To access the configuration panel, select the CONFIGURE ENSEMBLE menu option located in the **configuration menu** of your dataset's detail view. (See Figure 2.10.)

Sources Datasets	Models - Clusters	Anomalies As	sociations Pre	dictions Tasks	Scripts 👻	
a 🔐 📴 •	•••	Diab	etes Diagnosis	' da	\$ _{2 -} (5)	
×				CONFIGURE MODEL	×	
Name	≎ Type	Count	Missing	CONFIGURE ENSEMBLE		Configure
				CONFIGURE CLUSTER		ensemble
Pregnancies	123	614	0	CONFIGURE ANOMALY		

Figure 2.10: Configure ensemble

When the configuration panel is displayed, you can:

- Select or deselect individual fields for them to be included in or excluded from the ensemble computation.
- Change the objective field used for the ensemble to be created.
- Manually configure a number of configuration options or automatically optimize these options.

Note: when the configuration panel is displayed, the edit is not visible, so you cannot set the dataset's permanent properties.

Configuration options are the same for ensembles as for models (Section 1.4) plus a few more: type of algorithm (Decision Forests —which includes bagging and random decision forests— and Boosted Trees), number of models, random candidates, and the Boosting parameters (number of iterations, early stopping, learning rate and setp out of bag). Sampling options are also important for the configuration of ensembles. (See Subsection 1.4.7).

You can find a detailed explanation of the configuration options below.

2.4.1 Objective Field

Also known as "target field", the objective field is the output variable you want to predict.

Select your objective field in BigML in either of two ways. Specify the objective field each time you create an ensemble from the configuration panel or set a field as the default objective for all ensembles by clicking the edit button and then the objective field button.

By default, BigML will use the last valid field in your dataset as objetive, with the exemption fields of type text and items that cannot be used as objective.

Sources Datasets	Supervised -	Unsupervi	NEV sed ▼	Predicti	ons 🔻 Tasks				WhizzML 🔻
	•••	Dia	betes	diagnos	is dataset Tra	aining (i	70%)	% . 3	(≡)⁵₊ 🕕
ENSEMBLE CONFIGURATI	ON								104
Objective field: Diabetes		ABC	v	Type: Decision	Forest	• •	umber of models:	Number of	iterations:
Advanced configura	ation								\odot
Ensemble name:									
Diabetes diagnosis data	set Training (7	'0%)'s ensem	nble				Reset	ិ៍ស្ត្រឺ Create	ensemble
							٩		×
Name	\$	Туре	٥	Count	Missing	Errors	Histogram		
Pregnancies		123		537	0	0	lin.	II	
Glucose		123		537	0	0			

Figure 2.11: Ensemble objective field

2.4.2 Automatic Optimization

You can turn on the **Automatic optimization** option so BigML will automatically tune the parameters of your ensemble (see Figure 2.12).

Sources Datasets Supervis	sed Unsupervised Predictions Tasks	WhizzML 🗸
רביים <mark>נו</mark> יי •••••	Sentiment	°‰_ ⊘∽ (≡)⊱ (i)
ENSEMBLE CONFIGURATION		рал 1967
Dbjective field: retweet_count 123	Automatic optimization Max. training tim 00:30:00	e: Ensemble candidates:
Type: Nu Decision Forest •	mber of models: Number of iterations:	
🏠 Advanced configuration		\odot
Ensemble name:		
Sentiment		Reset

Figure 2.12: Automatic optimization

The high number of possible combinations for parameter values makes it difficult to find the optimum configuration since the combinations that lead to a poor result outnumber the ones that result in a satisfying performance. Hand-tuning different configurations is a time-consuming process that requires a high level of expertise and intuition. To combat this problem, BigML offers first-class support for automatic ensemble parameter optimization.

Behind the scenes, BigML uses the same technology for ensemble parameter optimization as the one used for OptiML. If you want to know more about the technical details, please read the Chapter 2 of the document OptiML with the BigML Dashboard [10].

When you turn on the **Automatic optimization** option, all the ensemble parameters will be disabled (because they will be automatically optimized), except the **Missing splits** and the **Weights** parameters which you can manually configure (see Subsection 2.4.6.2 and Subsection 2.4.8).
ources Datasets	Supervised - Unsuperv	rised - Predictions -	Tasks	WhizzML
₽, <mark>8</mark> ,	•• Diabete	es diagnosis_weight_t	field Training (80%)	🎭 ు (≡)⁵⊷
SEMBLE CONFIGURATIO	N			
jective field:		Automatic optimization	Max. training time:	Ensemble candidates:
liabetes	ABC V		00:30:00 ©	128
ype: Decision Forest	Number of models:	Number of iteration	s: 7	
Advanced configurat	tion			
Tree:			The parame	eters will be automatically optimized 🙆
Missing splits:				MISSING SPLITS 0
Node threshold:	512			NODE THRESHOLD ? 512
Randomize:	Random candidates:			RANDOMIZE Disabled
Boosting:			The parame	eters will be automatically optimized \bigcirc
Weights:				۹
Select a weight option				WEIGHTS 1 No weights
Ensemble sampling:			The parame	eters will be automatically optimized 🕤

Figure 2.13: Configure the missing splits and weights for your ensemble

Note: there is a maximum of 256 trees per ensemble that will be tried out during the optimization process. If you think that your ensemble needs a higher number of models, you can manually configure it.

Since the optimization process can take some time, BigML offers two configurable parameters to limit the time to create the optimized ensemble: a training duration (see Subsection 2.4.2.1) and the ensemble candidates (see Subsection 2.4.2.2).

2.4.2.1 Training duration

The scale parameter to regulate the ensemble runtime. It's set as an integer from 1 to 10. It indicates the user preference for the amount of time they wish the optimization to take. The higher the number, the more time that users are willing to wait for possibly better ensemble performance. The lower the number, the faster that users wish the ensemble training to finish. The default value is set to 5.

The training duration is set in a scale. The actual training time depends on the dataset size, among other factors.

Sources Datasets	Supervised - Unsupervised ·	 Predictions - 	Tasks	WhizzML 🗸
₽ 		diabetes		Iề- ‰. (≡) [¢] - (〕
ENSEMBLE CONFIGURATION				PDF
Objective field: diabetes	ABC V	omatic optimization	Training duration:	Ensemble candidates:
Type: Decision Forest	Number of models:	Number of iterations: 64		
Advanced configuratio	n			\odot
Ensemble name: diabetes			Reset	ិត្តិ Create ensemble

Figure 2.14: Training duration

2.4.2.2 Ensemble candidates

The maximum number of different ensembles (i.e., ensembles using a unique configuration) to be trained and evaluated during the optimization process. The default numbre is 128 ensembles which is usually enough to find the best ensemble, but you can set it from 4 up to 200. Only the top-performing ensemble will be returned. If the training duration is very low (see Subsection 2.4.2.1) given the dataset size, it is possible that not all the ensemble candidates will be tried out.

Sources Datasets S	Supervised - Unsupervise	d 👻 Predictions 👻	Tasks	WhizzML 🗸
A 17. 80		diabetes		ılề- 🍫 (∋) [¢] ()
ENSEMBLE CONFIGURATION				POF
Objective field: diabetes	ABC V	utomatic optimization	Training duration:	Ensemble candidates:
Type: Decision Forest	Number of models:	Number of iterations: 64		
Advanced configuration	ı			\odot
Ensemble name:				
diabetes			Reset	ିର୍ଲ Create ensemble

Figure 2.15: Ensemble candidates

2.4.3 Type

This option allows you to choose between two methods to build your ensemble: Decision Forests and Boosting. (See Figure 2.16.) By selecting **Decision Forests** you can either build a Bagging or a Random Decision Forests ensemble. By default, Decision Forests take a random subset of instances from the dataset, which creates a Bagging ensemble. You can also add an additional element of randomness by choosing random features at each split (see Subsection 2.4.6.4) so you get a Random Decision Forest ensemble. By choosing **Boosted Trees** BigML builds gradient boosting trees. Read a technical description about Decision Forests and Boosted Trees in Subsection 2.2.1.

There is not an easy answer for the question of which method yields the best results. Usually you will have to try and test all options. However, depending on your dataset's characteristics, you can sometimes have an initial idea of which may perform better.

In Boosted Trees, the effect of additional trees is essentially an expansion of the hypothesis space in a way that it is not for Decision Forests. So if you expect the decision function to be very complex and you have a lot of data, boosting may work better than Decision Forests.

On the other hand, if you have a noisy domain, where overfitting is a concern, Decision Forests may be a better option than Boosted Trees, since boosting is more vulnerable to label noise.

Finally, if you have what you suspect is an "easily learnable" function, the additional power offered by Boosted Trees, or even Random Decision Forests, may not help; a more simple method, like Bagging or even models may perform better than the other two options.

Sources Datasets Supervi	sed - Unsupervised	NEW Prediction	NEW Is ▼ Tasks			w	hizzML 🔻
₽ ₽ . 8	Diabe	tes diagnosis	dataset Tra	ining (70%)		🎭 🦽 (i	≡) * _ (i)
ENSEMBLE CONFIGURATION							104
Objective field: Diabetes	ABC	Type: Decision Fo		Number of	of models: 10	Number of iter	ations: 10
Advanced configuration		Boosted	Trees				◙
Ensemble name: Diabetes diagnosis dataset Trai	ning (70%)'s ensemble	•		R	eset	ិត្តិ Create ense	emble
					Q		×
Name	\$ Туре	Count	Missing	Errors	Histogram		
Pregnancies	123	537	0	0	llu.	 	
Glucose	123	537	0	0			

Figure 2.16: Ensemble type: Decision Forests or Boosted Trees

2.4.4 Number of Models

This option allows you to configure the number of decision trees to create your ensembles when you select **Decision Forest** as your ensemble type (see Subsection 2.4.3). By default, the number of models is set to 10 and the maximum allowed is 1,000 trees. (See Figure 2.17.) For Boosted Trees the number of models will be determined by the number of iterations (see Subsection 2.4.5) with a maximum of 2,000 trees per ensemble.

Generally, increasing the number of trees will yield better results. Furthermore, there is no downside except higher computational time. The situations where more models are likely to provide the most improvements are those where the dataset is not very large (e.g., in the thousands of instances or less), the data is very noisy, and (in the case of Random Decision Forests) when there are many correlated features that are all somewhat useful.

Take into account that each additional model tends to deliver less marginal improvement, so if the difference between nine and ten models is very small, it is very unlikely that an eleventh model will make a big difference.

Sources Datasets S	Supervised 👻 Unsu	npervised ▼ Pr	edictions • Tas	ks		WhizzML 🔻
	• Il	Diabetes diag	nosis dataset	Training (70%)		°⊗_ (≡) [*] (
ENSEMBLE CONFIGURATION	1					P07
Objective field: Diabetes	ABO	Typ	e: icision Forest	Number	of models:	Number of iterations:
Advanced configuratio	n					©
Diabetes diagnosis dataset	t Training (70%)'s e	nsemble			Reset	ਿੰਨੀ Create ensemble
					٩	×
				_		
Name	Type	Count	t Missing	Errors	Histogram	
Name Pregnancies	Type		t Missing 0	Errors 0	Histogram	

Figure 2.17: Number of models

2.4.5 Number of Iterations

This parameter sets the maximum number of iterations to be performed when **Boosted Trees** is selected. For regression ensembles, one boosted tree will be generated for every iteration. For classification ensembles, however, "N" trees will be generated for every iteration where "N" is the number of classes in the objective field.

By default, the number of iterations is 10 and the maximum allowed is 1,000 with a limit of 2,000 maximum single models built. If you set 1,000 iterations using a dataset with more than two classes for the objective field, in the case the ensemble reaches the maximum of 2,000 trees, it will stop.

Note: when one of the early stopping options is enabled (see Subsection 2.4.7.1), the final number of iterations may be lower than the number of iterations configured.

Sources Datasets Su	ipervised 🔻 🛛	NEW Unsupervised ▼	Predictions -	Tasks		WhizzML 🔻
a ₩, <mark>8</mark> ,	• [Diabetes	diagnosis data	aset Training	(70%)	‱_ (≡)5- (i
ENSEMBLE CONFIGURATION						Por
Dbjective field: Diabetes	•	ABC V	Type: Boosted Trees	¥	Number of models:	10 10 10
Advanced configuration	n					$\overline{\mathbf{O}}$
	Training (70%))'s ensemble - b	oost		Reset	ि Create ensemble
Ensemble name: Diabetes diagnosis dataset	Training (70%)'s ensemble - b	oost			िद्धे Create ensemble Q X
Diabetes diagnosis dataset	Training (70%)			lissing Erro	[Q X
Diabetes diagnosis dataset		e 🗘		lissing Erro	[Q

Figure 2.18: Number of iterations

2.4.6 Trees

Since ensembles are composed of several decision trees, some of the parameters that can be configured for BigML models, can also be applied for ensembles like **Missing splits** and the **Node threshold**. You can also use the **Randomize** parameter to select a random subset of your fields at each split and create a Random Decision Forest. You can find a detailed explanation of the three parameters in the subsections below.

2.4.6.1 Pruning

If pruning is enabled, BigML determines whether each tree split increases the confidence (for classification ensembles) or decreases the expected error (for regression ensembles). If it does not, then the split is pruned away. As explained in Subsection 1.2.4, pruning strategies are key to avoid overfitting, a phenomenon that reduces an ensemble's ability to generalize. The statistical pruning is **only available for decision forests** (see Subsection 2.4.3).

In BigML you can choose three different strategies for pruning (Figure 2.19):

- Smart Pruning: considers pruning the nodes with less than 1% of the instances.
- Statistical Pruning: considers every node for pruning.
- No Statistical Pruning: deactivates pruning altogether.

By default, BigML uses Smart Pruning to create your decision forests.

2.4.6.2 Missing Splits

When training an ensemble, BigML may encounter missing values, which can be either considered or ignored for the definition of splitting rules.

To include missing splits in your ensemble, enable the **missing splits** option. (See Figure 2.19.) If missing values are included in your ensemble, you may find rules with predicates of the following kind: field x = "is missing" or field x = "y or is missing".

BigML includes missing values following the MIA approach¹¹ [14].

By default, BigML does not include missing splits.

2.4.6.3 Node Threshold

Set the **node threshold** to set a limit to each single tree growth within the ensemble. (See Figure 2.19.) A lower threashold simplifies the ensemble while helping to avoid overfitting. However, it may also have reduce the ensemble's predictive power compared to deeper ensembles. The ideal number of nodes may depend on the dataset size and the number of features. Larger datasets with many important features may require more complex ensembles. Reducing the number of nodes can also be useful to get an initial unerstanding of the basic data patterns. Then you can start growing the ensemble from there.

By default, BigML sets a **512 node threshold**. Since nodes are computed in batches, on occasion the final number of nodes can be greater than the node threshold. (See Subsection 1.2.2.)

2.4.6.4 Randomize and Random Candidates

The **randomize** option allows another layer of randomization and it works for Decision Forest and for Boosted Trees. If randomize is enabled, a random subset of the input fields will be selected at each tree split. If you selected Decision Forest as the ensemble type (see Subsection 2.4.3), this will create a Random Decision Forest.

When you click the randomize option, the **random candidates** option will be enabled so you can configure the number of input fields in the random subset to be consider at each split. BigML provides three options (Figure 2.19):

¹¹http://oro.open.ac.uk/22531/1/decision_trees.pdf

- **Default**: the number of fields is the square root of the total number of input fields. This is a basic rule which works pretty well in most cases.
- · Number of fields: this sets a fixed number of the fields to be considered at each split.
- Ratio of fields: sets the number of fields to be considered at each split as a percentage of the total number of input fields.

Note: for text fields each term (e.g., each word if space is the chosen separator) counts as an individual field.

,	Supervised - Unsupervised - Predictions - Tasks	Whi	zzML 👻
¥. 8	Absenteeism_at_work [filtered] Training (69%)	ılề- 🗞 (≡)	۶.
SEMBLE CONFIGURATI	ΟΝ		
ective field:	Automatic optimization		•
r pe: Decision Forest	Number of models: Number of iterations: Image: Control of the iteration of the ite		
Advanced configura	ation		۲
Tree:			۲
Statistical pruning		STATISTICAL PRUNING	0
Missing splits:			0
		AUTO	
Missing splits:		AUTO MISSING SPLITS NO NODE THRESHOLD	0
Missing splits:	512 Random candidates:	AUTO MISSING SPLITS NO NODE THRESHOLD 512 RANDOMIZE	0

Figure 2.19: Trees parameters configuration

2.4.7 Boosting Parameters

When you select **Boosted Trees** as the ensemble type (see Subsection 2.4.3), the parameters in the Boosting tab will be enabled: **early stopping** and **learning rate**. See the following subsections for a detailed explanation.

2.4.7.1 Early stopping

The **early stopping** options try to find the optimal number of iterations by testing the single models after every iteration and resulting in an early stop if not significant improvement is made. Consequently, the total iterations for the Boosted Trees, may be lower than the one set in the **Number of iterations** parameter (see Subsection 2.4.5).

You can select one of these three options:

• Early out of bag: this option tries to find out the optimal number of iterations by recursively building single trees and testing the out-of-bag samples after every iteration. This option will use the parameters set in the ensemble sample to build the single trees (see Subsection 2.4.9). If no significant improvement is made, it may result in an early stop. By default this option is enabled. (See Figure 2.20.)

- Early holdout: this option tries to find out the optimal number of iterations by recursively building single trees and holding out a portion of the dataset for testing at the end of every iteration. If no significant improvement is made on the holdout, it may result in an early stop. The percentage of the dataset holdout is set to 30% by default, but you can configure it. By deafult this option is disabled.
- **None**: this option deactivates the early stopping parameter so the total iterations will be the same as the ones specified in the **Number of iterations** parameter (see Subsection 2.4.5). By deafult, this option is disabled.

2.4.7.2 Learning rate

The **learning rate**, also known as the gradient step, controls how aggressively the boosting algorithm fits the data. You can set values greater than 0 and smaller than 1. Larger values will prevent overfitting, but smaller values generally work better (usually 0.1 or lower). By default it is set to 0.1. (See Figure 2.20.)

ources Datasets Supervised -	Unsupervised -	Predictions 🔻	Tasks		v	VhizzML 🔻
₩ ₩ ₩, ₽ ₩, •••••	Diabetes	diagnosis data	aset Trainir	ıg (70%)	🎭 🕉 ((≡)∱ (
ISEMBLE CONFIGURATION						,
ojective field:		Type: NEW		Number of models:	Number of iter	
Diabetes	ABC	Boosted Trees	v	0	10	10
Advanced configuration						۲
Tree:						়
Boosting:						۲
Early stopping:						0
Early out of bag					Early out of bag	
Learning Rate (LR):						0
10%					10%	
Weights:						⊘
Tree sampling:						◙
Dataset sampling:					537 instances	♥
Determined and the second s					D-4	



2.4.8 Weight Field Options

It is not unusual for a dataset to have an unbalanced objective field, where some categories are common and others very rare. For example, in datasets used to predict fraud, usually fraudulent transactions are very scarce compared to regular ones. When this happens, ensembles tend to predict the most frequent values simply because the overall ensemble's performance metric improves with that approach. However, in cases such as fraud prediction, you may be more interested in predicting rare values rather than successfully predicting frequent ones. In that case, you may want to assign more **weight** to the scarce instances so they are equivalent to the abundant ones.

BigML provides three different options to assign specific weight to your instances.

2.4.8.1 Balance Objective

When you set the **balance objective** weight (see Figure 2.21), BigML automatically balances the classes of the objective field by assigning a higher weight to the less frequent classes, with the most frequent class always having a weight of 1. This option is only available for classification ensembles. For example, take the following frequencies for each class:

[False, 2000; True, 50]

By enabling the **Balance objective** option, BigML will automatically apply the following weights:

[False, 1; True, 40]

In this example, the class "True" is getting forty times more weight as it is forty times less frequent than the most abundant class.

2.4.8.2 Objective Weights

The **objective weights** option allows you to manually set a specific weight for each class of the objective field. BigML oversamples your weighted instances replicating them as many times as the weight stablishes. If you do not list a class, it is assumed to have a weight of 1. Weights of 0 are also valid. This option is only available for classification ensembles. (See Figure 2.21.)

This option can be combined with the **Weight field** (see Subsection 2.4.8.3). When combining it with the **Weight field**, both weights are multiplied. For example if you assign a weight of 3 for the "True" class and the weight field assigns a weight of 2 for a given instance labeled as "True", that instance will have a total weight of 6.

2.4.8.3 Weight Field

The **Weight Field** option allows you to assign individual weights to each instance by choosing a special weight field. (See Figure 2.21.) It can be used for both regression and classification ensembles. The selected field must be numeric and it must not contain any missingvalues. The weight field will be excluded from the input fields when building the ensemble. You can select an existing field in your dataset or you may create a new one in order to assign customized weights.

For example, below is a dataset for which we included a field called "Weight" that assign a ten time higher weight to fraudulent transactions in comparison to non-fraudulent ones. BigML provides a powerful tool, the BigML Flatline editor, to add new fields to your dataset, such as a weight field. As an additional example, we could also take into account the transaction "Amount" to calculate the weights, so transactions with higher amounts will have higher weights.

Trans. ID	Products	Online	Amount \$	Fraud	Weight
xxxxxx098	XYZGH	yes	3,218	FALSE	1
xxxxx345	VBHGF	no	1,200	FALSE	1
xxxxxx123	UYFHJ	yes	5,000	FALSE	1
xxxxxx567	HSNKI	no	390	FALSE	1
xxxxx789	SHSYA	yes	500	TRUE	10
xxxxxx093	DFSTU	yes	423	FALSE	1
xxxxxx012	TYISJ	yes	60,000	FALSE	1
xxxxx342	SJSOP	no	789	FALSE	1
xxxxxx908	IOPKJ	no	9,450	FALSE	1
xxxxx334	HIOPN	yes	50,678	TRUE	10

Table 2.1: Weight Field example for transactional dataset

ources Datasets Supervis	ised - Unsupervised -	Predictions -	Tasks			WhizzML
	Diabetes	s diagnosis data	set Trainir	ng (70%)	% _ (5-	(≡) ^{\$} ,
SEMBLE CONFIGURATION						
jective field: iabetes	ABC	Type: MEM Boosted Trees	¥	Number of models:	Number of ite	erations:
Advanced configuration						(
Tree:						$\overline{\mathbf{O}}$
Boosting:						⊘
Weights:						۵
Select a weight option Balance objective Weight field					WEIGHTS No weights	0
Tre Objective weights						$\overline{\mathbf{O}}$
					537 instances	
Dataset sampling:						
Dataset sampling: Dataset advanced sampling:					Default settings	♥

Figure 2.21: Weighting parameters

2.4.9 Trees Sampling

The trees sampling is a different concept than the dataset sampling (see Subsection 2.4.10). The dataset sampling is the one present in other BigML resources and it applies sampling just once to the input dataset before building the resource. In the case of the trees sampling, sampling is applied to the dataset as many times as the number of models that compose the ensemble. This way, separate samplings are created for each tree composing the final ensemble.

The default is to sample with a rate of 100% and with replacement, meaning that the same instance can be selected more than once. This ensures a different sampling for each tree. The out of bag option is not available for trees sampling.

How to Get Repeatable Ensembles

If you create two ensembles using the same dataset and same configuration you will see that the individual trees composing each ensemble may be different. This is due to sampling "randomization" that BigML algorithm uses. Although this does not usually affect the predictive performance of your ensembles, there may be situations in which it is desirable to produce exactly the same ensembles. You can achieve this by setting the **Sampling** option to deterministic (see <u>Subsection 2.4.9.2</u>).

See below an explanation of each sampling option to build the single trees.

2.4.9.1 Rate

The sampling **rate** is the percentage of instances being extracted from the dataset and included in your per-tree sample. A sampling rate of 100% means that all instances are included; a rate of 10% means that only every tenth instance is included in each single model. This option may take any value between 0% and 100%. You can easily configure the **rate** by moving the slider in the **configuration panel for sampling**, or by typing the percentage in the tiny input box, both highlighted in Figure 2.22.

By default, BigML uses a 100% rate combined with replacement (see Subsection 2.4.9.3).

2.4.9.2 Sampling

The **sampling** option represents the type of the sampling process, which can be either random or deterministic. (See Figure 2.22.)

When using deterministic sampling the random-number generator will always use the same seed, producing repeatable results. (See <u>Subsection 2.4.9.</u>)

By default, BigML uses random sampling.

2.4.9.3 Replacement

The **replacement** option controls whether a single instance can be selected multiple times or not. Sampling without replacement ensures that each instance cannot be selected more than once. (See Figure 2.22.)

By default, BigML generates samples with replacement.

Sources Datasets Supervis	sed 👻 Unsupervised 🕶	Predictions Ta	sks		W	hizzML 🔻
· ····	Diabetes	diagnosis datase	Traini	ng (70%)	% _ (≭	≡) * _ (i
SEMBLE CONFIGURATION						PDI
bjective field: Diabetes	ABC v	Type: NEW Boosted Trees	v	Number of models:	10 Number of itera	10
Advanced configuration						۲
Tree:						়
Boosting:						♥
Weights:						⊘
Tree sampling:						۲
Ensemble rate:	100%	ENSEM. RATE		ENSEM. SAMPLING Random	ENSEM. REPLACEMEN	0
.					7 5	
Dataset sampling:					537 instances	$\overline{\mathbf{O}}$
Dataset advanced sampling:					Default settings	$\overline{\mathbf{r}}$

Figure 2.22: Trees sampling for ensembles

2.4.10 Dataset Sampling

Sometimes you do not need all the instances contained in your testing dataset to build your ensemble. If you have a very large dataset, sampling may be a good way of getting faster results. (See Figure 2.23.)

The same sampling options described in the **Datasets with the BigML Dashboard document** [9] to sample datasets, are also available when building BigML ensembles. They are divided in two groups: sampling and advanced sampling options. (See Figure 2.23.)

2.4.10.1 Rate

The sampling **rate** is the frequency of instances being extracted from the dataset and included in your sample. A sampling rate of 100% means that all instances are included; a rate of 10% means 10% of the instances are included. This option may take any value between 0% and 100%. You can easily configure the **rate** by moving the slider in the **configuration panel for sampling**, or by typing the percentage in the tiny input box, both highlighted in Figure 2.23.

By default, BigML uses a 100% rate.

2.4.10.2 Range

The sampling **range** is the subset of the dataset instances from which to sample, e.g., from instance 5 to instance 1,000. The **rate** will be applied over the range configured.

By default, all instances are included, i.e., the range is (1, num. rows in dataset).

2.4.10.3 Sampling

The **sampling** option represents the type of the sampling process, which can be either random or deterministic.

When using deterministic sampling the random-number generator will always use the same seed, producing repeatable results.

By default, BigML uses random sampling.

2.4.10.4 Replacement

The **replacement** option controls whether a single instance can be selected multiple times or not. Sampling without replacement ensures that each instance cannot be selected more than once.

By default, BigML generates samples without replacement.

2.4.10.5 Out of Bag

The **out of bag** option allows you to include in your sample only those instances that were not selected in the first place, thus effectively inverting the sampling outcome. It is only selectable when a sample is deterministic and the sample rate is less than 100%. The total percentage of instances included in your sample will be one minus the **rate** (when replacement is not allowed). This can be useful for splitting a dataset into training and testing subsets.

By default, BigML will not use out of bag instances.

Sources Datasets Sup	pervised - Unsupervise	d 👻 Predictions 🕶	Tasks		W	nizzML 🔻
a 17. 19	Diabe	etes diagnosis da	taset Training (70	%)	🇞 - (≡	;) % _ (i
NSEMBLE CONFIGURATION						20
Dijective field: Diabetes	ABC	Type: NEW Boosted Trees	Num	ber of models:	Number of itera	itions:
Advanced configuration						۲
Tree:						$\overline{\mathbf{O}}$
Boosting:						$\overline{\mathbf{O}}$
Weights:						$\overline{\mathbf{O}}$
Tree sampling:						$\overline{\mathbf{O}}$
Dataset sampling:				53	37 instances	۲
Rate:	100%				SAMPLING RATE	0
Dataset advanced sampli	ing:			Cu	stom settings	۲
Range: 537 instances	537	RANGE 1 - 537	SAMPLING Random	REPLACEMENT NO	OUT OF BAG	0

Figure 2.23: Dataset sampling arguments for ensembles

2.4.11 Advanced Ordering

Ordering options are relevant to ensure that BigML can correctly determine whether it can take an early split of your dataset to accelerate the training process. In particular, early splitting can only be safely used if the training instances have been previosuly shuffled. (See Subsection 1.2.3.1.)

If your instances are already shuffled, BigML allows you to choose the **linear** option. This will make the process of building the ensemble much faster, since it will not required to reshuffle the dataset. If you need to shuffle your instances, BigML provides two options to that aim, **deterministic shuffling** and **random shuffling**, which are described below.

Ordering options have no influence on datasets of less than 34GB, since the whole dataset is used to build the ensemble.

By default, BigML uses **deterministic shuffling** to ensure the same (deterministic) sample of the instances is used and the built ensemble is thus repeatable.

2.4.11.1 Deterministic Shuffling

The **deterministic shuffling** option ensures that the row shuffling of a dataset is always the same, so that retraining a BigML ensemble from the same dataset yields the same results. (See Figure 2.23.)

By default, this option is true.

2.4.11.2 Linear Shuffling

The **linear shuffling** option is useful when you know that your instances are already in random order. Using linear shuffling, the BigML ensemble will be constructed faster. (See Figure 2.23.)

By default, this option is false.

2.4.11.3 Random Shuffling

The **random shuffling** option will ensure that a different shuffling will be tried each time you train your ensemble. (See Figure 2.23.)

By default, this option is false.

Sources Datasets Supervised -	Unsupervised -	Predictions -	Tasks		I.	WhizzML 🔻
	Diabetes	diagnosis data	aset Trainir	ng (70%)	🍖 👶	(≡)⁵, 🕕
ENSEMBLE CONFIGURATION						104
Objective field: Diabetes	ABC V	Type: NEW Boosted Trees	¥	Number of models:	Number of ite	erations: 10
advanced configuration						۲
Tree:						$\overline{\mathbf{O}}$
Boosting:						♥
Weights:						$\overline{\mathbf{O}}$
Tree sampling:						$\overline{\mathbf{O}}$
Dataset sampling:					537 instances	$\overline{\mathbf{O}}$
Dataset advanced sampling:				(Custom settings	◙
Advanced ordering:				(Custom settings	•
Ordering:					ORDERING Deterministic	0

Figure 2.24: Ordering options for ensembles

2.4.12 Creating Ensembles with Configured Options

After finishing the configuration of your options, you can change the default ensemble name in the editable text box. Then you can click on the Create ensemble button to create the new ensemble, or reset the configuration by clicking on the Reset button.

ctive field:	BC v	tomatic optimization			
ecision Forest	Number of models:	Number of iteration	64		
Advanced configuration					(
Tree:					♥
Boosting:					♥
Weights:					$\overline{\mathbf{O}}$
First split:					ਂ
Ensemble sampling:					َ⊘
Dataset sampling:				7,388 instances	ਂ
Dataset advanced sampling	r.			Default settings	ਂ
Advanced ordering:				Default settings	◙
emble name: quality			Re	set Fra Create er	semble

Figure 2.25: Create ensemble after configuration

2.4.13 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to the Reset button (See (Figure 2.25)). This is to show how to create the ensemble programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 2.26) below:

ojective field:		Automatic optimization	
Гуре:		Number of models: Number of iterations: 🕢	
Decision Forest		×	
		est preview	
🏷 Advanced o	API endpoin 1 ₹ { 2	"objective_field": "00001a",	
Tree:	3 4 5	"number_of_models": 10, "node_threshold": 512, "name": "lc_quality",	
Boosting:	6 7 +	"optimize": false, "ensemble_sample": {	
Weights:	8 9 10	"rate": 1, "replacement": true, "seed": "c7l8l4f0fb38391a53976be721e8c5e2"	
	11	},	
First split:		Check the full list of arguments in the API documentation	

Figure 2.26: Ensemble API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for ensembles, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the chosen configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

2.5 Visualizing Ensembles

Being able to effectively visualize an ensemble is paramount to exploring it, interpreting it, and explaining why it produces certain outcomes. BigML provides two different visualizations, a Partial Dependence Plot (PDP) and a list of the single models:

- **Partial Dependence Plot**: a graphic representation of the marginal effect that the combination of two fields (predictors) have on the objective field (ensemble predictions) keeping the rest of the field values constant.
- **Model list**: provides a list of the single models that form the ensemble. This visualization is only available for **Decision Forest** ensembles (see Subsection 2.4.3).

Note: BigML does not provide the model list for Boosted Trees because they cannot be interpreted the same way as the models for other ensemble types (see Subsection 2.5.2).

In the top menu you will find a summary of the ensemble results: the number of models in the ensemble, the sample rate, the objective field used, if the ensemble has been randomized, the type of ensemble used (Decision Forest or Boosted Trees), and the number of instances in the dataset.



Figure 2.27: Ensemble top menu

ources Datasets	Supervised -	Insupervised Pre	dictions - Tasks			WhizzML
	••••	Arrythmia d	iagnosis dataset'	s ensemble	â 🗘	
MODELS 432	RATE 100%	REPLACEMENT	OBJECTIVE FIELD		GING	INSTANCES 452
				NUME	ER OF MODELS	
Y V6: Amplitude T wav	e v	1.1250	£}	0	PREDICTION	esc Release
6.0				Normal		Vote share
5.0				41%		100%
3.0				Normal	_	100%
2.0						
1.0				All classes		٣
0				0	INPUT FIELDS	All fields: 🔲
-1 -				Age		0.33%
-2 -				0		103 46
-3						
-4 -				Height 0		948
-5				-0-		166
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	1 ¹⁰ 12 13 14	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Weight		0.22%
			.1880 🏿	0		218

For Decision Forests, below the top menu, you will find the icons corresponding to each one of the views (the PDP and the model list) to switch from one view to another. (See Figure 2.28).

Figure 2.28: Switch from the chart view to the model list view

You can find a detailed explanation of each view in the following subsections.

#### 2.5.1 Partial Dependence Plot (PDP)

The PDP is the main default view you will find when creating an ensemble. The main goal is to represent the marginal effect of a set of variables (input fields) on the ensemble predictions disregarding the rest of the variables. It is a common method for visualizing and interpreting the impact of the variables on ensemble predictions, and it can be used for classification and regression ensembles.

# Note: the ensemble PDP is not a representation of the dataset values; it is a representation of the ensemble results and their dependence from a set of variables used as inputs.

In order to ensure responsiveness, the PDP is built using 10 models by default. For ensembles with a higher number of models, a **random sample of 10 models** will be selected to calculate the predictions. A warning message will appear at the top of the ensemble view to indicate that the chart has been built with a lower number of models because this may cause slight differences between the chart predictions and the ensemble actual predictions. Although in most cases these differences should be imperceptible, you can use the slider to increase the number of trees (up to 100 trees) and the re-sampling option to take another random sample of trees. Click on the corresponding options as shown in Figure 2.29.

ources Datasets Supervised		ictions - Tasks		WhizzML
₽, <mark>0</mark> ,	Arrythmia diag	nosis dataset's ensemble	v1 📋 🗘	<b>⊘</b> _~ (≡) [*] _~
250 100%	ARRHYTHMIA	>\$ NO	DECISION FOREST	452
		NUMBER OF MO	DDELS	
		-0-	- 47	
-		X >		esc Releases
Y V1: Amplitude R' w 123	• 10.8000	£3	PREDICTION	
18		Ischemic cha	nges (Coronary Artery Disea	se) Vote share
10		0%		100%
16				
14 -		Atrial Fibrillat		0.00%
12			anges (Coronary	70.21%
		Left bundle k	pranch block	0.00%
10 -		All classes		Y
8.0 -		17	INPUT FIELDS	All fields: 🖃
6.0		Heart rate		) 19.05% 🗹
4.0 -		14		192
			0	- 74
2.0		AVR: Ampli	tude T wave	4.52%
<u>с</u> с « « « «	v o vo to to vo	-6.47 -6.47		8.98
, או כן א, כן ע	∧ 0 √0 ₽0 № .	67 67	-0	-1.14
(	-2.2	2500 Ø V3: Q wave		3.43%

Figure 2.29: Trees slider and resampling option for ensembles PDP

You can visualize classification and regression ensembles in the heatmap chart. In the case of **classification** ensembles, the different classes of the objective field are represented by **different colors**. The different color shadings for each class represent the different **votes** in the case of **Decision Forests**, i.e., the percentage of trees voting for a given class in the ensemble (see Subsection 2.6.3.2) and the class **probabilities** in the case of **Boosted Trees** (see Subsection 2.2.1). For **regression** ensembles, the different prediction values are represented by differences in the **color scale**.



Figure 2.30: Classification and regression ensembles

The chart view is always composed of three main parts: the CHART itself, the PREDICTION legend and



the INPUT FIELDS form. (See Figure 2.31.)

Figure 2.31: Ensemble chart

• The CHART allows you to view the impact of the two input fields on the objective classes predictions. You can select any categorical or numeric field for each axis. You can also switch the axis by clicking on the option on top of the chart area. (See Figure 2.32.) In the grey area next to the axis selectors you can see the axis values. You can freeze the view by pressing Shift and release it again by pressing Escape from your keyboard. When the view is frozen, an edition icon will appear and you can edit the axis values to obtain a prediction for that value. (See Figure 2.32.)



Figure 2.32: Ensemble CHART options

• The PREDICTION legend allows you to visualize the objective field classes (classification ensembles) or the predicted value (regression ensembles). In the case of classification ensembles you will also obtain the votes, i.e., the percentage of trees voting for a given class int he ensemble (see Subsection 2.6.3.2), for Decision Forests or the class probabilities for Boosted Trees (see Subsection 2.2.1). By default, color tones and shadings are set according to the range of values shown in the chart area. This is the default because for some configurations of the chart the predictions may vary a small amount relative to the global range. For example, imagine the chart is showing temperature predictions based on location, time-of-year, and time-of-day. San Diego's daily range (13° C to 18° C) could be tiny compared to the Earth's global range (-62° C to 48° C). You can change this behavior and see the color scales and shading according to the total range of possible predicted values by clicking on the icon Total. (See Figure 2.33.) For classification ensembles, this option allows you to see the color shading for the total range of potential values (from 0% to 100%). For regression ensembles, the Total colors option allows you to see the color scale for the total range of predictions. For classification ensembles you can also select to see only one of the classes using the class selector at the bottom of the legend. (See Figure 2.33.)



Figure 2.33: PREDICTION legend options

• Below the chart legend, you can find the INPUT FIELDS form. (See Figure 2.34.) You can configure the values for any **numeric** or **categorical** field. Text and items fields are not yet supported. By changing their values, you can see the predictions changing in real-time. You can sort the fields by their importance, select or disable them. If you disable an input field, it will be ignored to calculate the final prediction. The strategy used to calculate predictions when some fields are disabled is the **proportional missing strategy** (see Subsection 2.6.3.1).

Note: it is important to notice that disabled fields will be ignored when calculating the chart predictions. This is because the original intent of the PDP is to understand the impact of the axis fields by ignoring the influence of all the other fields. So if you trained the ensemble with missing values (see Subsection 2.4.6.2) and they have some impact on predictions, you will not see it in the chart predictions. In this case it will be a mismatch between the chart predictions and your final predictions.



Figure 2.34: INPUT FIELDS form in ensemble chart

#### 2.5.1.1 Export chart as an image

Download the ensemble chart as an image in PNG format with or without legends. To download it with legends, press Shift from your keyboard to freeze the chart view. (See Figure 2.35.)



Figure 2.35: Download ensemble chart in PNG format

#### 2.5.1.2 Interpreting Partial Dependence Plots

You can easily see field impact on predictions using the ensembles chart. See below three different situations using an ensemble which aims to predict if a person has diabetes based on several input fields:

• Both fields impact predictions: in the image below, the combination of the selected fields, "BMI" (Body Mass Index) and "Glucose", have a high impact on predicting diabetes since variations in both fields cause variations in predictions.



Figure 2.36: Both selected fields impact predictions

• **Only one of the fields impacts predictions**: looking at the image below we can conclude that "Skinfold" is not a good predictor for diabetes since variations in this field don't affect predictions. However, the level of "Glucose" has great impact on predictions. (See Figure 2.37.)

	••• ∲	Diabetes diag	gnosis dataset el'	's ensemble 📑 🗘	<b>(</b> ≡) ⁵ -
MODELS 10	RATE 100%	REPLACEMENT	OBJECTIVE FIELD DIABETES	ENSEMBLE BAGGING	INSTANCES 768
				NUMBER OF MODELS	
Y Skinfold	•	61.8750	, fř	PREDICTION	esc Releases
				True	Vote share
90 -				60%	100%
- 08				True	90%
70 -					
60 -				All classes	Y
50				INPUT FIELDS	All fields: 🔲
40 -				Pregnancies	3.93%
30 -				0	21 3
20				Blood pressure	6.98%
10				0	152 <b>69</b>
0 .					03
0 20 10	6, 8, 6	29 <i>149 16</i> 9	18 ⁰	Insulin	6.15%
Glucos	Se .	* 174	.1250	0	1057

Figure 2.37: One of the selected fields impact predictions

• Both fields have low or no impact on predictions: if you select variables with little or no influence on predictions, you can see that variations in the selected fields don't lead to differences in predictions. In this case, any combination of "Blood pressure" and "Insulin" always returns the same value for diabetes, "False".

ources Datasets	Supervised -	Unsupervised   Pre	dictions Tasks	_	WhizzML •
¥- 0,	••••	Diabetes dia	gnosis dataset el's	ensemble 📋 🗘	(≡) ⁵ , (
MODELS 10	RATE 100%	REPLACEMENT	OBJECTIVE FIELD DIABETES	ENSEMBLE BAGGING	INSTANCES 768
				NUMBER OF MODELS	
Blood pressure		· 68.6250 d	۲. Ex	PREDICTION	esc Releases v
120 -				alse	Vote share
110 -			10	00%	100%
90			F	alse	100%
80 .			A	ll classes	v
60 -				INPUT FIELDS	All fields:
50				Pregnancies	All fields.
40 .				0	21
30 -					3
20				Glucose	32.80%
10 -				0	248
°, °	\$° \$°	20 60 60 10	e %e		
~ ~~	T G	r & & Y	- <del>0</del> 5	Skinfold I	4.80%
X Insu	in	• 687	7.3750 🖋	-0-	20

Figure 2.38: None of the selected fields impact predictions

#### 2.5.2 Model List

The model list is only available for **Decision Forests**. The model list is not provided for **Boosted Trees** because they cannot be analyzed the same way as the models for other ensemble types. Instead of predicting the objective field, each boosted tree tries to fit a gradient to correct the mistakes made by the previous single tree. Therefore the single models are hardly interpretable individually.

The same BigML interactive visualization for models is used to show each single tree composing the ensembles. The ensemble model list provides a general overview on the ensemble and a means to get down to the single model level. The list of models comprise the following information for each of them:

- Model preview: this is a snapshot of the tree representation for each model.
- Model link: you can access each underlying tree model by clicking on this link.
- Data distribution histogram: the distribution of the target field values in the training set that was used to build the model.
- **Predicted distribution histogram**: predictions distribution for the model. The predicted distribution should roughly match the data distribution.



Figure 2.39: Ensemble model list view

If you click on the model preview or link, you will be taken to that model view (see Figure 2.40), where you can use all of the model visualization features described in Section 1.5, such as BigML proprietary tree and sunburst dynamic visualization.



Figure 2.40: Tree visualization

# 2.6 BigML Ensemble Predictions

## 2.6.1 Introduction

The ultimate goal in building a BigML ensemble is being able to make predictions for previously unseen instances with an unknown label. In BigML, you can make predictions for **single instances** or for **many instances in a batch**. Each **prediction** comes with a measure indicating its **reliability**, expressed either as a **probability** (for classification ensembles), as a **confidence** or **votes** (only for classification Decision Forests), or as an **expected error** (only for regression Decision Forests).

The predictions tab in the main menu of your BigML Dashboard is where all your saved predictions are listed. (See Figure 2.41.)

Sources	Datasets	Supervised -	Unsupervised - Pre	dictions	<ul> <li>Tasks</li> </ul>			WhizzM	L ▼
<b></b>	Ē		2	Pred	lictions			Ô	Q
ជា	Name			Obje	ctive	\$ Prediction	٥	Ē	
				No predi	ctions				
Show 10	predictions			No predictio	ins found		1	< < >	

Figure 2.41: Predictions list empty view

Ensemble predictions are saved under the CLASSIFICATION & REGRESSION option in the menu. (See Figure 2.42.)

Sources	Datasets	Supervised -	Unsupervised -	Predictions -	Tasks		Whizz№	ИL <del>▼</del>
=	Ē		CLASSIFICATIO	DN & REGRESSION 🗸	ions		Ô	Q,
ŝ	Name		ANOMALY DET	ECTION		\$ Prediction	\$	
			ASSOCIATION TOPIC MODEL		S			
Show 10	predictions			No predictions	found		K < 2	$\geq$

Figure 2.42: Menu options of the predictions list view

From this view you can select to view the list for your **single instances predictions** or your **batch predictions** by clicking in the corresponding icons. (See Figure 2.43 and Figure 2.44.)



Figure 2.43: Single predictions icon



Figure 2.44: Batch predictions icon

In the **predictions list view**, you can see, for each prediction, the **Model, Ensemble or Logistic Regression** icon used for the prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created). (See Figure 2.45.)

Figure 2.45: Predictions list view

You can also search your predictions by name by clicking the search button on the top right menu.

#### 2.6.2 Creating Ensemble Predictions

As shown in Figure 2.46, BigML provides three options to make predictions from your ensembles:

- 1. PREDICT: to predict a single instance using the prediction form.
- 2. BATCH PREDICTION: to predict multiple instances simultaneously.

Sourc	ces Datasets <b>Supervised -</b> Unsuperv		edictions - Tasks				WhizzN	
			Ensembles	Search by	name		<b></b>	⊂ × €
ıll	Name	≎ Type ≎	Objective $\diamond$	÷	<b>B</b> \$	\$	<b>(</b>	
ılî	US car accidents dataset   Training (80%) 512-node, 10-model bootstrap decision forest	8 ol	Injury Severity	1w3d	592.9 KB	15	7	6
ılî	US car acc S12-node,	*	Crash Date.day-of-we	1 w 3d	592.9 KB			
ılî	Iris dataset 512-node, 1		Species	1w3d	3.6 KB	1	5	8
ılî	car_crash	.0	Injury Severity	2w 1d	10.9 MB	12	2	
ılî	US car acc	, o <b>o</b>	Injury Severity	3w	592.5 KB	21		
ılî	Fictional Wine sales v1 512-node, 31-model bootstrap decision forest	*	Total Sales	3w 1d	50.4 KB	5		
ılî	Fictional Wine Sales	木	Total Sales	3w 1d	50.4 KB	4	10	11
ılî	Fictional Wine Sales 512-node, 10-model bootstrap decision forest	٨	Total Sales	3w 1d	50.4 KB			
ılî	Fictional Wine Sales	٨	Total Sales	3w 1d	50.4 KB	1		
ılî	Diabetes diagnosis dataset   Training (70%) v4	00	Diabetes	3w 1d	18.9 KB	19	14	32
show [	10 🗘 ensembles	1 to 10	) of 379 ensembles		K	< 1 2	3 4 5	> >

Figure 2.46: Menu options to create predictions

#### 2.6.2.1 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the ensemble, so you can easily set the input data and get an immediate response. This option is only available from the BigML Dashboard for ensembles with less than 100 fields. If you want to perform single instance predictions for ensembles with a higher number of fields, you can use the BigML API¹².

Follow the steps detailed below to create a single prediction:

¹²https://bigml.com/api/predictions

1. Choose the PREDICT option under the ensemble 1-click menu. (See Figure 2.47.)



Figure 2.47: Predict option from ensemble 1-click menu

Alternatively, you can choose the PREDICT option in the **pop up menu** in the list view as shown in Figure 2.48.

			66	Ensembles						Q,
II	Name	٥	Type 🗘	Objective	¢ 🛱	\$	ê \$	\$	\$	
1 <mark>î</mark>	Diabetes diagnosis dataset - sample (20.00	8	00	Diabetes	3	3d 19h	54.1 KB	0	Û	2
ıî ^c	Arrythmi.		00	Arrhythmia	1	8d 19h	344.0 MB			
lî ^c	Fictional		木	Total Sales	Ę	id 23h	1.8 MB			
ıî.			入	Total Sales	Ę	id 23h	4.0 MB			1
lî ^c	Fictional		入	Total Sales	Ę	id 23h	402.6 KB			
ıî ^c	Diabetes tragnosis trataset - sample (ou.ou	2	o0	Diabetes	Ę	id 23h	215.7 KB			
ıî.	Diabetes diagnosis dataset - sample (80.00		o0	Diabetes	Ę	id 23h	215.7 KB			1
ıî.	Diabetes diagnosis dataset - sample (80.00		00	Diabetes	ŧ	id 23h	215.7 KB			1
ĥ	Diabetes diagnosis dataset's ensemble		00	Diabetes	Ę	id 23h	269.8 KB			

Figure 2.48: Predict option from ensemble pop-up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the ensemble as predictors ordered by field importance. The importance percentage is found next to the field name as shown in Figure 2.49. You may not find all the fields from your original dataset because the ensemble may find them irrelevant or redundant in terms of their predictive impact.



Figure 2.49: Single predictions form

3. Select the fields you want to be taken into account for your prediction as shown in Figure 2.50. Non-selected fields will be considered as missing values during the prediction. If your ensemble was trained with Missing splits (see Subsection 2.4.6.2), then missing values are considered by the ensemble as any other valid value. If your ensemble was built without missing values then any of the Missing strategies may apply during your prediction (see Subsection 2.6.3.1.)



Figure 2.50: Select fields in the prediction form

- 4. Set input values for your selected fields. Depending on the field type, you will need to input the values differently:
  - Numeric fields: move the slider or input a specific value in the text box.
  - Categorical fields: select one class from the selector.
  - Text fields: write one or several terms in the free text box.
  - Date-time fields: select the appropriate values from the selector.
  - Items fields: when you write the first three characters of an item name, several items matching those characters will appear, so you can select the right one. You can input more than one item for a field.
- 5. **Get the prediction** on the top of the form. For classification ensembles you will get all classes distribution and for regression ensembles you will get the predicted value for the objective field. Both types of ensembles will also show a certainty measure along with the prediction:
  - For classification Decision Forests, you can get the probability, the confidence or the votes depending on the option you choose. For regression Decision Forests you get the expected error.
  - For classification Boosted Trees, you get the probability. For regression Boosted Trees the expected error cannot be calculated. Read more about ensemble predictions in Subsection 2.2.1.2.

Missing strategy: 💦 🔏	Diabetes: True	etes diagnosis dataset   Trai 54.26% 54.26% 54.26% Sability threshold:	50%
Glucose	45.74% ç ³⁶⁹	54.25%	50%
Glucose	5 ³⁰⁰	-t ^{ub}	50%
Glucose	5 ³⁰⁰	-t ^{ub}	
Glucose			
Glucose			
Glucose	Select a positive class	pability threshold:	
Glucose	Select a positive class    Prob	pability threshold:	
0	31.55%	ВМІ	
	248 <b>124</b>	0.0	83.88
Age	■ 12.00%	Diabetes pedigree	<ul> <li>10.17%</li> </ul>
8	84 <b>46</b>	0.0	3.01
Blood pressure		Skinfold	• 06.91% <b>V</b>
	142 <b>71</b>	0	78 39
		0	
Pregnancies	e	Insulin 0	930
	10		465
New prediction name Diabetes diagnosis dataset   Training	(70%) \/3		Save

Figure 2.51: Single prediction view

BigML predictions are synchronous, i.e., when you send the input data you get an immediate response. Read more about local predictions in Subsection 2.6.2.1.1.

6. Optionally Save the prediction so you can access them afterwards from the ensemble predictions list view.

Sources	Datasets Supervised  Unsupervised	Predictions  Tasks		WhizzML 🔻
*	Ē	Predictions		â <
ជា	Name	Objective	Prediction	> 🛗 0
£Å	Prediction for Total Sales	Total Sales	69.603493	1h 57min
£ 8	Prediction for Total Sales	Total Sales	71.95465	2h 3min
é 😚	Prediction for great_success	great_success	FALSE	17h 29min
f 😚	Prediction for Diabetes	Diabetes	True	19h 48min
fa	Prediction for Diabetes	Diabetes	False	19h 49min
få	Prediction for great_success	great_success	FALSE	19h 51min
få	Prediction for great_success	great_success	FALSE	19h 53min
få	Prediction for Total Sales	Total Sales	71.95465	19h 53min
få	Prediction for Diabetes	Diabetes	True	22h 29min
£Å	Prediction for Total Sales	Total Sales	61.093	5d 11h
Show 10	predictions	1 to 10 of 277 predictions	K	

Figure 2.52: Single predictions list view

#### 2.6.2.1.1 Local Predictions

BigML provides local predictions from the BigML Dashboard for single instance predictions. Local predictions allow you to get a real-time prediction without consuming any credits or requiring an internet connection. This is possible because your ensemble is **saved in the browser's memory** so when the input values change, BigML immediately evaluates all models, obtaining their predictions and then combining them in a matter of microseconds.

Local predictions are only available for ensembles built with **15 models** or less for Decision Forests and **30 models** or less in the case of Bosted trees. For ensembles with higher number of models, you can still perform remote single predictions.

#### 2.6.2.1.2 Predictions with Images

BigML ensembles can be trained from images using extracted image features (Subsection 2.2.3). Because image features are automatically generated numeric fields, creating ensemble predictions with images is the same as creating other ensembles. The only thing different is input fields of images.

Note: When the input fields contain images, in order to create the single prediction, BigML will extract image features automatically to match what were used in the dataset to train the ensemble.

Sources	Datasets Supervised - Unsupervised - Tasks	WhizzML 👻
<b>T</b> .	Predict using grape-strawberry texture	<b>%</b> - ()
	label: ?	۲
Missing st	rategy: 💦 🔏 🔞 🔞 Select a positive class 🔻 Probability threshold:	All input fields:
image_id	<b>•••••••••••••••••••••••••••••••••••••</b>	
	Select image	
New pree	1659.jpg closed, image, 2 fields (1 path, 1 image)	
grape-s	696969.jpeg           open, image, 2 fields (1 path, 1 image)	Predict
	predict-images.zip closed, image, 6 sources, 236 fields (234	

Figure 2.53: Select a single image source in the image input field

The ensemble in Figure 2.53, "grape-strawberry texture", was created from a dataset containing image features *Wavelet subbands*. Creating a prediction using the ensemble will be directed to the **prediction** form which presents all input fields used by the ensemble. One of them is the image field. Because this is a single prediction, an image is input by using a single image source. Clicking on the input field box, single image sources available will be in the dropdown list. There is also a search box which can be used to locate specific ones.

Sources	Datasets Supervised - Unsupervised - Predictions - Tasks	WhizzML 🗸
¥.	Predict using grape-strawberry texture	نى ئ
	label: ?	۲
Missing st	trategy: 🔥 🔏 😰 👔 Select a positive class 🔻 Probability threshold:	50% All input fields: 🗹
image_id	100.00%	
	Select image	
	696969.jpeg open, image, 2 fields (1 path, 1 image)	
New pree grape-s	predict-images.zip	Predict
	grape-strawberry.zip closed, image, 70 sources, 237 fields (1 c	

Figure 2.54: List the components of a composite source

Oftentimes single image sources were used for creating a composite source, they become component sources of the composite source. Or an image was uploaded as a part of an archive file (zip/tar) which created a composite source. In those cases, the composite source will be shown in the dropdown list, along with an icon "List components". In the example in Figure 2.54, predict-images.zip is a composite source, click on the icon to show its component sources.

Sources	Datasets	Supervised <del>-</del>	Unsupervised -	Predictions -	Tasks		WhizzML 🔫
¥.			Predict u	ising grape-strav	vberry texture		<b>%</b> - (i)
			li	abel: ?	$\sim$	?	۲
Missing s	trategy:	8	5 <b>(?</b> d	Select a positive clas	Probability th	ireshold:	All input fields:
image_ic	i		10	0.00%			
	Select image			* <b>D</b>			
	🔶 🔛 pre	edict-images.zip					
New pro	1		(	2			
New pre-	15601		h, 1 image)			e.	Predict
	1659.j	<b>pg</b> mage, 2 fields (1 pati	h, 1 image)				
	32832 open, i	0.jpg mage, 2 fields (1 pati	h, 1 image)				

Figure 2.55: Select a component of a composite source

After the component sources of the composite are listed, scroll the dropdown list to find the desired one, then click to select it, as shown in Figure 2.55. There is also a search box to locate specific component sources.

<b>7</b> -		Predict using bl	uebird-gps-tag	gged		G- (		
		label: ?		?			۲	
lissing strategy: 💦 💦	{%	Select a pa	ositive class 🔻	Probability threshold:		All input fields:	50%	
-158.32		45.18 -97.10325	image_id Selec	t image	•		2	
gpslatitude 23.57	C	58.08 37.57352						
New prediction name								

Figure 2.56: Ensemble prediction form, image field and more

In addition to images, ensembles may use other fields, which will be in the prediction form too. As shown in Figure 2.56, all the fields can be selected, and their input values be set by dragging the knobs on the sliders or by entering precise values in their input boxes.

Once all fields are selected, click on the green button Predict to create a prediction.

Sources	Datasets Supervised <del>•</del> U	supervised - Predictions - T	asks	WhizzML 🔻
A 1	T.	grape-strawberry tex	tture	(≡) ⁵ , (Ξ)
Inputs				$\odot$
		label : grape	79.94%	
{% ?	000			0 %
Ensemble in 5.71	SUPPORT % - 55.72 %	Filter predictions for: label	× 88.85 %	ROBABILITY 99.08 %
1	model/61ae6b0d128d1c677300001c	grape		99.07%
2	model/61ae6b0d128d1c677300001e	grape		98.21%
3	model/61ae6b0e128d1c6773000020	grape		97.64%
4	model/61ae6b0e128d1c6773000022	grape		98.80%
5	model/61ae6b0e128d1c6773000024	grape		97.83%
6	model/61ae6b0e128d1c6773000026	strawberry		97.14%
7	model/61ae6b0f128d1c6773000028	strawberry		88.86%
8	model/61ae6b0f128d1c677300002a	grape		98.14%
9	model/61ae6b0f128d1c677300002c	grape		97.99%
10	model/61ae6b0f128d1c677300002e	grape		97.68%
Show 10	predictions	1 to 10 of 10 predictions		

Figure 2.57: Ensemble single image prediction

After a new prediction is created, as shown in Figure 2.57, the predicted class is at the top of the form along with its probability. The prediction interface is the same as ones created by non-image ensembles. Everything described earlier in this section (Subsection 2.6.2.1) applies.

#### 2.6.2.2 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the ensemble you want to use to make predictions and a dataset containing the instances which you want to use as prediction inputs. BigML will create a prediction for each instance in the dataset. Follow the steps detailed below to create a batch prediction:

1. Select the BATCH PREDICTION option under the ensemble 1-click menu (see Figure 2.58) or the CREATE BATCH PREDICTION option in the pop up menu of the list view (see Figure 2.59.)

a s ^e ir	tin - ••••	Diabetes	diagnosis d	ataset	t - san	npl	Ŷ	 (=) ^{/~}
					st P	REDICT	•	
	SAMPLE RATE	REPLACEMENT		ENS	с 📮 в	ATCH PREDICTION		
10	100%	YES		BA	ee E	VALUATE		
			0057000574		P.	ARTIAL DEPENDENCE	PLOT	
	<u> </u>	odel/57287042b95b39	32857000571		ŕ. <b>⊼</b> ∘	ELETE ENSEMBLE		
	D	ata distribution:			2 M	IOVE TO		
- Ali	· 1.0		_	_	-			

Figure 2.58: Batch predictions option from ensemble 1-click menu

Sources	Datasets Ensembles  Clusters	And	omalies	Associations	Predictions	s Tasks	;		Scrip	ots 🔻
			66	Ensembles						0
III	Name	\$	Туре 🗘	Objective	\$	\$	<b>ê</b> \$	\$	\$	08 80
ıl <mark>î</mark>	Diabetes diagnosis dataset - sample (20.00	8	00	Diabetes		3d 21h	54.1 KB	2	0	2
ılî	Arrythmi Streen		.0	Arrhythmia		3d 21h	344.0 MB			
ılî	Fictional		入	Total Sales		6d 1h	1.8 MB			
ıß	Fictional		٨	Total Sales		6d 1h	4.0 MB			1
ılî	Fictional		٨	Total Sales		6d 1h	402.6 KB	1		1
ıß	MOVE TO Diabetes tragmosis trataset - sample (ou.uu	)		Diabetes		6d 1h	215.7 KB	4		
ılî	Diabetes diagnosis dataset - sample (80.00		o0	Diabetes		6d 1h	215.7 KB			1
ıß	Diabetes diagnosis dataset - sample (80.00		o0	Diabetes		6d 1h	215.7 KB			1
ıl	Diabetes diagnosis dataset's ensemble		00	Diabetes		6d 1h	269.8 KB			
Show 10	♀ ensembles		1 to	9 of 9 ensembles					K < 1	> >

Figure 2.59: Batch predictions option from ensemble pop up menu

- Select the dataset containing all the instances you want to create a prediction for. The instances should contain the input values for the fields used by the ensemble as predictors. You can also select a subset of the ensemble fields to be taken into account by configuring your prediction (see Subsection 2.6.3.5.) BigML batch predictions can handle missing data in your prediction dataset (see Subsection 2.6.3.1.)
- 3. **Optionally, select the ensemble** you want to use for the prediction. BigML pre-selects the ensemble you created the batch prediction from at step 1, but you can change it at any time in the batch prediction view by selecting another ensemble from the ensemble selector displayed in the right pane. You can even switch to a model or logistic regression by selecting the corresponding icon in the top left menu.
| Sources       | Datasets      | Ensembles 🔻       | Clusters         | Anomalies A          | Associations     | Predictions Tasks             | (NEW)<br>Scripts ▼ |
|---------------|---------------|-------------------|------------------|----------------------|------------------|-------------------------------|--------------------|
| <b>A I</b>    | Si l          |                   |                  | New                  | Batch Predict    | ion                           | <b>5</b> - (i)     |
| វវា           | Éŵ            |                   |                  |                      |                  |                               |                    |
| £ សា          | Diabetes di   | agnosis dataset - | sample (80.00°   | %)'s ens × ×         |                  | Search dataset                | 4                  |
| fr 💦 🛛 🖬      | Diak          | petes             | Tue, C           | 03 May 2016 09:32:46 |                  | Please enter 1 more character |                    |
| 215.7<br>size |               | 10<br>fields      | 614<br>instances | 10<br>models         |                  |                               |                    |
| Descri        | ption:        |                   |                  | ♥                    |                  |                               |                    |
| Prediction    | name:         |                   |                  |                      |                  |                               |                    |
| Batch Pre     | ediction of [ | Diabetes diagno   | sis dataset -    | sample with          | uli [*] | Rese                          | Predict            |

Figure 2.60: Select dataset for batch predictions

4. After you have selected the ensemble and the dataset, the batch prediction configuration options (see Subsection 2.6.3) will appear along with a preview of the prediction output, which is formatted as a comma-separated list of values (CSV format). (See Figure 2.61.) The default output includes all the fields in your prediction's dataset plus a last column containing the calculated predictions.

Note: BigML does not include the predictions' probability, confidence or expected error by default so you will have to configure your output file to include that information as explained in Subsection 2.6.3.6.

<u> </u>				
Diabetes diagnosis dataset - sample (80.00%)'s ens		Diabetes da	taset for predictions	×
Diabetes Tue, 03 May 201	6 09:32:46	ılî		Mon, 09 May 2016 09:09:03
215.7 KB         10         614         10           size         fields         instances         mode		5.4 KB size		
Description:	▣	Description:		$\overline{\mathbf{O}}$
<ul> <li>Configure</li> <li>Preview of the prediction file (using the type of each field)</li> <li>Pregnancies, Glucose, Blood pressure, Skinfold, Insul 123, 123, 123, 123, 123, 123, 123, 123,</li></ul>	lin,BMI,Diabe	tes pedigree,Age,Diat	etes,weight,Diabe	tes
123,123,123,123,123,123,123,123,123,ABC,ABC,ABC 123,123,123,123,123,123,123,123,123,ABC,ABC,ABC				

Figure 2.61: Configuration options displayed and output preview

5. By default, BigML generates an output dataset containing the batch prediction results. You find it in the BigML Dashboard's dataset list view and can use it as any other dataset to analyze the batch prediction output afterwards. If you do not want a dataset with all the prediction results to be created, you can deselect the button highlighted in Figure 2.62

ources Datasets Ensembles TClusters Anom	alies Associatio	Predictions	Tasks	Scripts 🗸
ilî kî	New Batch P	rediction		<b>5</b> - (
18 Pr				
Diabetes diagnosis dataset - sample (80.00%)'s ens		Diabetes d	lataset for predictions	x v
Diabetes Tue, 03 May 201	6 09:32:46	ılî		Mon, 09 May 2016 09:09:03
215.7 KB 10 614 10 size fields instances mode		5.4 KB size	10 fields	154 instances
Description:	$\odot$	Description:		$\odot$
Configure Preview of the prediction file (using the type of each field)				⊚
Pregnancies, Glucose, Blood pressure, Skinfold, Insul 123, 123, 123, 123, 123, 123, 123, 123,	lin,BMI,Diabetes	pedigree,Age,Dia	betes,weight,Diabete:	s 
rediction name:				

Figure 2.62: Create dataset from batch predictions

6. Once you are done configuring your batch prediction, click the predict green button to generate it. This process may take some time depending on the size of the input dataset. (See Figure 2.63.)

	New Pete	ch Prediction		1
11 52	New Bato	ch Prediction		۵. (
<u><u><u>6</u></u></u>				
Diabetes diagnosis dataset - sample (80.00%)'s er	IS X V	Diabetes data	set for predictions	× ×
Tue, 03 May	2016 09:32:46	ıl		Mon, 09 May 2016 09:09:03
	10 odels	5.4 KB size	10 fields	154 instances
Description:	◙	Description:		$\overline{\mathbf{O}}$
୍ଦ୍ଧୁ Configure				⊙
Preview of the prediction file (using the type of each field)			an matche bit data	_
Pregnancies, Glucose, Blood pressure, Skinfold, In: 123, 123, 123, 123, 123, 123, 123, 123,	sulin,BMI,Diabe	etes pedigree,Age,Diabet	es,weight,Diabete	5
123,123,123,123,123,123,123,123,123,ABC,ABC,ABC 123,123,123,123,123,123,123,123,123,ABC,ABC,ABC				
123,123,123,123,123,123,123,123,123,ABC,ABC,ABC 123,123,123,123,123,123,123,123,123,ABC,ABC,ABC 123,123,123,123,123,123,123,123,123,ABC,ABC,ABC				
123,123,123,123,123,123,123,123,ABC,ABC,ABC				
123,123,123,123,123,123,123,123,ABC,ABC,ABC				
123,123,123,123,123,123,123,123,ABC,ABC,ABC		_		

Figure 2.63: Create batch predictions

 After the batch prediction has been created, you will be able to download a file with all the instances found in your input dataset along with the prediction corresponding to each one of them. (See Figure 2.64.)

Sources Datasets Ensembles T Clusters Anomalies Assoc	eiations Predictions Tasks	Scripts 👻
a Satch Predictio	n of Diabetes dia	<b>5</b> . ()
Diabetes Diagnosis Dataset - Sample (80 💿	Diabetes Dataset For Predictions	谢
Output preview Pregnancies, Glucose, Blood pressure, Skinfold, Insulin, BMI, Diabe 8, 183, 64, 0, 0, 23, 3, 0, 672, 32, True, 1, True 5, 116, 74, 0, 0, 25, 6, 0, 201, 30, False, 1, False 10, 115, 0, 0, 0, 35, 3, 0, 134, 29, False, 1, True 8, 125, 96, 0, 0, 0, 0, 0, 0, 232, 54, True, 1, False 1, 189, 60, 23, 846, 30, 1, 0, 398, 59, True, 1, True 1, 103, 30, 38, 83, 43, 3, 0, 183, 33, False, 1, False 1, 103, 30, 38, 43, 30, 1, 0, 344, 31, True, 1, False 1, 101, 50, 15, 36, 24, 2, 0, 526, 26, False, 1, False	tes pedigree,Age,Diabetes,weight,Diabetes	

Figure 2.64: Download batch prediction output CSV file

8. If you did not disable the option to create a dataset, as explained above (see step 4), an Output dataset button will also be available to allow you to directly jump to the output dataset. (See Figure 2.65.)



Figure 2.65: View batch predictions output dataset

### 2.6.2.2.1 Batch Prediction with Images

BigML ensembles can be trained from images using extracted image features (Subsection 2.2.3). The input of a batch prediction is a dataset. So when creating a batch prediction with images, the dataset has to have the same image features used to train the ensemble. The image features are in the dataset used to create the ensemble.

	<b></b>		
	New Batch Pred	liction	<b>\$</b> - (
ରୀ 👫ର 🏄 🔟	* 2		
grape-strawberry texture	× •	Search dataset	*
	predict-images texture	predict	٩
A all C label	6 instances, 162 fields (160 numeric, 1 path, image), 1 non-preferred	Images texture           6 instances, 162 fields (160 num)	eric, 1 path, 1 image), 1 n
13.4 MB 162 size fields	instances models	predict-images resnet18 6 instances, 514 fields (512 num	eric, 1 path, 1 image), 1 n
Description:	$\overline{\mathbf{O}}$	predict-images 6 instances, 236 fields (234 num	eric, 1 path, 1 image), 1 n
		iris dataset - extended - samp 60 instances, 142 fields	ole (100.00%) - extende
Prediction name:		iris dataset - extended - samp	ole (100.00%) - extende

Figure 2.66: Batch prediction using an image dataset

As shown in Figure 2.66, the input for the ensemble batch prediction is selected as predict-images texture, which is a dataset consisting of six images and contains a set of extracted image features, *Wavelet subbands*.

Image features are configured at the source level. For more information about the image features and how to configure them, please refer to section Image Analysis of the Sources with the BigML Dashboard¹³[11].

¹³https://static.bigml.com/pdf/BigML_Sources.pdf

## 2.6.3 Configuring Ensemble Predictions

BigML provides several options to change its default behavior when calculating predictions. For single predictions as well as for batch predictions you can configure the strategy used for handling **missing values** (see Subsection 2.6.3.1.) In the case of Decision Forests you can also configure the **prediction operating kind**, i.e. the criteria used to combine the single tree predictions, confidence, probability or votes (see Subsection 2.6.3.2.) For classification ensembles you can also use a **threshold** for your single an batch predictions. For batch predictions you can configure the automatic **fields mapping** performed by BigML (Subsection 2.6.3.5), and define the **output file settings** (Subsection 2.6.3.6.)

## 2.6.3.1 Missing Strategies

When you create a new prediction, BigML will automatically navigate through the corresponding ensemble to find the leaf node that best classifies the new instance.

However, it may just so happen that your new data (the instances you want to predict) does not have populated values for all the fields used in building the original ensemble. For example, imagine that you are trying to predict diabetes and you have the patient's glucose level and BMI (Body Mass Index) but not his blood pressure. If the ensemble arrives at a node where the blood pressure level is required, BigML can handle this missing value by using one of these two strategies:

- Last prediction: it returns the prediction value and confidence of the parent node.
- **Proportional**: it combines all subtrees' predictions beneath the current node based on the data distribution of their child nodes in order to compute the prediction value and confidence.

For single predictions you can select either Missing strategies by clicking in the icons shown in Figure 2.67.



Figure 2.67: Missing strategies for single predictions

Sources Datasets Supervised - Unsupervised - Predi	ctions - Tasks	WhizzML 👻
A State New Ba	atch Prediction	<b>\$</b> - (i)
ଲ fa ∠ ≤		
Diabetes diagnosis dataset   Training (70%) v4 × ×	Diabetis diagnosis dataset   Test (30%)	×
Diabetes         Wed, 13 Dec 2017 17:58:27	Wed,	08 Jun 2016 15:39:39
18.9 KB         9         537         31           size         fields         instances         models	8.1 KB 10 size fields	231 instances
Description:	Description:	$\odot$
්සු Configure		۲
Select a positive class V	Probability threshold: 50%	
Default numeric value: Select a default value		
Excluded fields:		$\overline{\mathbf{O}}$
1 Make sure that the fields in the Ensemble and Dataset match		
Fields mapping:	Default fiel	ds 📀

For batch predictions you can find both options under the configuration panel as shown in Figure 2.68.

Figure 2.68: Missing strategies for batch predictions

#### 2.6.3.2 Combine single tree predictions: probability, confidence or votes

Ensembles are composed of several trees. Each tree returns a different prediction given an input data. These single predictions need to be combined to get a final prediction for the ensemble. For **Boosted Trees** there is only one way to combine the single tree predictions explained in Subsection 2.2.1.2, so you will not see any options in the prediction view. For **Decision Forests** there are three different options (called operating kinds in BigML API¹⁴) to combine the single tree predictions that compose the ensemble and get a final prediction:

#### Probabilities:

 For classification ensembles per-class probabilities are averaged taking into account all trees composing the ensemble. The class with the highest probability is the winner class. For example find below a classification ensemble built using three decision trees trying to predict two classes, "True" and "False":

Trees	True	False
Tree 1	80%	20%
Tree 2	40%	60%
Tree 3	60%	40%
Ensemble	60%	40%



¹⁴https://bigml.com/api/

The predicted class is "True" because it has a higher probability ([80% + 40% + 60%]/3 = 60%) than the "False" class ([20% + 60% + 40%]/3 = 40%).

 For regression ensemble the probability option averages the predictions of the trees composing the ensemble. For example, considering again three different trees in an ensemble but predicting a numeric output this time:

Trees	Total Sales	Expected Error
Tree 1	\$200	\$2.40
Tree 2	\$250	\$2.10
Tree 3	\$180	\$1.45
Ensemble	\$210	\$1.98

Table 2.3:	Example of	rearession	ensemble	usina	probability
14010 2101	Example of	10910001011	01100111010	aonig	probability

The final result will be a total sale of \$210 ([\$200 + \$250 + \$180]/3 = \$210) with an error of \$1.98 ([\$2.4 + \$2.1 + \$1.45]/3 = \$1.98).

- Confidences:
  - For classification ensembles per-class confidences are averaged taking into account all trees composing the ensemble. The class with the highest confidence is the winner class. It is calculated in the same way as the prediction in Table 2.2 but using the per-class confidences instead of the probabilities.
  - For regression ensembles the confidence option averages the predictions of the trees composing the ensemble in the same way that explained for probabilities in Table 2.3 but weighted by the expected error.
- Votes:
  - For classification ensembles each tree prediction is considered as one vote. The "votes" of a given class is the percentage of trees in the ensemble that vote for that class. You can find below an example that shows a classification ensemble built using three decision trees to predict two classes, "True" and "False":

Trees	Predicted class		
Tree 1	True		
Tree 2	False		
Tree 3	True		
Ensemble	True		

Table 2.4: Example of classification ensemble using votes

Since there are more trees that predicted the class "True" (two versus only one tree that predicted the class "False"), the final prediction is "True" with the 66.67% of votes (2/3 = 0.6667).

#### Note: if two or more classes have the same number of votes, the first class in alphabetical order will take precedence.

 For regression ensembles the votes option averages the predictions of the trees composing the ensemble. It gives the same results as the probability (see Table 2.3).



You can choose any option (probabilities, confidences or votes) from the BigML Dashboard to calculate the Decision Forests **single predictions** (see Figure 2.69) or **batch predictions** (see Figure 2.70).

Figure 2.69: Probabilities, confidences or votes for Decision Forest single predictions

urces Datasets Supervised - Unsupe	rvised - Predicti	ons 🔻 Tasks		WhizzML 🔻
1. II.	New Batc	h Prediction		<b>3</b> - (
£1 F£1 ∠ ≤				
Diabetes diagnosis dataset   Training (70%)	rl × •	Diabetis diag	gnosis dataset   Test (20%)	×
Diabetes Fri,	01 Dec 2017 20:01:35	I		Wed, 01 Jun 2016 16:17:02
18.9 KB 9 537 size fields instances	10 models	5.4 KB size	10 fields	154 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
់ត្នូ Configure				۲
Select app	sitive class <b>v</b> P	robability threshold: —	5	0%
Default numeric value:  Select a default value	¥			
Excluded fields:				$\overline{ullet}$
1 Make sure that the fields in the Ensemble and	d Dataset match			
			Defau	

Figure 2.70: Probabilities, confidences or votes for Decision Forest batch predictions

### 2.6.3.3 Probability, confidence, and votes thresholds

The thresholds are only available for **classification** ensembles, and it usually makes sense for unbalanced binary classifications, when you want to minimize false positives at the cost of false negatives. The positive class will be predicted if the probability, confidence or the votes are greater than the given threshold, otherwise the following class with greater probability, confidence or votes will be predicted instead.

To configure a threshold for your single predictions follow these steps:

 Select the probability, the confidence, or votes measure depending on the criteria you want to use for Decision Forests (see Figure 2.71). To learn more about these three ways to combine single tree predictions in the ensemble refer to Subsection 2.6.3.2. Boosted trees will only have the option to set a probability threshold.

Sources Datasets Su	ipervised - Unsupervised -	Predictions - Tas	sks	WhizzML 🗸
	Predict using	g Diabetes diagnos	is dataset   Trai	<b>८ॐ</b> ∽ (≡) [*] ∕∽ <b>(</b> )
	Diabete	s: True	56.91%	۲
	43.09%		56.91%	
	40 ⁶⁰		TUE	
Missing strategy:	Probability	lect a positive class	Probability threshold:	All input fields: 🕑
Glucose	34.98	% 🗹 BMI		<b>—</b> 22.65% <b>(</b>
0	248	124	0	83.88 <b>40.74</b>
Age		% 🗹 Diabetes pe	edigree	. 9.35%
8	84	46 0.0	6	3.01 <b>1.46</b>
Blood pressure	. 5.709	Skinfold		5.14%
°	142	71 0	0	78 <b>39</b>

Figure 2.71: Select probability, confidence or votes

Sources Dataset	ts Supervise	ed 👻 Unsupervised	Predictions -	Tasks	WhizzML 🚽
t and the second		Predict	using Diabetes diag	gnosis dataset   Trai	∽ (=)≁ (
		Dia	abetes: True	56.91%	
		43.09%		56.91%	
		40.00%			
		50 ³⁵⁶		THE	
Vissing strategy:	& &	5% <b>(</b> ?) <b>•</b> 0	Select a positive class	Probability threshold:	50%
Missing strategy:	s 8	19 (? d	Select a positive class	<ul> <li>Probability threshold:</li> </ul>	
Missing strategy:	æ &	(7)		Probability threshold:	All input fields: 🗹
	& &	<b>248</b>	False True	Probability threshold:	All input fields:
Glucose	& &		False True	Probability threshold:	All input fields:
Glucose 0	& &		False True 124	Probability threshold:	All input fields:
Glucose 0	& &   		False           True           124           15.18%           ✓           Diabet           0.0		All input fields:
Glucose 0 Age	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	248	False           True           124           15.18%           ♥           Diabet		All input fields:
Glucose 0 Age		248	False           True           124           15.18%           ✓           Diabet           0.0	tes pedigree	83.88 40.74

2. Select the **positive class**, i.e. the class for which you want to apply the threshold:

Figure 2.72: Select the positive class

3. Set a value for the **threshold** using the slider. The positive class will only be predicted when the probability, confidence or votes of the prediction is above the established threshold, otherwise the following class with higher probability, confidence or votes will be predicted instead.



Figure 2.73: Set a threshold

For batch predictions, you will find the same options under the CONFIGURE panel. (See Figure 2.74.)

Sources Datasets Supervised - Unsupervised	Predict	tions - Tasks		WhizzML 🔻			
New Batch Prediction							
n fn ∠ ≤							
Diabetes diagnosis dataset   Training (70%) v1	× •	Diabetis dia	agnosis dataset   Test (20%)	× ×			
Fri, 01 Dec 2	017 20:01:35	ıP	V	Ved, 01 Jun 2016 16:17:02			
	I <b>O</b> dels	5.4 KB size	10 fields	154 instances			
Description:	◙	Description:		$\overline{\mathbf{O}}$			
Configure				۲			
Select a positive cla	ass v F	Probability threshold: =	50	16			
Default numeric value: Select a default value							
Excluded fields:				$\overline{ullet}$			
1 Make sure that the fields in the Ensemble and Datas	et match						
Fields mapping:			Default	fields 🕤			

Figure 2.74: Configure a threshold for batchpredictions

### 2.6.3.4 Default Numeric Value

If the dataset used to make the batch prediction contains instances with **missing** values for the numeric fields you can easily replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero** using the **Default numeric value** before creating your batch prediction, (See Figure 2.75.)

Sources Datasets Supervise	d ▼ Unsupervised ▼ Prediction	new ons ▼ Tasks		WhizzML 🔻
a 17	New Ba	atch Prediction		ن ا
<u>क</u> 66 1				
Diabetes diagnosis datas	set's ensemble x v	Diabetes dia	agnosis dataset	×
Diabetes	Tue, 26 Jul 2016 20:43:53	ıß		Wed, 01 Jun 2016 14:29:52
27.0 KB 10 size fields	768 4 instances models	27.0 KB size	10 fields	768 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
Configure				۲
88 20	Select a positive class	Probability threshold:		50%
Default numeric value: 🕄 Sel	ect a default value			
Make sure that the fields in t				
Fields mapping:			Defa	ult fields 💿

Figure 2.75: Default numeric value for batch predictions

### 2.6.3.5 Field Mapping

By default, BigML maps fields based on their **names**. If there is a mismatch between the field names in your ensemble and those in the input dataset you selected for the batch prediction, you can specify the right correspondence between the two sets of fields by explicitly assigning to each field appearing in the "Ensemble fields" column its associated input field in the "Dataset fields" column. (See Figure 2.76.)

If the dataset's and ensemble's field names do not match but their IDs do, which happens when corresponding fields appear in the same order, you can tell BigML to use the **field ID** instead of the field name to map the fields. To do this, click the green switcher shown in Figure 2.76.

If you do not want some of the fields to be considered during the evaluation, you can also **manually** search for those fields and remove them from the "Dataset fields" column.

ources Datasets Ensembles <del>-</del> Clusters Anomalies As	ssociations Predictions Tasks Scrip	ts Ӻ
الأ. الأ. المن العام المن المن المن المن المن المن المن ال	Batch Prediction 🕉-	G
ன் சில		
Diabetes diagnosis dataset - sample (80.00%)'s ens × ×	Diabetes dataset for predictions *	¥
Coll O Diabetes Tue, 03 May 2016 09:32:46	Mon, 09 May 2016 09:05	9:03
215.7 KB 10 614 10 size fields instances models	5.4 KB 10 154 size fields instances	
Description:	Description:	)
්සු Configure		۵
Select a positive class •	Probability threshold: 50%	
Make sure that the fields in the Ensemble and Dataset match		
Fields mapping:	Apply this map	<b>)</b>
Ensemble fields	Dataset fields Auto-mapping by: NAME COM ID	
1. Pregnancies	1. Pregnancies	
2. Glucose 123	2. Glucose 123 × v	
3. Blood pressure 123	3. Blood pressure 123 × v	
4. Skinfold	4. Skinfold 123 × v	
5. Insulin 123	5. Insulin 123 × ×	

Figure 2.76: Fields Mapping for batch predictions

The fields mapping from the BigML Dashboard has a **limit of 200 fields**. For batch predictions with a higher number of fields, use the argument field_map from BigML API¹⁵ if you need to map your fields.

### 2.6.3.6 Output Settings

As mentioned, batch predictions can create a file containing all input instances along with the predictions BigML calculated for each of them. Define the following settings to customize your prediction file:

- **Separator**: this option allows you to choose a separator for your output file values. The default separator is the comma. You can also select the semicolon, the tab, or the space.
- **New line**: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".

¹⁵https://bigml.com/api/batchpredictions#bp_batch_prediction_arguments

• **Output fields**: this option allows you to include or exclude any of your dataset fields from the output file from the preview shown in Figure 2.77.

Note: a maximum of 100 fields are displayed in the preview, but all your dataset fields are included in the output file by default unless you exclude them.

- **Headers**: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability and/or confidence column name, field importances column names, single tree predictions column names, etc.). By default, BigML includes the headers.
- **Prediction column name**: this option allows you to customize the name for your predictions column. By default BigML uses the name of the ensemble's objective field.
- Probability, confidence (or expected error) or votes: this option allows you to include an additional column in the output file with the probability, the confidence (or expected error in case of regression trees) or the votes per instance. In the case of classification Boosted Trees you will always see the probability option and in the case of classification Decision Forests it will depend on the option selected to combine the single tree predictions (see Figure 2.70). These measures are not included by default in your batch predictions.

Note: remember that the expected error cannot be calculated for Boosted Trees, therefore this option will be disabled for regression Boosted trees.

- Probability, confidence (or expected error) or votes column name: this option allows you to customize the name for the probability, confidence (or expected error) or votes column in case you include it in the output file. By default BigML uses "probability", "confidence" (for both confidence and expected error) or "votes".
- Single tree predictions: this option allows you to include a column for each of the individual model predictions only in the case of Decision Forests. That will add a column per model, named <prediction_name>_n where n is the position of the model in the model list in the ensemble, starting at 1.
- All class confidences: this option allows you to include the probabilities for each class in the objective field for classification Decision Forests. There is a column per class, named "<class_name> confidence".
- All class votes: this option allows you to include the votes for each class in the objective field for classification Decision Forests and Boosted trees. There is a column per class, named "<class_name> votes".
- All class probabilities: this option allows you to include the probabilities for each class in the objective field for classification Decision Forests and Boosted trees. There is a column per class, named "<class_name> probability".
- **Importances**: this option allows you to include a column for each of the field relative importances for the ensemble predictions (for Boosted trees and Decision Forests). There is a column per field, named "<field_name> importance".

Separator:	New line:		
, (comma)	<ul> <li>Unix, Linux or OS X (LF)</li> </ul>	▼ \$≡	
Prediction column n	ame: Probability column name:	:	
a,b,c -,-,-	<b>{%</b>		
Output Fields:			
× Pregnancies	123 x Glucose	123 × Blood pressu	ure <b>123</b>
⊮ Skinfold	123 × Insulin	123 × BMI	123
× Diabetes pedigree	123 x Age	123 x Diabetes	ABC
123,123,123,123,123,123,	od pressure,Skinfold,Insulin,BMI,Diab	petes pedigree,Age,Diabetes,weigh	it,Diabetes

Figure 2.77: Output settings for batch predictions

## 2.6.4 Visualizing Ensemble Predictions

Visualization of ensemble predictions changes depending on whether you are predicting one **single** instance or you are predicting multiple instances using the **batch predictions** option. (See Subsection 2.6.4.1.)

### 2.6.4.1 Single Predictions

There are some essential differences between Decision Forests and Boosted Trees predictions, hence the visualizations for both of them differ. These differences are highlighted in the following paragraphs.

For single predictions you can find the prediction for your objective field at the top of the form along with the performance measure.

For **classification ensembles**, you will get the predicted class at the top and all the objective field class probabilities in the histogram below as shown in Figure 2.78.



Figure 2.78: Single predictions view for classification ensembles

For **regression Decision forest ensembles** you will get a numeric prediction and the expected error for that prediction as shown in Figure 2.79.

Sources Dat	tasets Supervis	sed 🗸 Unsupervis	sed - Predict	tions - Tasks		WhizzML 👻
<b>*</b> *		PI	redict using Fig	ctional Wine Sal	es v1	<b>نۇ</b> - ()
		То	tal Sales: 75.7	4	19.66	Expected error
Missing strategy	r: 🔥 🔏	[%]				All input fields: 🕑
Price			65.56%	Grape		<b>—</b> 17.06% <b></b>
0.0		42.0	20.4	Cabernet Sauvig	non	¥
Country		•	9.74%	Rating		5.91%
Argentina			¥	88		92 90
From Oregon?			1.73%			
No			Ŧ			
New prediction	name					
Fictional Wine	Sales v1					Predict

Figure 2.79: Single predictions view for regression ensembles

For **regression Boosted trees ensembles** you will get a numeric prediction (see Figure 2.80) but the expected error for that prediction cannot be calculated as in the case of Decision Forests (see Subsection 2.2.1).

Sources Datasets	Supervised 🔻	Unsupervised 👻	Predictions	▼ Tasks	WhizzML 🕶	
		E	Prediction f	or Total Sales	(=) [€] ,         (i)	
	Total Sales: 71.95					
		Missing strate	egy: 💦 🥈	3	All input fields: 🗹	
Price 0	0	42	≥ 60.28% <b>20.4</b>	Grape Cabernet Sauvignon	— 20.64%	
Country Argentina		•	) 15.00%	From Oregon?	2.18%	
88	0	92	90			
New prediction name Prediction for Total S	ales				Predict	

Figure 2.80: Single predictions view for regression Boosted Trees

In any of the above-mentioned cases, you can change any time the value of the displayed input fields to have your prediction recalculated in real-time.

If you have saved your prediction, you can go back to it and visualize it.

Read a detailed explanation of confidences, probabilities and expected error calculations in Subsection 2.2.1.2.

### 2.6.4.1.1 Prediction explanation

Prediction explanation helps understand why an ensemble makes a certain prediction. This is very useful in many applications, and the reasons behind an ensemble's prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

For any classification or regression ensemble, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Predict (see Figure 2.81).

Sources Datasets Supervis	ed  Vinsupervised  Predice	ctions - Tasks	_	WhizzML 🔻
The local prediction feature is only enabled	d for ensembles with up to 30 models.			х
<b>1</b>	Predict usin	ng Titanic Survival		<b>5</b> - ()
	Survived: ?		?	
Missing strategy:	Select a positive class	bability threshold:	50%	
Age	44.35%	Fare today	_	All input fields: 🗹
0.0	92.5 <b>44.93</b>	0	49438	24719
Class/Dept	■ 12.53%	Joined		6.38%
1st Class	v	Belfast		T
New prediction name		(	Click to compute the <b>explan</b> along with the prediction	
Titanic Survival			ē.	Predict

Figure 2.81: Explain prediction

The prediction explanation represents the most important factors considered by the ensemble in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 2.82. The importances across all input fields should sum 100%.

Sources Da	atasets Supen	vised	WhizzML 🗸
8 4.		Titanic Survival	<\$- (≡), (Ξ), (Ξ)
		Survived: TRUE	%
			đ
PREDICTION EXPL	ANATION		PNG CSV ISO
nput data		Importance	
Class/Dept	1st Class		45.08%
Joined	Belfast		24.39%
Age	44.93		16.04% +
			13.62% +



For some input fields you will see a "+" icon next to the importance. This is because the importance may not be directly associated with the input value, i.e., it can be explained by other reasons. In the Figure 2.83 below, the importance of 13.62% for the field "Fare today" is not explained by this field being equal to 24,719. Rather, it is because this field value is not missing (which accounts for an importance of 8.80%) and because it is higher than 17,000 (4.82% of importance).



Figure 2.83: See the detailed explanation

The prediction explanation for ensembles is calculated using the results of over a thousand distinct predictions using random perturbations of the input data. For this reason, the calculation of the explanation may take some time to be computed.

Note: the input field importances in the prediction explanation are different from the overall field importances of the ensemble. A field can be very important for the ensemble but insignificant for a given prediction.

### 2.6.4.2 Batch Predictions

For batch predictions, you always get a file and an optional output dataset.

## 2.6.4.2.1 Output File

From the batch prediction view, you can access the output file containing your **predictions** for each of your dataset instances in the last column. (See Figure 2.84.) You can configure several options to **customize your output file** including the separator for the columns, the name of your prediction column, the dataset fields you want to include, whether you want to include the a first row with the headers for your column names. You can find a detailed explanation of those options in Subsection 2.6.3.6.



Figure 2.84: Download batch prediction output file

See an output file example in Figure 2.85 where the two last columns contain the prediction and the confidence for each instance.

```
Pregnancies, Glucose, Blood pressure, Skinfold, Insulin, BMI, Diabetes, Confidence
8, 183, 64, 0, 0, 23.3, True, 0.6574
5, 116, 74, 0, 0, 25.6, False, 0.845
10, 115, 0, 0, 0, 35.3, True, 0.6469
8, 125, 96, 0, 0, 0.0, False, 0.9356
1, 189, 60, 23, 846, 30.1, True, 0.7574
1, 103, 30, 38, 83, 43.3, False, 0.675
7, 103, 66, 32, 0, 39.1, False, 0.7682
1, 101, 50, 15, 36, 24.2, False, 0.948
0, 100, 88, 60, 110, 46.8, False, 0.5413
```

Figure 2.85: An example of a batch prediction output file

### 2.6.4.2.2 Output Dataset

By default, BigML creates a dataset out of your batch prediction. (See Subsection 2.6.3.6.) You can access your output dataset from the batch prediction view by clicking the Output dataset button shown in Figure 2.87.



Figure 2.86: View batch predictions output dataset

In the output dataset you can find an additional **field** (named by default as per your ensemble's objective field) containing the **predictions** for each one of your instances. If you configured your batch prediction to include the **confidence** or **expected error** you will be able to find it in the last field of your output dataset as shown in Figure 2.87.

Sources Datasets	Ensembles - Clusters	Anomalies	Associations	Predictions	Tasks Scripts ▼
	••	Diabe	etes dataset fo	or pre	°⇔ - ∽ (=) ^{&gt;} - ①
					Q <b>X</b>
Name	\$ Туре	Count	Missing	Errors	Histogram
Pregnancies	123	154	0	0	
Glucose	123	154	0	0	
Blood pressure	123	154	0	0	
Skinfold	123	154	0	0	hanadili haa
Insulin	123	154	0	0	Information and a second
BMI	123	154	0	0	.addid Hutanaa
Diabetes	ABC	154	0	0	
confidence	123	154	0	0	
Show 10 ᅌ fields		1	to 8 of 8 fields		IC C 1 > XI

Figure 2.87: Batch predictions output dataset

### 2.6.4.2.3 Batch Prediction 1-Click Actions

From the batch prediction view you can perform the following actions (see Figure 2.88):

• BATCH PREDICTION AGAIN: this option will redirect you to the batch prediction creation view, with the same ensemble and prediction dataset already selected. This option allows you to rapidly recreate the batch prediction using a different configuration.

- BATCH PREDICTION WITH ANOTHER DATASET: this option allow you to easily create a batch prediction using the same ensemble and a different dataset.
- BATCH PREDICTION USING ANOTHER ENSEMBLE: this option allows you easily create a batch prediction using the same dataset and a different ensemble.
- NEW BATCH PREDICTION: this option redirects you to the batch prediction creation view where you can select a prediction dataset and an ensemble to create your prediction.

Sources Datasets Ensembles - Clusters	Anomalies /	Associations	Predictions	Tasks	Script	ts 🔻
	Batch Pred	iction of Dia	betes dia		<b>G</b> -	(
			Q	BATCH PREDICTION AGAI	IN	
A				BATCH PREDICTION WITH	ANOTHER DATAS	
Diabetes Diagnosis Dataset - Samp	le (80 🔍		Diab A	BATCH PREDICTION USIN	IG ANOTHER ENS	1
Output preview			E	NEW BATCH PREDICTION		F
	- 1' DUT -			DELETE BATCH PREDICTI	ON	
Pregnancies, Glucose, Blood pressure, Skinfold 8,183,64,0,0,23.3, True,0.6574639999999999 5,116,74,0,0,25.6,False,0.844936 10,115,0,0,0,35.3, True,0.646888333333333 8,125,96,0,0,0,7alse,0.935643 1,189,60,23,846,30.1, True,0.7574466666666666 1,103,30,38,83,43.3,False,0.674905555555555 7,103,66,32,0,39.1,False,0.76819 1,101,50,15,36,24.2,False,0.948028000000000	6 5	rubetes, com				
Download batch prediction			ılı	Output dataset		

Figure 2.88: Batch prediction 1-click actions

## 2.6.5 Consuming Ensemble Predictions

BigML provides plenty of means for developers to integrate BigML ensemble predictions within their apps. In the following sections, we will describe how you can use BigML REST API and BigML Python bindings to work with ensemble predictions.

### 2.6.5.1 Using Ensemble Predictions via the BigML API

Ensemble predictions have full citizenship in the BigML API. This means you can programmatically create, update, list, delete, and use them for predictions. For example, this is how you can create a single prediction using the command line from a given ensemble and defining the input data. This will require properly setting the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \ -X POST \ -H
'content-type: application/json' \ -d '{"ensemble":
    "ensemble/50650bdf3c19201b64000020", "input_data": {"000001": 3,
    "000002":4.5, "000003"}}'
```

For more information on using ensemble predictions through the BigML API, please refer to prediction REST API documentation¹⁶.

#### 2.6.5.2 Using Ensemble Predictions via the BigML Bindings

BigML bindings provide a convenient way to access the BigML REST API from your language of choice. They offer a higher-level view of BigML Machine Learning resources and algorithms in a number of languages, including Python, Node.js, Java, Swift, and Objective-C. For example, this is how you can create an ensemble prediction in Python using BigML bindings:

¹⁶https://bigml.com/api/predictions

BigML bindings also provide the means to carry through predictions locally, without ever hitting the network, which can greatly improve the latency of predicting from your apps. This is made possible by BigML ensembles being white-box, meaning you can download them and use them independently from BigML. For example, the following code snippet shows how you can download an ensemble and use it for making a local prediction using BigML bindings for Python:

For more information on using ensembles through the BigML API, please refer to BigML bindings documentation.

## 2.6.6 Descriptive Information

Descriptive information is what allows you to describe a prediction so you can find it later and easily recognize it among other predictions.

Each prediction is associated with **name**, **description**, **category**, and **tags**. See the following sections for a brief description of each concept. In Figure 2.89, you can see the possibilities that the MORE INFO menu option gives to edit them.



Figure 2.89: Edit predictions

#### 2.6.6.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: the name always follows the structure "Prediction for <objective field name>"
- **Batch predictions**: BigML combines your prediction dataset name and the ensemble name: "Batch prediction of <ensemble name> with <dataset name>".

Predictions names are displayed on the list view and also on the top bar of a prediction view. Predictions names are indexed to be used in searches. You can rename your predictions at any time from the **More info** panel. The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, since they are automatically assigned unique internal identifiers.

#### 2.6.6.2 Description

Each ensemble prediction also has a **description** that it is very useful for documenting your Machine Learning projects. Predictions take the description from the ensembles used to create them.

Descriptions can be written using plain text and also markdown¹⁷. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 2.90.)

Edit description	×
You can add formatting and links using a simple markdown language: B $I := 400$	
Write your **description** here	
Description:	
Write your description here	
Cancel	Update
Cancer	opuale

Figure 2.90: Markdown editor for evaluations descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

#### 2.6.6.3 Category

Each prediction is associated with a **category** taken from ensemble used to create it. Categories are useful to classify predictions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 2.5.

¹⁷https://en.wikipedia.org/wiki/Markdown

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 2.5: Categories used to classify predictions by BigML

## 2.6.6.4 Tags

A prediction can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the ensemble use to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

## 2.6.7 Ensemble Predictions Privacy

The link displayed in the MORE INFO panel is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared from your BigML Dashboard by sharing a link, as you can do with other resources.

Sources	Datasets	Models 🔻	Clusters	Anomalies	Associations	Predictions	Tasks	Scrip	ots ▼
s s	ılî	få		P	rediction for D	liabetes		<b>5</b> -	6
DETAILS									⊘
INFO									ਂ
PRIVACY									۹
Private lin	k  Only yo	u can view this pr	ediction logged	l into your accou	nt.				
				https://tropo.d	ev.bigml.com/dasl	nboard/prediction	/5730d593b95b394f44000086		

Figure 2.91: Private link of a prediction

# 2.6.8 Moving Ensemble Predictions to Another Project

When you create a prediction it will be assigned to the same **project** where the original ensemble is located. You cannot move predictions between projects as you do with other resources.

## 2.6.9 Stopping Ensembles Predictions

**Single predictions** are synchronous resources, so you cannot cancel them during the creation since you get the result immediately.

**Batch predictions** are asynchronous resources, so you can stop the creation before the task is finished. You can use the DELETE option from the 1-click action menu (Figure 2.92) or from the pop up menu on the prediction list view. (See Figure 2.93.) You can see in Figure 2.93 that the objective field column has the label PROCESSING to indicate the batch prediction is still in progress. If you stop the prediction during its creation, you will not be able to resume the same task again, so if you want to create the same prediction, you will have to restart a new task.

Sources Datasets Models - Clust	ers Anomalies Associations	Predictions Tasks	Scripts -
	Prediction for Tota	al Sales	> <b>८५</b> - (≡) ^{№₩} ()
10 out (	if 10		
		BATCH PREDICTION	
		EVALUATE	
		PARTIAL DEPENDENCE PLOT	
		DELETE PREDICTION	
		MOVETO	
		5	

Figure 2.92: Stop prediction from the 1-click menu

Sources	Datasets Models ▼ Clusters Anon	nalies Ass	sociations <b>Predictions</b>	Tasks	Scripts 👻
ŝ			Predictions		Q
វា	Name	\$	Objective	Prediction	≎ 🛗 ≎
к <mark>9</mark>	Prediction for Species	۲	Species	Iris-versicolor	5m
68	Prediction for Surv		Survived	1	5m 1w
к ⁹	Prediction for ratins		rating	3.57752	6m

Figure 2.93: Stop prediction from the predictions list view

# 2.6.10 Deleting Ensemble Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the 1-click action menu (see Figure 2.94) or using the pop up menu on the predictions list view (see Figure 2.95.)

		r Diabetes	<u>ن</u> الح
	Diabetes: FAI	PREDICT QUESTION BY QUESTION	
	Diabetes: PAI		
	Missing strategy: 💦 💦		All input fields:
Glucose	29.21% 🕑 BI	MI	22.96%

Figure 2.94: Delete prediction from the 1-click menu

Sources	Datasets M	odels 🔻	Clusters	Anomalies	Ass	ociations	Predictions	Та	sks	Scripts	(NEW)
នា	¥. X.	+	Ē			Predictio	ns				Q,
វារ	Name				\$	Objective		٥	Prediction	\$ ÷	\$
к <mark>е</mark>	Prediction for Spec	ies			8	Species			Iris-versicolor		5m
ŕå	Prediction for Survi		EW DETAILS			Survived			1	51	m1w
KI.	Prediction for rating		ELETE PREDICTIO	NC		rating			3.57752		6m

Figure 2.95: Delete prediction from popup menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted and there is no way you (or even the IT folks at BigML) can retrieve it.



Figure 2.96: Delete prediction confirmation

# 2.7 Consuming Ensembles

In the previous sections, we have described how you can create ensembles, configure them, use them to make predictions, and more. This section will introduce a number of BigML features that enable interesting ways of taking advantages of ensembles: exporting them locally, and using them programmatically via the BigML REST API and Bindings.

## 2.7.1 Exporting and Downloading Ensembles

You can export your ensemble in a number of programming languages, including Python, Java, and Node.js. Just click on the download icon in the top menu and select your preferred option.

The main goal of downloading your ensemble in a programming language is to make local predictions faster and at no cost. (See Figure 2.97.)

	ATE REPLACEMENT		INSTAN DER	Downloa ensemb
ble Ensemble Download		×		ensemb
nable model:		5 d.		
* *		005		
ngs -				
Or download it from: {@link htt	ps://github.com/bigmlcom/bigm	<u>.</u>		elect
ce you have the source code, you	can install it using maven:			guage
code>mvn install	den den est			
	rgs ou can get it from: {@link <u>https:</u> - Clone it: ccode-git clone <u>https</u> - git- gr download it from: {@link http /archive/moster.zig}	onuble model: ou con get it from: {@link https://github.com/bigmlcom/bigml- - Clone it: <code> - Clone it: <code></code></code>	anable mode: anable model:	able Ensemble Download  anable model:  anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model: anable model:

Figure 2.97: Download your ensemble

## 2.7.2 Using Ensembles Via the BigML API

Ensembles have full citizenship in the BigML API. This means you can programmatically create, update, list, delete, and use them for predictions. For example, this is how you can create en ensemble from the command line with custom values for a few available arguments. This will require you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/ensemble?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"dataset": "dataset/4f66a80803ce8940c5000006",
        "name": "my ensemble",
        "number_of_models": 32}'
```

For more information on using ensembles through the BigML API, please refer to ensemble REST API documentation.

## 2.7.3 Using Ensembles Via the BigML Bindings

BigML bindings provide a convenient way to access BigML REST API from your language of choice. They offer a higher-level view of BigML Machine Learning resources and algorithms in a number of languages, including Python, Node.js, Java, Swift, and Objective-C. For example, this is how you can create en ensemble in Python using BigML bindings:

For more information on using ensembles through the BigML API, please refer to BigML bindings documentation.

# 2.8 Ensemble Limits

BigML imposes a few limits on the characteristics of an ensemble that it can handle. Some of them are the same that applies to models.

- **Number of trees**: A maximum of 1,000 trees are allowed for Decision Forests and a maximum of 2,000 trees for Boosted Trees.
- Number of iterations: A maximum of 1,000 iterations are allowed for Boosted Trees.
- Fields: There is no enforced limit to the number of fields that can be present in a model.
- Instances: There is no enforced limit to the number of instances that can be handled.
- Classes: A maximum number of 1,000 distinct classes per field is allowed.
- **Terms-tokens**: BigML can handle up to 1,000 tokens total. In case multiple text fields are defined, then the token limit per field is divided by the number of text fields.
- Terms-full terms: BigML can handle up to 256 characters total.
- Items: A maximum of 10,000 items per field is allowed.
- Node threshold: BigML supports a value between 3 and 2,000.

# 2.9 Descriptive Information

Each ensemble has an associated **name**, **description**, **category**, and **tags**. In Figure 2.98, you can see the options that the **More info** panel gives to edit them.

Sources Datasets Ensembles -	Clusters Anomalies Associations Predictions Ta	sks Scripts -
	wines dataset's ensemble	ⓐ ♤ - ⇒ ⓐ
DETAILS		$\odot$
INFO		۲
Name: wines dataset's ensemble	Category: Miscellaneous	v
Description:	Tags:	,
	Total Sales	,
PRIVACY		⊘



## 2.9.1 Name

Each ensemble has an associated **name** that is displayed on the list view and also on the top bar of an ensemble view. Ensemble names are indexed to be used in searches. When you create an ensemble, by default, it gets the name of the file that you used to create it You can edit it using the MORE INFO menu option on the right corner of the ensemble view (see Figure 2.98).

The name of an ensemble cannot be longer than **256** characters. There is no restriction on the characters that can be used in an ensemble name. More than one ensemble can have the same name even

within the same project, since they are automatically assigned unique internal identifiers.

## 2.9.2 Description

Each ensemble also has a **description**, taken from the dataset used to create them, that it is very useful for documenting your Machine Learning projects. Descriptions can be written using plain text and also markdown¹⁸. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 2.99.)

You can add formatting and links using a simple markdown language: B $I \equiv {}^{_{\tiny{\tiny{\rm CD}}}}$	
Write your **description** here	
Description:	
Write your <b>description</b> here	

Figure 2.99: Markdown editor for ensemble descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

### 2.9.3 Category

Each ensemble has an associated category inherited from the dataset used to create it. Categories are useful to classify ensembles according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

An ensemble category must be one of the categories listed on table Subsection 2.9.3.

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 2.6: Categories used to classify ensembles by BigML

## 2.9.4 Tags

An ensemble can also have a number of **tags** associated with it that can help to retrieve it via BigML's API or to provide ensembles with some extra information. Each tag is limited to a maximum of 128 characters. Each ensemble can have up to **32** different tags.

## 2.9.5 Counters

For each ensemble, BigML also stores a number of counters to track the number of other resources that have been created using it as a starting point. In the ensemble view, you can see a menu option that displays these counters. It also allows you to quickly jump to all the resources of one type that have been created with this ensemble as shown in Figure 2.100.

Sources	Datasets	Supervised - Unsupervised -	Predictions <del>-</del>	Tasks		WhizzML 🔻
a 🖓	· ·	US car ac	cidents datase	et   Training (80%)	â 🗘	ૐ- (≡)⁵- ③
_	🔁 1  🗎	/IEW 16 PREDICTIONS FROM THIS ENSEMBL				
	🔄 7 v	IEW 7 BATCH PREDICTIONS FROM THIS ENS	FIELD		ТҮРЕ	INSTANCES
10	6	VIEW 6 EVALUATIONS FROM THIS ENSEMBLE	ERITY	X NO	DECISION FOREST	3,999
	82 O N	O FUSIONS THAT INCLUDE THIS ENSEMBLE				2

Figure 2.100: Menu option to quickly access to resources created with an ensemble

# 2.10 Ensemble Privacy

Privacy options for an ensemble can be defined in the More Info panel, displayed in Figure 2.101.

There are two levels of privacy for BigML ensembles:

- **Private**: only accessible by authorized users (the owner and those who have been granted access).
- Shared: by enabling the secret link you will get two different links to share your ensemble. The first one is a sharing link that you can copy and send to others so they can visualize and interact with your ensemble chart. The second one is a link to embed your model directly on your web page.

You can also share the individual component models (see Section 1.11).

Sources Datasets Supervised -	Unsupervised  Predictions  Tasks Wh	izzML 🔻							
a <b>™ 6</b> 0 • • • • •	🖄 Diabetes diagnosis dataset el's ensemble 🛛 🔋 🐼 🧭 🔅	¢. ().							
DETAILS		$\overline{\mathbf{O}}$							
INFO		ਂ							
PRIVACY		۲							
Private link   Only you can view this ensemble	logged into your account.								
Secret link   Only people with the secret link of	an view this ensemble.								
	https://bigml.com/shared/ensemble/48IsPeIQJYg0dvCuHHweqT43Z17								
	<iframe fra<="" src="https://bigml.com/embedded/ensemble/48IsPeIQJYg0dvCuHHweqT43Z17" td=""><td></td></iframe>								

Figure 2.101: Ensemble privacy options

# 2.11 Moving Ensembles to Another Project

When you create an ensemble it will be assigned to the project where the original dataset used to create it belongs to.

Ensembles can only be assigned to a single project. However, you can move ensembles between projects. The MOVE TO menu option to do this can be found in two places:

1. In the ensemble list view, within the 1-click actions for each ensemble (Figure 2.102).

Sources Datasets	Ensembles - Clus	sters Anomalies	Associations	Pre	dictions	Tasks			Scripts -
a s² dì	8 <b>m</b> ∎	wines	dataset's ens	emb	le	Ē	¢	∽.	(≡)∕¯ <b>(</b> ]
						REDICT			
MODELS	SAMPLE RATE	REPLACEMENT		ENS	б 🔚 в	ATCH PREDICTION			
10	100%	YES	×	BA	E	/ALUATE			
1	ma	odel/572cb144b95b39	263d600000b		P.	ARTIAL DEPENDENCE PL	.OT		
					ra D	ELETE ENSEMBLE			
$ \rightarrow $	Da	NEW PROJECT			<b>₽</b>				
••		PROJECT 6							

Figure 2.102: Menu option to move ensembles

2. Within the one-click actions of an ensemble in the ensemble list view as you can see on Figure 2.103.



Figure 2.103: Menu option to move ensembles from the ensemble list view

In both cases, by selecting the MOVE TO menu option, you will be displayed a list of existing projects that you can assign your resource to. Alternatively, create a new project for your ensemble using the NEW PROJECT menu option.

# 2.12 Stopping Ensemble Creation

You can also stop an ensemble creation process while BigML is not yet done with it from the one-click actions menu (Figure 2.104). This can be useful, e.g., when you are creating an ensemble from a large dataset, or an ensemble with many nodes, and later realize that you should have done something differently, be it when configuring the resource, preparing the dataset, etc.



Figure 2.104: Menu option to stop an ensemble creation

A modal window (Figure 2.105) will be displayed asking you for confirmation.

Sources Datasets	Ensembles - C	Clusters Anomalies	Associations	Predict	ions Tasks		Scripts 🔻
	8 <b>m</b> ↓ ••••	Churn in the t	elecom indust	ry datas	set - samp	<b>i</b> 🗘	(=)√ ()
				\$	PREDICT		
MODELS	SAMPLE RATE	REPLACEMENT		ENS Fri	BATCH PREDICTION		
15	100%	YES		BA	EVALUATE		
1		nodel/5757db1d1a632l	b2ob800027b		PARTIAL DEPENDEN	CEPLOT	
	<u> </u>	1006/3737001012032	000000270	fs	DELETE ENSEMBLE		
1 ALCON	i î î	Data distribution:			MOVE TO		
				_			

Figure 2.105: Menu option to stop an ensemble's creation

Note: if you stop the model during its creation, you will not be able to resume the same task. If you want to create the same model, you will have to start a new task.

# 2.13 Deleting Ensembles

You can delete your ensembles in two ways:

• From the ensemble view, using the DELETE ENSEMBLE option from 1-click action menu.(See Figure 2.106.)



Figure 2.106: Menu option to delete a ensemble

• Using the DELETE ENSEMBLE pop up menu option on the ensemble list view. (See Figure 2.107.)

Sources E	Datasets Ensembles  Clus	sters And	omalies	Associations	Prediction	s Tasks			Scr	ipts –
			- É é	Ensembles	6					Q
Name	9	\$	Туре 🗘	Objective	\$	<b>#</b> \$	<b>ê</b> \$	\$	\$	© <b>®</b> ⊖ ≎
wines	s dataset's ensemble	8	$\checkmark$	Total Sales		1h 28min	7.1 KB	0	0	0
Show 10 ᅌ			1 t	o 1 of 1 ensembles					K < 1	> >1
	CREATE BATCH PREDICTIO	N								
	EVALUATE									
	VIEW DETAILS									
	PR DELETE ENSEMBLE									
	MOVE TO	3	•							

Figure 2.107: Ensemble deletion pop up menu option

A modal window (see Figure 2.108) will be displayed asking you for confirmation. Once you delete an ensemble, it is deleted permanently and there is no way you (or even the IT folks at BigML) can retrieve

it.



Figure 2.108: Ensemble deletion modal window

# 2.14 Takeaways

This chapter explains ensembles in detail. See below a list of key points:

- An ensemble is a collection of decision tress which are combined together to create a stronger model with better predictive performance.
- Ensembles are one of the best performing Machine Learning algorithms, often winning Machine Learning competitions across a multitude of domains and use cases.
- Ensembles are very fast to train and test, which significantly streamlines real-life Machine Learning projects.
- You can use ensembles to solve classification and regression problems.
- BigML provides three types of ensembles: Bagging (a.k.a. Bootstrap Aggregating), Random Decision Forests and Boosted Trees. Bagging builds each model from a random subset of dataset. Random Decision Forests adds an additional element of randomness by choosing random features at each split. Boosted Trees iteratively builds each single model trying to learn from the previous model mistakes.
- You can build ensembles from datasets that have been created in BigML. (See Figure 2.109)
- An ensemble can be an input to an evaluation, to a prediction, or to a batch prediction. (See Figure 2.109).
- BigML ensembles support any type of fields as input fields (categorical, numeric, text and items fields).
- You can create an ensemble with just 1-click or configure it as you wish. Ensembles are virtually parameter free, giving excellent results with no tuning.
- If you don't specify any objective field, BigML will take the last valid field in your dataset.
- The default number of models for your ensemble is set to 10, and the maximum allowed from the Dashboard is 1,000 for Decision Forests and 2,000 for Boosted Trees.
- In BigML, you can choose three different pruning strategies when building your ensemble: smart
  pruning, statistical pruning, or no statistical pruning.
- By default, BigML ensembles don't consider missing values when choosing splitting rules, but you can explicitly include them.
- BigML provides three different options to assign specific weight your instances: balance objective, objective weights, weight field.

- They inherit all the good qualities of individual trees including handling missing data and speed of prediction. However, they are not as easy to interpret as a single decision tree.
- You can visualize your ensemble using the ensemble chart. The chart is a graphic representation of the marginal effect a subset of input fields have on the objective field (ensemble predictions) disregarding the rest of the fields.
- As with individual decision trees, the field importance for ensembles provides a measure of a field importance on predictions relative to the others.
- You need to evaluate your ensemble's performance with data that the ensemble has not seen before.
- For Decision Forests, the final prediction and, probability,confidence (or expected error), or votes is not known until all the component tree predictions are combined with the selected voting strategy (a.k.a. operating kinds in BigML).
- BigML provides three different strategies to combine the final Decision Forests prediction: probability, confidence and votes.
- Predictions for Boosted ensembles do not use combiners since the final rprediction is an additive process rather than averaged.
- For classification Boosted Trees the probability of each class is returned at the prediction time while regression Boosted Trees do not have any accuracy measure for predictions.
- You can predict single instances or multiple instances in batch using your ensemble.
- BigML provides local predictions from the Dashboard for single instances, which allow you to get a real-time prediction without consuming any credits or requiring any internet connection.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. For batch predictions, you always get a CSV file and an optional output dataset.
- You can download an ensemble in a number of programming languages including Python, Java, Node.js, and Objective-C, among others, to use it in your local environment, and make predictions faster at no cost.
- You can furnish your ensemble with descriptive information (name, description, tags, and category).
- You can move an ensemble between different projects.
- Ensembles cannot be shared, but you can share the individual component models.
- You can stop the ensemble creation before the task has finished.
- You can permanently delete an ensemble.


Figure 2.109: Model Workflows



# **Linear Regressions**

# 3.1 Introduction

There are multiple Machine Learning problems that can be solved using supervised Machine Learning techniques. These problems require predicting an output variable (objective field) given a number of input variables (input fields). They can be divided into classification and regression depending on whether you need to predict a category (label or class) or a continuous value (a real number), respectively. To learn more about concrete use cases for both problems refer to Section 1.1.

Linear Regression is a supervised Machine Learning technique that can be used to solve regression problems. These problems can also be solved with other Machine Learning methods, such as models, ensembles, or deepnets. These methods are explained in Chapter 1, Chapter 2, and Chapter 5 respectively. Depending on the problem you are trying to solve and the data available, some techniques may perform significantly better than others. The main difference between linear regression and others is that linear regression assumes your objective field has a linear relationship with your input fields. For this reason, linear regressions work best in those problems where this assumption is accurate.

This chapter contains a comprehensive description of BigML's linear regression models including how they can be created with 1-click (Section 3.3), all configuration options available (Section 3.4), and the different visualizations provided by BigML (Section 3.5). See Section 3.6 for an explanation of how linear regressions can be used to make predictions. You can also export your linear regressions in different formats to make local predictions faster at no cost (Subsection 3.6.4.1). The process to evaluate your linear regressions' predictive performance in BigML is explained in a different chapter (Chapter 7).

In BigML, the third tab (Supervised) of the main menu of the Dashboard allows you to list all of your available linear regressions. The linear regression list view (Figure 3.1), details the Dataset used to create it, the Name, Objective (objective field field name), Age (time elapsed since it was created), Size, and number of evaluations, predictions, and batch predictions that have been created using that linear regression. The SEARCH menu option in the top right corner of the linear regression list view allows you to search your linear regressions by name.

Source	es Datasets Supervised -		lictions 👻 Task:	5			WhizzN	1L <del>-</del>
:/:		Line	ar Regressions			8	8 🛍	Q,
III	Name	(	Objective	tj 🛗 tj	ê ţ	<b>\$</b>		08
ılı	Prediction of lending club monthly pay bias	yment i	nstallment	5min	4.3 MB			
ılî	Predicting Wine Quality bias	c	quality	36min	258.2 KB			
ılî	airbnb price prediction bias	t	orice	40min	13.8 KB			
ılî	<b>iris dataset</b> bias	t	betal width	13h 1min	4.5 KB			
ılî	Predict plasma glucose from patient o	lata p	blasma glucose	13h 5min	25.6 KB			
ılî	<b>automobile miles per gallon</b> bias	r	npg	13h 10min	17.7 KB			
ılî	Predicting abalone age from measure bias	ments	Class_number_of_ring	ıs 13h 17min	191.6 KB			
ıß	<b>Loan_risk_data</b> bias	c	credit_amount	18h 14min	135.8 KB			
ılî	US car accidents linear regression bias	F	Fatalities in crash Fata	1d21h	771.2 KB			
ılî	Hotel_reviews bias	F	Rating_Percentage	2w 6d	1.2 MB			
Show 1	0 ᅌ linear regressions	1 to 10 of 1	0 linear regressions				K < 1	> >1

Figure 3.1: Linear regression list view

By default, when you first create an account at BigML, or every time that you start a new project, your list of linear regressions will be empty. (Figure 3.2)

🔀 🔊 Linear Regressions 🔯 🛍					
Name Name	Objective 🖞 🛗				
	No linear regressions				

Figure 3.2: Empty Dashboard linear regression view

Finally, in Figure 3.3 you can see the icon used to represent a linear regression.



Figure 3.3: Linear Regression icon

# 3.2 Understanding Linear Regressions

As mentioned in the introduction of this chapter, linear regression is a supervised learning algorithm used to solve regression problems. It's simple to understand and has very good interpretability.

Linear Regression assumes a linear relationship between the input fields, also called predictors, and the single objective field, or the output variable. More specifically, the objective field can be modeled from a linear combination of the input fields:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots + \beta_n x_n$$

This is the linear regression formula, also called linear equation, where y is the objective field,  $(x_1, x_2, \ldots, x_n)$  represent the n variables, also called predictors, for the input fields in the input data, and  $(\beta_1, \beta_2, \ldots, \beta_n)$  are the coefficients which are the scale factors assigned to the respective variables. The one additional coefficient  $\beta_0$  is often called the intercept or the bias coefficient.

Learning a linear regression model means estimating the values of the coefficients with the data available. A **positive coefficient** ( $\beta_i > 0$ ) for a input field, indicates a positive correlation between the input field and the objective field, while a **negative coefficient** ( $\beta_i < 0$ ) indicate a negative correlation. Higher absolute coefficient values for a field results in a greater influence of that field on predictions. When a coefficient becomes zero, it effectively removes the influence of the input field on the model and hence on the predictions.

BigML Linear Regression produces an estimate for the coefficient values  $\beta_0, \beta_1, ..., \beta_n$  using a least-squares fit on the training data.

BigML provides a table containing all your linear regression coefficients. (See Subsection 3.5.2)

When the linear regression has learned the coefficients, you can use the model to make **predictions** for new instances. In a prediction, the linear regression returns a predicted value of the objective field.

By definition, the **input fields**  $(x_1, x_2, \ldots, x_n)$  in the linear regression formula need to be numeric values. However, BigML linear regressions can support any type of fields by applying a set of transformations to categorical, text, and items fields. Moreover, BigML can also handle missing values for any type of field. The following subsections detail both behaviors.

# 3.2.1 Input Field Transformations

Apart from numeric fields, BigML linear regressions are optimized to support categorical, text and items fields by applying a set of transformations in order to convert them to numeric values:

• **Categorical fields** are **Dummy** encoded by default. Dummy encoding converts one n-class categorical field to n separate variables. One of the classes is designated as the reference or dummy class, which is assigned a value of 0 for each variable. The dummy class, if not specified by the user, is the first class value in lexicographic order. So there are n-1 variables (n classes minus the dummy class); one additional variable for missing values is created if the training dataset contains missing values for this field. For a given instance, the variable corresponding to the instance's categorical value has its value set to 1 (except if the categorical value is the dummy class), while the other variables are set to 0.

BigML also provides **Contrast** coding and **Other** coding, that you can configure for each of your categorical fields. See Subsection 3.4.6 for a complete explanation of categorical fields encoding.

- For text fields, each term is mapped to a corresponding numeric variable, whose value is the number of occurrences of that term in the instance. Text fields without term analysis enabled are excluded from the model (read the Sources with the BigML Dashboard document to learn more about text analysis [11]).
- For **Items fields**, each different item is mapped to a corresponding numeric field, whose value is the number of occurrences of that item in the instance.

# 3.2.2 Missing Values

BigML linear regressions can handle missing values for any type of field. For categorical, text, and items fields, missing values are always included as another category, term or item by default.

For numeric fields, missing values are always included. If a field in the training data contains missing data, then a corresponding binary-valued predictor will be created which takes a value of 1 when that field is missing in a particular row, and 0 otherwise. The other predictors pertaining to that field will have a value of 0 when the value is missing. Once the linear regression is created, you can find an additional coefficient for each field at the end of the coefficient table. (See Figure 3.4) Learn more about the coefficient table in Subsection 3.5.2.

Alternatively, you can replace your missing numeric values by another valid value like the field's mean, median, maximum, minimum or zero (see Subsection 3.4.3).

If the input data does not contain missing values for a field, the coefficient for missing values will be zero, except in the case of text fields which can be different from zero. This is due to the fact that BigML has a limit of 1,000 terms for text fields, so there may be instances not containing any of the terms considered to build the model and appear as missing values instead. (See Subsection 3.8.0.1 to know more about term limits for text fields.)

Sources Datasets Su	Ipervised - Unsupervised -	Predictions - Task	s	WhizzML 🔫
	Predicting	abalone age from mea	surements	
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIEL		FIELD CODINGS YES
	VARIABLES: Filter bias or predi	ctors	SIGNIFICANCE LEVEL:	0.1 💌
Bias and predictors		ђ. Туре	Coefficients	ţ.
Bias		123	3.90015	s a
Sex = M		ABC	0.05522	<u>∧</u> σ
Sex = I		ABC	-0.82452	σ
Sex = F		()) ABC	0	
Length		123	-0.47157	σ
Diameter		123	11.07550	σ
Height		123	10.73700	σ
Whole_weight		123	8.98378	σ
Shucked_weight		123	-19.78820	σ
Viscera_weight		123	-10.57760	σ
Shell_weight		123	8.73112	s o
Sex		missing	1.88682	σ
Diameter		missing	2.90388	s a
Show 25 ᅌ variables		1 to 13 of 13 variables		

Figure 3.4: Missing numeric coefficients at the end of linear regression table

# 3.2.3 Number of Predictors

Because of input field transformations and missing values, one input field may become more than one predictors in the linear equation to fit the data. This table summarizes the nubmer of predictors which will be generated for each input field type.

Input field type	No Missing Value	Missing Values
Numeric	1	2
Categorical, dummy-encoded	(number of classes) - 1	number of classes
Categorical, contrast or other-encoded	number of classes	(number of classes) + 1
Text	number of terms	(number of terms) + 1
Items	number of items	(number of items) + 1

Table 3.1: Number of predictors per input field

Bias term, which is also called intercept term and enabled by default, is a predictor.

For example, if the data has 10 numeric input fields, 2 of them have missing values, it will generate 8 + 2 * 2 = 12 predictors. It also has 2 categorical fields, one has 6 classes and is dummy-encoded without missing values, another has 3 classes and is contrast-encoded with missing values, which will generate 6 - 1 + 3 + 1 = 9 predictors. It's also got a text field which has 15 terms and missing values, and this will generate 15 + 1 = 16 predictors. In addition, it has 1 items field which has 8 items and no missing values and this will become 8 predictors. Bias term is enabled. Altogether, this linear regression will have 46 predictors from its 14 input fields.

# 3.2.4 III-Conditioned Problems

A linear regression is ill-conditioned when there is insufficient data to estimate the value of the coefficients. Typically, this is when the number of rows is fewer than the number of predictors. In this case, the coefficients which are unable to be estimated will be set to 0, and in the JSON response of the model, the stats output will not contain *standard_errors*, *z_values*, *confidence_intervals*, and *p_values*. A warning will also be added to the model status.

Predictions with an ill-conditioned linear regression will have confidence and prediction intervals equal to 0.

# 3.3 Creating Linear Regressions with 1-Click

To create a linear regression in BigML you have two options: either the 1-click option which uses the default values for all available configuration options, or you can tune the parameters in advanced by using the configuration options explained in Section 3.4. This section explains how to create a linear regression with 1-click.

From the dataset view, select the 1-CLICK LINEAR REGRESSION option in the **1-click action menu** menu. (See Figure 3.5)

Sources Datasets S	upervised - Unsupervise	ed 👻 Predic	ctions 🕶 Tasks	_	WhizzML 👻
a 17. 87		abalo	one dataset	ılề₊ 🕸₊ 🦽,	(≡)⁵- (ĵ)
*				1-CLICK UNSUPERVISED	×
Name	1, Туре	t̂↓ Count		ANOMALY	
Sex	ABC	4,	🚀 LINEAR REGRESSION 🛛 NEW		
Length	123		LOGISTIC REGRESSION DEEPNET	話 で TOPIC MODEL ジャート	
Diameter	123	4	TIME SERIES		
Height	123	4,	RANDOM SPLIT	LINEAR SPLIT	
Whole_weight	123	4,	DOWNLOAD CSV	DOWNLOAD TABLEAU	_
Shucked_weight	123		MOVE TO		
Viscera_weight	123	4,17	77 0 0	• <b>                                    </b>	

Figure 3.5: Create 1-click linear regression from dataset 1-click action menu

Alternatively, you can use the 1-CLICK LINEAR REGRESSION option in the **pop up** menu from the dataset list view. (See Figure 3.6.)

Sourc	ces Datasets Supervised	NEW ▼ Unsupervised ▼	Predicti	ons 🔻 Ta	asks								Whiz	zML	•
				atasets									Ô		0
9	Name	t,	<b>**</b> 1,	<b>i</b> 1,	ŝ	ŕa	*	Σ	×	~	¥.	Ъ.,	$\nabla$	00 6-8	×
ılî	US car accidents [ordered] 4999 instances, 20 fields (8 categorica	l, 11 nu 🖉 💌	4m	771 K											
50	1-CLICK SUPERVISED	1-CLICK UNSUPERVIS	ED	726 K											
ılı	ENSEMBLE	ANOMALY		44 K											
ılî				26 K											
ılı		PCA		26 K											
ılî	TIME SERIES			5 K						1					
50	1-CLICK TRAINING	TEST SPLIT		3 K						1					
ιlî	RANDOM SPLIT	LINEAR SPLIT		6 K											
54	VIEW DETAILS			3 K											
50	MOVE TO		>	190 K											

Figure 3.6: Create 1-click linear regression from dataset popup menu

Either option builds a linear regression using the default values for all available configuration options. (See Section 3.4.)

Note: for some datasets, the 1-click option may be disabled. This can be due to the fact that your dataset does not contain any numeric field or the field taken as the default objective field field is not numeric. If you do not specify any objective field BigML takes the last numeric field in your dataset as the objective field by default. To change the objective field, either configure your linear regression or select the objective field from your dataset (both options are explained in Subsection 3.4.1)

# 3.4 Linear Regression Configuration Options

While the 1-click creation menu option (see Section 3.3) provides a convenient and easy way to create a BigML linear regression, you can also have more control over the linear regression creation and configure a number of parameters that affect the way BigML creates linear regressions. Click the CONFIGURE LINEAR REGRESSION menu option in the **configuration menu** of your dataset view. (See Figure 3.7.)

Sources Datasets	Supervised - Unsupervised ·	<ul> <li>Predictions - Tasks</li> </ul>		WhizzML 🔫
a \$7. \$7	•• [	abalone dataset	ıl°∗ °¢.	🦽 (≡)⁵ (≣)
Name	t̂ _↓ Type t̂ _↓		CONFIGURE UNSUPERVISED	×
Sex	ABC		ASSOCIATION	
Length	123	DEEPNET	PCA	lha
Diameter	123	TIME SERIES		lh
Height	123	4,177 0	°	Itte

Figure 3.7: Configure linear regression

# 3.4.1 Objective Field

The objective field, or "target field", is the field you want to predict. Linear regressions only support **numeric** fields as the objective field.

BigML takes the **last numeric field** in your dataset as the objective field by default. If you want to change the objective field, you have two options: you can select another field from the configuration panel to build the linear regression, or you can change it permanently from your dataset view.

• Select the **Objective field** from the linear regression **configuration panel**. This option will only affect the linear regression you are building that time. (See Figure 3.8.)

Sources Datasets So	upervised 🗸 Unsupe	ervised - Predictio	ns 🕶 Tasks			WhizzML 🔻
≙ ŢŢ, ∰, ••••		abalone	dataset		ılî• <b>%</b> . ¢	≶- (≡) ^{\$} - (i
INEAR REGRESSION CONFIGU	JRATION					POF
Objective field:		Automatic optimiza	ition			0
Class_number_of_rings	123					
Length	123					
Diameter	123					
Height	123					
Whole_weight Shucked_weight	123					
Viscera_weight	123					$\overline{\mathbf{O}}$
Shell_weight	123					U
Class_number_of_rings	123					
abalone dataset				Reset	🛛 🏄 Create lir	near regression
<b></b>					Q	ж
Jame	†ј Туре	t̂↓ Count t̂↓	Missing †	rrors t _i Histo	gram	
Sex	ABC	4,177	0	0		
ength	123	4,177	0	0		hu
Diameter	123	4,177	0	0		la.,

Figure 3.8: Configure the objective field to create the linear regression

• Change the **default objective field** for the dataset. This option will save your objective field preference for any model you build. Click on the edition icon next to the field name when you mouse over it, a pop up window will be displayed. Then click on the Objective field icon and Save it. (See Figure 3.9.)

Sources Datasets Supervis	sed	redictions 👻 Ta	isks		WhizzML 👻
	Update field 000001 detai	ls			ılề₊ 🐎 đ, (≡)⁵, 🛈
					Q *
Name	Name Length		t. Errors	ţ	Histogram
Sex	Label		0	0	
Length	Description		0	0	ututili
Diameter	Description		0	0	
Height			0	0	
Whole_weight	his field is the objective field		0	0	
Shucked_weight	X Canc	el 🖌 Save	0	0	
Viscera_weight	123	4,177	0	0	
Shell_weight	123	4,177	0	0	

Figure 3.9: Change the default objective field

# 3.4.2 Automatic Optimization

You can turn on the **Automatic optimization** option so BigML will automatically tune the parameters of your linear regression (see Figure 3.10).

Sources Datasets Supervis	ed • Unsupervised • Predictions •	Tasks	WhizzML 🕶
a 17. 80	abalone data	set	ılề₊ 🍫 ⊘₅ (≡)⁵₊ ③
LINEAR REGRESSION CONFIGURATIO	N		201
Objective field:     Class_number_of_rings     123       Default numeric value:     Select a default value     *	Automatic optimization	Max. training time: 00:30:00 O	Linear regression candidates:
Advanced configuration			$\odot$
Linear regression name:			
abalone dataset		Reset	差 Create linear regression

Figure 3.10: Automatic optimization

The main focus of optimization in linear regression is the bias term, also known as the intercept term. Hand-tuning it is a time consuming process and BigML offers first-class support for automatic linear regression parameter optimization.

Behind the scenes, BigML uses the same technology for linear regression parameter optimization as the one used for OptiML. If you want to know more about the technical details, please read the Chapter 2 of the document OptiML with the BigML Dashboard [10].

When you turn on the **Automatic optimization** option, all the linear regression parameters will be disabled (because they will be automatically optimized), except the **Default numeric value** and the **Weights** parameters which you can manually configure (see Figure 3.11).

Sources Datasets Supervised -	Unsupervised - Predictions -	Tasks	WhizzML 🗸
≙ <b>₽</b> , <mark>0</mark> ,	abalone datas	set	ılề₂ 🎭 🧆 (≡)⁵₂ 🛈
LINEAR REGRESSION CONFIGURATION			PDF
Objective field: Class_number_of_rings 123 *	Automatic optimization	Max. training time: 00:30:00 O	Linear regression candidates:
Default numeric value: Select a default value			
Advanced configuration			$\odot$
Linear regression name: abalone dataset		Reset	Create linear regression

Figure 3.11: Configure the default numeric value

Since the optimization process can take some time, BigML offers two configurable parameters to limit the time to create the optimized linear regression: a training duration (see Subsection 3.4.2.1) and the linear regression candidates (see Subsection 3.4.2.2).

#### 3.4.2.1 Training duration

The scale parameter to regulate the linear regression runtime. It's set as an integer from 1 to 10. It indicates the user preference for the amount of time they wish the optimization to take. The higher the number, the more time that users are willing to wait for possibly better linear regression performance. The lower the number, the faster that users wish the linear regression training to finish. The default value is set to 5.

The training duration is set in a scale. The actual training time depends on the dataset size, among other factors.

Sources Datasets Supervised -	<ul> <li>Unsupervised</li></ul>	WhizzML 🗸
a 17. 10	abalone dataset	ılề₊ 🍫 🍊₊ (≡)⁵₊ 🛈
LINEAR REGRESSION CONFIGURATION		POT
Objective field: Class_number_of_ri 123 v	Automatic optimization Training duration:	Linear regression candidates:
Default numeric value:       Select a default value		
Advanced configuration		$\odot$
Linear regression name: abalone dataset	Rese	t Create linear regression

Figure 3.12: Training duration

#### 3.4.2.2 Linear regression candidates

The maximum number of different linear regressions (i.e., linear regressions using a unique configuration) to be trained and evaluated during the optimization process. The default number is 128 candidates which is usually enough to find the best linear regression, but you can set it from 4 up to 200. Only the top-performing linear regression will be returned. If the training duration is very low (see <u>Subsection 3.4.2.1</u>) given the dataset size, it is possible that not all the linear regression candidates will be tried out.

Sources Datasets Supervised	d     Unsupervised     Prediction	s 🕶 Tasks	WhizzML 🔻
a 17. 80	abalone	dataset	ılề₂ 🎭 🍊₂ (≡)⁵₂ 🛈
LINEAR REGRESSION CONFIGURATION			PDF
Objective field: Class_number_of_ri 123 v	← Automatic optimization	Training duration:	Linear regression candidates:
Default numeric value:       Select a default value			
Advanced configuration			$\overline{\mathbf{O}}$
inear regression name:			
abalone dataset		Reset	Create linear regression

Figure 3.13: Linear regression candidates

### 3.4.3 Default Numeric Value

Linear regressions can include missing values as valid values for any type of fields as explained in Subsection 3.2.2. However, there can be situations for which you don't want to include them in your model. For those cases, the **Default numeric value** parameter is an easy way to replace missing numeric values by another valid value. You can select to replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 3.14.)

Sources Datasets	Supervised	vew) ▼ Unsuper	vised 🔻 Pi	redictions 👻 🏾	lasks				WhizzML 🔻
8 ¥. 8.	•••		<b>   </b> at	oalone dataset	t			<b>b</b> . (5-	(≡)⁵, 🕕
LINEAR REGRESSION CO	NFIGURATION								PDS
Objective field: Class_number_of_rings	123	•	Automatic o	ptimization					0
Default numeric value: Select a default value									
Mean Median	ation								◙
Minimum Zero						Reset		Create linea	r regression
<b>•</b>							Q		×
Name	η.	Туре	t̂↓ Count	tj Missing	tų Errors	†j His	stogram		
Sex		ABC	ġ.	4,177	0	0			
Length		123		4,177	0	0		utillu	
Diameter		123		4,177	0	0		dillih.	

Figure 3.14: Select a default numeric value to replace missing numeric values

Note: if your dataset does not contain missing values for your numeric fields, this parameter will not have impact on your linear regression.

#### 3.4.4 Weights

It is not unusual for a dataset to have unbalanced fields, which means there are many instances in certain ranges, and few in others. For example, in datasets used to model company financials, there are many more companies with employees numbered from 50-500, while there are only a few with more than 100,000 employees. So as the company size increases, there are fewer cases to fit. In that case, you may want to assign more **weights** to the scarce instances so they are equivalent to the abundant ones.

BigML provides an option to assign specific weights to your instances (see Figure 3.15).

Sources Datasets Supervised	NEW Unsupervised  Predictions Tasks	WhizzML 🔻
≙ Ţ, <mark>6</mark> 0,	abalone dataset	ılề₊ 🍫 🦽₊ (≡)⁵₊ 🛈
LINEAR REGRESSION CONFIGURATION		2 PD1
Objective field:           Class_number_of_rings         123	Automatic optimization	Q
Default numeric value: Select a default value		
State Advanced configuration		۲
Weights:		۲
Select a weight option		WEIGHTS No weights
Bias:		◙
Field codings:		$\odot$
Sampling:		4,177 instances 💿

Figure 3.15: Weight options for linear regression

#### 3.4.4.1 Weight Field

The **Weight field** option allows you to assign individual weights to each instance by choosing a special weight field. The selected field should be integer, with a minimum value of 1, and it must not contain any negative or missing values. However, any non-negative weight field will be accepted. If the minimum value is different from 1, each value in the weight field will be divided by the minimum value and rounded to the nearest integer.

If an instance has a weight of 3 it will be replicated three times in the dataset to train the model.

The weight field will be excluded from the input fields when building the linear regression. You can select an existing field in your dataset or you may create a new one in order to assign customized weights.

#### 3.4.5 Bias

You can include or exclude the **Bias** from the model, a.k.a. the intercept term of the linear regression formula. (See formula in Section 3.2.) For most cases, including the bias results in a better model. By default it is included. (See Figure 3.16.)

Sources Datasets Supervised -	Unsupervised - Predictions - Tasks	WhizzML 🗸
a ₽, <mark>6</mark> ,	abalone dataset	ılề₊ 🍫 🦽₋ (≡)⁵₊ 🛈
LINEAR REGRESSION CONFIGURATION		101
Objective field: Class_number_of_rings 123 v	Automatic optimization	0
Default numeric value: Select a default value		
State Advanced configuration		۲
Weights:		⊚
Bias:		۲
Bias:		BIAS
<b>b</b> ₀		Yes
Field codings:		©
Sampling:		4,177 instances 💿
Dataset advanced sampling:		Default settings 🕞
Advanced ordering:		Default settings 💿

Figure 3.16: Bias parameter

### 3.4.6 Field Codings

**Categorical fields** must be converted to numeric values in order to train a linear regression model. By default, they are **Dummy** encoded, with the default dummy class as the first class in lexicographic order. BigML also allows you to configure two other types of coding for each one of your categorical fields: **Contrast coding**, and **Other coding**. See the following subsections for a detail explanation of each option. (Learn more about input fields transformations in Subsection 3.2.1.)

#### 3.4.6.1 Dummy Coding

The main goal of using dummy coding¹ is to compare a class selected as the reference or **control class** with the rest of classes. The control class is assigned a value of 0 for each variable. The control class is called **dummy class** in BigML and it is usually a class with a representative number of instances compared to the other classes in the dataset. See an example of dummy coding schema for three different classes, with the "Class 1" being the dummy class, in Table 3.2:

Classes	C0	C1	C2
Class 1	0	0	0
Class 2	1	0	0
Class 3	0	1	0
MISSING	0	0	1

Table 3.2: Dummy coding example for 3 classes

¹https://en.wikipedia.org/wiki/Categorical_variable#Dummy_coding

#### To set **Dummy coding** for a field:

1. Click on the configuration icon next to the field name. (See Figure 3.17.)

Sources Datasets S	upervised 🗸 Unsup	pervised  Predictions	Tasks		W	′hizzML <del>▼</del>
		abalone datas	et	- di	¢. 🎭. ⊘. (	(≡)∲, 🚺
LINEAR REGRESSION CONFIGU	JRATION					PDF
Objective field: Class_number_of_rings	123 🔻	Automatic optimization				0
Default numeric value: Select a default value						
Advanced configuration						▣
Linear regression name:			[	Reset	Create linear re	aression
Linear regression name: abalone dataset				Reset	Zreate linear re	
Linear regression name: abalone dataset	1 Type	Ceunt 1 Missin	a t Erro		٩	gression
Linear regression name: abalone dataset	ft Type Configu	re field coding 4,177	g t _t Erro 0		٩	
Linear regression name: abalone dataset	Configu	ire field coding	*	rs 1 Histogram	٩	
Linear regression name: abalone dataset	Configu	4,177	0	rs 1 Histograr 0	٩	
Linear regression name: abalone dataset Name Sex Length	Configu ABC	4,177	0	rs t Histogram 0 0	٩	

Figure 3.17: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field does not have a previous configuration for field codings, it will be disabled. **Enable** field coding configuration by clicking on the green switcher shown in Figure 3.18.

	odels <del>×</del> Clusters Anomalies As	sociations Predictions Tasks	Scripts 🔻
a S' 17. •	onfigure field coding for purpose field	^	· . (=)- ()
Objective field:	DISABLED ENABLED		ps: <b>O</b>
class			0.0001
	Dummy coding:		
Configure	Class	Select	$\overline{\mathbf{r}}$
Logistic regression name:	business	۲	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	0	×
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athas	^	
	Configured coding: NONE	Cancel Save	
credit_history			

Figure 3.18: Enable field coding configuration

3. Select the class you want to set as the dummy class. (See Figure 3.19.)

	Iodels • Clusters Anomalies Association Configure field coding for <b>purpose</b> field	is Predictions Tasks	
bjective field:	DISABLED CABLED		os: 🕜
class	Dummy coding:		
State Configure	Class	Select	
ogistic regression name:	business	•	
oan risk data dataset's	domestic appliance	Select this class as dummy	
<u>8</u> -	education	Ŏ	2
me	furniture/equipment	0	
ecking_status	new car	0	
ration		^	
	Configured coding: NONE	Cancel Save	
edit_history			
	ABC 1,000 0		

Figure 3.19: Select the dummy class

4. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **DUMMY**". (See Figure 3.20.)

Sources Datasets Mo	odels - Clusters Anomalies A	ssociations Predictions Tasks	Scripts -
	onfigure field coding for <b>purpose</b> field		× ()/~
Objective field:	DISABLED CON ENABLED		.ps:
class	Dumr	ny coding selected	0.0001
	Dummy coding:		
Configure	Class	Select	$\odot$
Logistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
×	education	۲	*
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	^	
credit_history	Configured coding: DUMMY	Cancel Save	

Figure 3.20: Field codings: dummy

Note: you cannot select several field codings for the same field simultaneously.

5. Close the modal window by clicking outside or by clicking Cancel .

Sources Datasets	Nodels – Clusters Anomalies A	ssociations Predictions Tasks	Scripts -
6 6. 6.	Configure field coding for purpose field		×
Objective field:	DISABLED ENABLED		ps:
class	Dumn	ny coding selected	0.0001
	Dummy coding:		
්සූ Configure	Class	Select	$\odot$
Logistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	۲	×
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	^	
credit_history	Configured coding: DUMMY	Cancel	

Figure 3.21: Close modal window

Note: if the Cancel button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

6. After configuring the field codings for a field, the configuration icon will become green. (See Figure 3.22.)

Sources Datasets Su	pervised - Unsupervised -	Predictions -	Tasks	_	WhizzML 🔻
®_,		abalone datase	t	ılê- 🗞 🤇	∮. (≡) <b>⁵.</b> (i)
LINEAR REGRESSION CONFIGU	RATION				PDF
Objective field: Class_number_of_rings	Autom	natic optimization			Ø
Default numeric value:       Select a default value					
Advanced configuration					$\odot$
Linear regression name:					
abalone dataset			Rese	et 🛛 🥢 Create lin	near regression
<b></b>				Q	×
Name	Field coding configure Dummy	d as _{ht} t Missing	t Errors		×
		d as Missing 4,177	t Errors t	Histogram	×
Name	Dummy		*	Histogram	×
Name	Dummy	4,177	0 0	Histogram	× III
Name Sex Length	C Dummy	4,177	0 0	Histogram	

Figure 3.22: Field codings configured

7. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 3.23.)

	/lodels ▼ Clusters Anomalies As		Scripts -
a s' 8	Configure field coding for purpose field	×	· (=)~ ()
Objective field:	DISABLED CON ENABLED		.ps: 😧
class	Dumm	y coding selected	0.0001
	Dummy coding:		
Configure	Class	Select	$\overline{\mathbf{a}}$
Logistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	٥	*
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	<u>^</u>	
	Configured coding: DUMMY	Cancel	
credit_history			

Figure 3.23: Disable field coding configuration

After creating your linear regression, your dummy class will be identified with the dummy icon in the **coefficients table** view (see Subsection 3.5.2). (See Figure 3.24.)

Sources Datasets Sup	pervised - Unsupervised -	Predictions - Task	KS	WhizzML 🗸
	Predicting	abalone age from mea	asurements	ૐ- (≡)⁵- 0
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIE		FIELD CODINGS YES
	S: Filter bias or predictors		SIGNIFICANCE LEVEL: 0.1	CSV
Bias and predictors		1 Туре	Coefficients	ţ
Bias		123	3.90015	s o
Sex = M		ABC	0.05522	σ
Sex = I	Du	AB C	-0.82452	σ
Sex = F		АВС	0	
Length		123	-0.47157	σ
Diameter		123	11.07550	σ
Height		123	10.73700	σ
Whole_weight		123	8.98378	σ
Shucked_weight		123	-19.78820	σ
Viscera_weight		123	-10.57760	σ

Figure 3.24: Dummy class in table view

#### 3.4.6.2 Contrast Coding

Contrast coding² allows you to set different values for different classes. Instead of the 0-1 values of **Dummy** coding, you will be able to set any integer or float value for each of the classes, plus an additional one for missing values. The sum of all values must equal 0. The values of the classes need to be set based on certain hypothesis, e.g., higher values for a class assume this class has more influence on the objective field than the others. A positive value indicates a positive relationship between the class and the objective field while a negative value indicates a negative relationship. A coefficient of 0 will exclude the class from the model. In the Table 3.3 you can see an example of contrast coding schema for three different classes.

Classes	C0
Class 1	0.5
Class 2	-0.25
Class 3	-0.25
MISSING	0

Table 3.3: Contrast coding example for 3 classes

To set Contrast coding for a field, follow these steps:

1. Click on the configuration icon next to the field name. (See Figure 3.25.)

²https://en.wikipedia.org/wiki/Categorical_variable#Contrast_coding

Sources Datasets Su	pervised	Unsupervised	<ul> <li>Predictions </li> </ul>	Tasks		WhizzML 👻
A Tr 60,		11	Loan risk data da	ataset	เมือง	<b>‰</b> _ (≞) [¢] , (i)
LINEAR REGRESSION CONFIGUR	RATION					PDF
Objective field: credit_amount	123	C Auto	matic optimization			0
Default numeric value: Select a default value						
Advanced configuration						$\overline{\mathbf{O}}$
Linear regression name:						+
Loan risk data dataset					Reset	Create linear regression
<b>•</b> •					٩	×
Name	ţ,	Туре 🖞	Count 🖞 Missir	ng t _i Error	s 🖞 Histogram	
checking_status		ABC 🍇	1,000	0	0	
duration		Configure field co	1,000 ding	0	° minulu.	Illinduraaa
credit_history	≤	ABC 🍖	1,000	0	0	
purpose		ABC 🔅	1,000	0	0	
credit_amount	Ф	123	1,000	0	• <b>•</b>	
savings_status		ABC 🍇	1,000	0	0	

Figure 3.25: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field does not have a previous configuration for field codings, it will be disabled. **Enable** field coding configuration by clicking on the green switcher shown in Figure 3.26

Sources Datasets			
	Configure field coding for credit_history field	×	. G. (≡) ⁵ . ③
LINEAR REGRESSION COM	DISABLED ENABLED		
Objective field:			0
credit_amount	Dummy coding:		
	Class	Select	
Default numeric value:	all paid	۲	
Select a default value	critical/other existing credit	0	
्रिक Advanced configura	delayed previously	0	
Linear regression name:	existing paid	0	reate linear regression
V V	no credits/all paid	0	×
Name	Configured coding: NONE	Cancel Save	
checking_status			
		• • minuta	ntutanaaa

Figure 3.26: Enable field coding configuration

3. Select the Contrast coding option. (See Figure 3.27.)

Sources Datasets			
	Configure field coding for credit_history field	Contrast	·_ ∽ (=) [*] - ③
LINEAR REGRESSION COP	DISABLED CONSTRUCTION ENABLED		201
Objective field:			0
credit_amount	Contrast coding:		
	Class	Value	
Default numeric value:	all paid	0	
Select a default value		0	
Que de la construcción de la con	critical/other existing credit	0	$\overline{\mathbf{O}}$
Advanced configure	delayed previously	0	
Linear regression name:		-	
Loan risk data dataset	existing paid	0	reate linear regression
<b>X</b>	no credits/all paid	0	*
Name	Configured coding: NONE	Cancel	
checking_status			
		· · · · · · · · · · · · · · · · · · ·	

Figure 3.27: Field codings: contrast coding

4. Set the values you want for your classes based on your hypothesis. All classes values must sum 0. (See Figure 3.28.) By using the BigML API, multiple contrast codings can be given for a field as long as all the codings are orthogonal to ensure there are no co-dependent coefficients. Check the corresponding

Sources Datasets			
A 17. 8.	Configure field coding for credit_history	y field ×	·_ (5- (=) ⁵ - ()
LINEAR REGRESSION CON	DISABLED CONTRACTOR		e e e
Objective field:			0
credit_amount	Contrast coding:		
	Class	Value	
Default numeric value:	all paid	1	
Advanced configur	critical/other existing credit	-1	$\overline{\mathbf{a}}$
	delayed previously	-0.75	
Linear regression name:	existing paid	0.5	reate linear regression
	no credits/all paid	0.25	×
Name	Configured coding: NONE	Cancel Save	
checking_status			J
		1.000 0 0 Internation	

#### documentation³.

Figure 3.28: Set the contrast coding values for each class

#### Note: you cannot select several field codings for the same field simultaneously.

5. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **CONTRAST**". (See Figure 3.29.)

 $^{^{3} \}verb+https://bigml.com/api/linearregressions \#lr_coding_categorical_fields$ 

Sources Datasets			
	Configure field coding for credit_history fi	eld ×	·_ Ø- (≡) [¢] - ③
LINEAR REGRESSION COM	DISABLED COM ENABLED		
Objective field:	Contrast o	oding selected	0
credit_amount	Contrast coding:		
Default numeric value:	Class	Value	
Select a default value	all paid	1	
De la compañía de la	critical/other existing credit	-1	
୍ଷି _{ରୁ} : Advanced configur	delayed previously	-0.75	
Linear regression name:	existing paid	0.5	reate linear regression
	no credits/all paid	0.25	н
Name	Configured coding: CONTRAST	Cancel Save	
checking_status			
			h

Figure 3.29: Contrast coding saved

6. Close the modal window by clicking outside or by clicking Cancel .

Sources Datasets			
	Configure field coding for credit_history fi	eid ×	· ↓ ⊘ -   (≡) ⁴ -   ③
LINEAR REGRESSION COM	DISABLED ENABLED		en e
Objective field:	Contrast o	coding selected	0
credit_amount	Contrast coding:		
Default numeric value:	Class	Value	
Select a default value	all paid	1	
0	critical/other existing credit	-1	
Advanced configure	delayed previously	-0.75	
Linear regression name:	existing paid	0.5	reate linear regression
	no credits/all paid	0.25	8
Name	Configured coding: CONTRAST	Cancel Save	
checking_status			

Figure 3.30: Close modal window

Note: if the Cancel button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

7. After configuring the field codings for a field, the configuration icon will become green. (See Figure 3.31.)

Sources Datasets Su	pervised 🗸	Unsupervised •	Predictions -	Tasks		WhizzML 🔻
		11	Loan risk data da	ataset		ılề- ‰_ ∽ (≡)% (i
LINEAR REGRESSION CONFIGU	RATION					PDF
Objective field: credit_amount	123	Auto	natic optimization			Ø
Default numeric value: Select a default value						
Advanced configuration						$\odot$
Linear regression name: Loan risk data dataset					Reset	Create linear regression
						Q
Name	4	Type †	Count † Missin	ng î _l Err	rors t _i	Histogram
checking_status		ABC 🇞	1,000	0	0	
duration		Field coding configu Contrast	red as 1000	0	0	
credit_history	<b></b>	АВС 🌼	1,000	0	0	
purpose		АВС 🏠	1,000	0	0	a di la casa di la cas
credit_amount	<b>\</b>	123	1,000	0	0	
savings_status		ABC 🇞	1,000	0	0	

Figure 3.31: Field codings configured

8. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 3.32.)

	lodels  Clusters Anomalies Asso Configure field coding for purpose field	ociations Predictions Tasks	
Objective field:	DISABLED CABLED		r_
class	Contrast of	coding selected	0.0001
	Contrast coding:	a di la casa	
Configure	Class	Value	$\overline{\mathbf{v}}$
Logistic regression name:	business	-0.25	
Loan risk data dataset's	domestic appliance	1	
	education	0.5	te logistic regression
Name	furniture/equipment	-0.25	
checking_status	new car	-1	
duration			
credit_history	Configured coding: CONTRAST	Cancel	

Figure 3.32: Disable field coding configuration

After creating your linear regression, you will be able to see your **Contrast coding** values in the **coefficients table** view (see Subsection 3.5.2) by clicking on the icon. (See Figure 3.33.)

Sources Datasets Supervised -	) Unsupervised - Predictions - Task	(S	WhizzML 🗸
A ₹. 80	Loan risk data dataset	t 💠	
INSTANCES DEFAULT NUT 1,000 NOT DEFIN			FIELD CODINGS YES
VARIABLES: Filter bias	or predictors	SIGNIFICANCE LEVEL: 0.1	CSV
Bias and predictors	↑, Туре	t̂↓ Coefficients	ţ
Bias	123	4434.90000	s o
checking_status = no checking	ABC	-166.90200	σ
checking_status = <0	ABC	-368.29900	σ
checking_status = 0<=X<200	(II) ABC	0	
checking_status = >=200	ABC	-801.30900	σ
duration	View Classes Values 23 (Contrast Coding)	125.82000	σ
credit_history	OR ABC	-23.10250	σ
purpose = radio/tv	ABC	-172.02600	σ
purpose = new car	ABC	91.26980	σ
purpose = furniture/equipment	ABC	67.70620	σ
purpose = used car	ABC	809.84000	σ

Figure 3.33: Contrast icon in table view

A modal window will be displayed with your codings values and you can download them in CSV or JSON

Sources Datasets	Supervised - Predict			
8 T. 8.	Credit_history		CSV ISON ×	ૐ- (≡)*- ③
INSTANCES	Class	Value		
1,000	existing paid	1		
	critical/other existing credit		1	CSV
Bias and predictors Bias	delayed previously		0.75	
checking_status = no checkin	all paid	C	0.5	
checking_status = <0	no credits/all paid	C	0.25	Δσ
checking_status = 0<=X<200		ABC	0	
				<b>△</b> σ

format by clicking on the corresponding icons. (See Figure 3.34.)

Figure 3.34: Contrast modal window in table view

#### 3.4.6.3 Other Coding

Other coding⁴ allows you to set different values for different classes. It works the same way as contrast coding (see Subsection 3.4.6.2), but in this case the values do not need to sum 0. In the Table 3.4 you can see an example of other coding schema for three different classes.

Classes	C0
Class 1	2
Class 2	-0.4
Class 3	3
MISSING	1

Table 3.4: Other coding

To set **Other coding** for a field, follow these steps:

1. Click on the configuration icon next to the field name. (See Figure 3.35.)

 $^{{}^{4} \}verb+https://en.wikipedia.org/wiki/Categorical_variable#Contrast_coding$ 

Sources Datasets Sup	bervised 🗸 Ur	nsupervised 👻	Predictions -	Tasks			WhizzML 👻
		111 :	abalone datase	t		ıl 🐂 🗞 🛛	(≝) <b>*</b> (≣)
LINEAR REGRESSION CONFIGUR	RATION						PDF
Objective field: Class_number_of_rings	123	Automatic	optimization				Ø
Default numeric value: Select a default value							
Advanced configuration							$\odot$
Linear regression name:					<b>D</b>	Create	linear regression
					Reset	Greate	iniear reuression
					Heset		
Name	t Type	nfigure field coding	nt 1 Missing	t _i Error		Q listogram	
		nfigure field coding	nt 1 Missing 4,177	t _e Errors		٩	
Name	Co	nfigure field coding	· ·	Ŧ	s 1 H 0	٩	
Name Sex		nfigure field coding	4,177	0	s 👘 H 0 0	Q listogram	
Name Sex Length	Co AB 123	nfigure field coding	4,177 4,177	0	s 👘 H 0 0	Q listogram	

Figure 3.35: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field does not have a previous configuration for field codings, it will be disabled. **Enable** field coding configuration by clicking on the green switcher shown in Figure 3.36

a (2° 10° - •	Configure field coding for purpose field	×	· (=)~ ()
Dbjective field:	DISABLED CON ENABLED		ps: 🕖
class			0.0001
	Dummy coding:		
්සූ Configure	Class	Select	
ogistic regression name:	business	۲	
Loan risk data dataset's	domestic appliance	0	te logistic regression
✓	education	0	
lame	furniture/equipment	0	
hecking_status	new car	0	
uration		^	
	Configured coding: NONE	Cancel Save	
redit_history			

Figure 3.36: Enable field coding configuration

n 6º 67. •	DISABLED CONTRACT	Other	· ↓ C5 - (=)~
ojective field:			ips: 🔞
lass	Dummy coding:		0.0001
Configure	Class	Select	$\overline{\mathbf{a}}$
	business	•	
gistic regression name: .oan risk data dataset's	domestic appliance	0	te logistic regression
	education	0	te logistic regression
⊻ ▼	furniture/equipment	0	:
	new car	0	
ecking_status	athar	<u>^</u>	_
ration	Configured coding: NONE	Cancel Save	
edit_history			

#### 3. Select the Other coding option. (See Figure 3.37.)



4. Set the values you want for your classes based on your hypothesis. You can set any float or integer value. (See Figure 3.38.) By using the BigML API, multiple other codings can be given for a field. Check the corresponding

	odels  Clusters Anomalies Assoc Configure field coding for purpose field	iations Predictions Tasks	(NEW) Scripts +
a S 19			:• (=) <mark>/•</mark> (=)
Objective field:	DISABLED CONTRACTOR CONTRACTOR		ps: 0
class	Other coding:	and the second second	
्र्कु Configure	Class	Value	$\overline{\mathbf{r}}$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	×
Name	furniture/equipment	0	
checking_status	new car	-2	
duration	athas	^	
credit_history	Configured coding: NONE	Cancel Save	
	<b>ABC</b> 1,000		

documentation⁵.

Figure 3.38: Set the other coding values for each class

 $^{5} \tt https://bigml.com/api/linearregressions \# lr_coding_categorical_fields$ 

Note: you cannot select several field codings for the same field simultaneously.

5. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **OTHER**". (See Figure 3.39.)

	odels  Clusters Anomalies Asso	ciations Predictions Tasks	
	DISABLED C ENABLED		:_
Objective field:	Other cod	ding selected	0.0001
	Other coding:		~
Configure	Class	Value	$\overline{\mathbf{a}}$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	
Name	furniture/equipment	0	×
checking_status	new car	-2	
	athar	0	
duration	Configured coding: OTHER	Cancel Save	
credit_history	Comigured Couling, OTHER	Gancer	
purpose	ABC 1,000	0 0	

Figure 3.39: Other coding saved

bjective field:			
	DISABLED CON ENABLED		: ▲ (ﷺ) (≡) (=) (₩) ( ips: •
ass	Other c	oding selected	0.0001
	Other coding:		
configure	Class	Value	۲
stic regression name:	business	1	
an risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	
e	furniture/equipment	0	
king_status	new car	-2	
	athar	0	
tion	Configured coding: OTHER	Cancel	
it_history	Connigated boding. Officer	ounder Dare	

6. Close the modal window by clicking outside or by clicking Cancel.

Figure 3.40: Close modal window

Note: if the **Cancel** button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

7. After configuring the field codings for a field, the configuration icon will become green. (See Figure 3.41.)

Sources Datasets Su	upervised - Unsupervised ·	<ul> <li>Predictions -</li> </ul>	Tasks	_	WhizzML 👻
		abalone_csv		เมือง	‱ , (≡) [*] , (∋
LINEAR REGRESSION CONFIGU	JRATION				N PDF
Objective field: Class_number_of_rings	Auto	omatic optimization			0
Default numeric value: Select a default value					
					$\overline{\mathbf{O}}$
Advanced configuration					
Advanced configuration				Reset	Create linear regression
Linear regression name:				Reset	Create linear regression
Linear regression name: abalone_csv	Timo Field coding configured	Count dia Missing	) t Errors	٩	
Linear regression name: abalone_csv	Tino		a † Errors 0	٩	
Linear regression name: abalone_csv	Field coding configured	d as Other	*	Q Histogram	
Linear regression name: abalone_csv	Field coding configured	d as Other 4,177	0	Q t Histogram 0	
Linear regression name: abalone_csv wame Sex Length	Field coding configured	4,177 4,177	0	Q B 1 Histogram 0	

Figure 3.41: Field codings configured

8. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 3.42.)

	Indels  Clusters Anomalies Assoc Configure field coding for purpose field	iations Predictions Tasks	Scripts -
a s [.] 8			·
Objective field:	DISABLED CON ENABLED		ps: 🕜
class	Other cod	ling selected	
	Other coding:		
්තු Configure	Class	Value	$\odot$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	×
Name	furniture/equipment	0	
checking_status	new car	-2	
duration	athau	0	
	Configured coding: OTHER	Cancel Save	
credit_history			

Figure 3.42: Disable field coding configuration

After creating your linear regression, you will be able to see your **Other coding** values in the **coefficients table** view (see <u>Subsection 3.5.2</u>) by clicking on the icon. (See Figure 3.43.)

Sources Datasets Su	pervised - Unsupervised -	Predictions - Tasks		WhizzML 👻
	🏄 at	palone other encoding	<b>\$</b>	) _∽ (≡) [∳] _∼ (i)
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD CLASS_NUMBER_OF_RINGS	BIAS FI	ELD CODINGS YES
	VARIABLES: Filter bias or predic	ctors	SIGNIFICANCE LEVEL: 0.1	· 🛕 🖤
Bias and predictors		it ihe if	Coefficients	t _i
Bias		Classes Values 23	4.15383	σ
Sex		Ф АВС	-0.22413	σ
Length		123	-0.55585	σ
Diameter		123	11.34070	σ
Height		123	10.89940	σ
Whole_weight		123	9.00165	σ

Figure 3.43: Other coding in coefficients table

A modal window will be displayed with your coding values and you can download them in CSV or JSON format by clicking on the corresponding icons. (See Figure 3.44.)

	PLOYMENTS GALLERY *	EXP LABS	TERESA V	WHAT'S NEW DE	VELOPERS Dashboard
	Checking_status		CSV	ा 🎽 🙀	
Sources Datasets	Class		Value		
	no checking		2		
	<0		1	2	(≝) (≕) (≕) (€)
INSTANCES	0<=X<200		0.4		
	>=200		-3		
	[MISSING]		0		CSV CSV
Bias and predictors				1	
Bias		123	0	0	
checking_status	(	бр авс			

Figure 3.44: Other coding modal window

# 3.4.7 Sampling Options

Sometimes you do not need all the data contained in your dataset to build your linear regression. If you have a very large dataset, sampling may be a good way of getting faster results. BigML allows you to sample your dataset before creating the linear regression, so you do not need to create a separate dataset first. You can find a detailed explanation of the sampling parameters available in the following subsections. (See Figure 3.45.)

#### 3.4.7.1 Rate

The **Rate** is the proportion of instances to include in your sample. Set any value between 0% and 100%. Defaults to 100%.

#### 3.4.7.2 Range

Specifies a subset of instances from which to sample, e.g., choose from instance 1 until 200. The **Rate** you set will be computed over the **Range** configured. This option may be useful when you have temporal data, and you want to train your linear regression with historical data, and test it with the most recent one to check if it can predict based on time.

#### 3.4.7.3 Sampling

By default, BigML selects your instances for the sample by using a random number generator, which means two samples from the same dataset will likely be different even when using the same rates and row ranges. If you choose deterministic sampling, the random-number generator will always use the same seed, thus producing repeatable results. This lets you work with identical samples from the same dataset.

#### 3.4.7.4 Replacement

Sampling with replacement allows a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once. By default, BigML generates samples without replacement.

#### 3.4.7.5 Out of bag

This argument will create a sample containing only out-of-bag instances for the currently defined rate, so the final total number of instances for your sample will be one minus the rate configured for your sample (when replacement is false). This can be useful for splitting a dataset into training and testing subsets. It is only selectable when a sample rate is less than 100%.

	Sources Datasets Logis	stic Regressions 🔻	Clusters Anoma	alies Associations	Predictions Ta	isks	Scripts
	≜ ≦ ^q ∰, ••••		Loan ris	k data dataset		<b>%</b> G.	(≡) [/] ~ (
	Objective field:	ABC	V Select a de		lissing numerics: 🕢	Eps: 😧	0.0001
	Configure						e
	Scales:						⊚
	Regularization:						$\odot$
Dete	Sampling:					1,000 instances	۲
Rate	Rate:	100%				SAMPLING RATE	0
Range	Advanced sampling:					Custom settings	۲
	Range: 1,000 instances	1,000	RANGE	SAMPLING	REPLACEMENT YES	OUT OF BAG	0
Sampling-	· · · · · · · · · · · · · · · · · · ·	G				,	
placement							
Out of bag	ı						

Figure 3.45: Sampling parameters for linear regression

# 3.4.8 Advanced Ordering

Ordering options are relevant to ensure that BigML can correctly determine whether it can take an early split of your dataset to accelerate the training process. In particular, early splitting can only be safely used if the training instances have been previosuly shuffled.

If your instances are already shuffled, BigML allows you to choose the **linear** option. This will make the process of building the model much faster, since it will not required to reshuffle the dataset. If you need to shuffle your instances, BigML provides two options to that aim, **deterministic shuffling** and **random shuffling**, which are described below.

Ordering options have no influence on datasets of less than 34GB, since the whole dataset is used to build the model.

By default, BigML uses **deterministic shuffling** to ensure the same (deterministic) sample of the instances is used and the built model is thus repeatable.

#### 3.4.8.1 Deterministic Shuffling

The **deterministic shuffling** option ensures that the row shuffling of a dataset is always the same, so that retraining a BigML model from the same dataset yields the same results.

By default, this option is true.

#### 3.4.8.2 Linear Shuffling

The **linear shuffling** option is useful when you know that your instances are already in random order. Using linear shuffling, the BigML model will be constructed faster.

By default, this option is false.

#### 3.4.8.3 Random Shuffling

The **random shuffling** option will ensure that a different shuffling will be tried each time you train your model.

By default, this option is false.

Image: Image		Sources Datasets Ensembles - Clu	isters Anomalies Associations Predictions Tasks	Scrip	ots
Dabetes     % Configure     Missing splits:		a s [.] 6	Diabetes Diagnosis' da	<b>%</b> G.	()
Missing splits: Missing splits: Node threshold: 192 Weights: Sampling: Advanced sampling: Advanced sampling: Custom settings Ordering: Ordering: Ordering: Deterministic					
No     No       Node threshold:     512       Weights:     Image: Sampling:       Sampling:     766 instances       Advanced sampling:     Custom settings       Advanced ordering:     Custom settings       Ordering:     Ordering:       Ordering:     Image: Sampling:		තීස් Configure			۲
Index initiality     512       Weights:     Image: Custom settings       Sampling:     768 instances       Advanced sampling:     Custom settings       Advanced ordering:     Custom settings       Ordering:     Image: Custom settings       Image: Custom settings     Image: Custom settings       Image: Custom settings     Image: Custom settings					0
Sampling: 768 instances © Advanced sampling: Custom settings © Advanced ordering: Custom settings © Ordering: Ordering: Ordering: Ordering: Ordering: Ordering: Ordering:					0
Advanced sampling: Custom settings (*) Advanced ordering: Custom settings (*) Ordering: Ordering: Orderi		Weights:		C	
Advanced ordering: Custom settings (a)		Sampling:		768 instances	)
Ordering: Ordering: Deterministic		Advanced sampling:		Custom settings	)
Deterministic	terministic	Advanced ordering:		Custom settings	)
	Linear				

Figure 3.46: Ordering argument for linear regression

# 3.4.9 Creating Linear Regressions with Configured Options

After finishing the configuration of your options, you can change the default linear regression name in the editable text box. Then you can click on the Create linear regression button to create the new linear regression, or reset the configuration by clicking on the Reset button.

jective field:	Automatic optimizatio	n			
ass_number_of_rings					
fault numeric value: elect a default value					
Advanced configuration					(
Weights:					⊽
Bias:					$\overline{\mathbf{O}}$
Field codings:					$\overline{\mathbf{O}}$
Sampling:				4,177 instances	$\overline{\mathbf{O}}$
Dataset advanced sampling:				Default settings	⊽
Advanced ordering:				Default settings	♥
ear regression name:			-		
alone dataset			Reset	🥢 Create linear re	egressio
Linear Regression Name		API Preview	Reset	Create Linear Reg	ressior

Figure 3.47: Create linear regression after configuration

# 3.4.10 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to the

Reset button (See (Figure 3.47)). This is to show how to create the linear regression programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 3.48) below:

		101
Objective field:	Common Automatic optimization	0
Default numeric v	API request preview API endpoint: /linearregression	×
Advanced o	<pre>1 - { 2     "objective_field": "000008", 3     "name": "abalone dataset", 4     "bias": true, 5     "optimize": false,</pre>	۲
Weights:	6 "dataset": "dataset/5c93b9e228e1f47a5e000388" 7 ]	$\overline{\mathbf{O}}$
Bias:		$\overline{\mathbf{a}}$
Field coding		$\overline{\mathbf{O}}$
Sampling:	Check the full list of arguments in the API documentation	n 
Advanced ord		
Linear regression na		

Figure 3.48: Linear regression API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for linear regressions, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

# 3.5 Visualizing Linear Regressions

After creating your linear regression, you will be able to analyze your results with BigML unique visualization: a **1D chart** and a **Partial Dependence Plot (PDP)**, to examine the impact of the input fileds in the objective field, and a **coefficients table**, for more advanced users, to interpret the resulting coefficients for each input field. The following subsections explain both visualizations in detail.

Switch among the three views of 1D chart, PDP and table by clicking on icons in the top bar menu of the linear regression view. (See Figure 3.49.)

	Predicting	abalone age from measurements	Ģ	(≡)⁵- (
INSTANCES	DEFAULT NUMERIC	OBJECTIVE FIELD	BIAS	FIELD CODINGS
	NOT DEFINED	CLASS_NUMBER_OF_RINGS	YES	YES

Figure 3.49: Switch chart, PDP and tableviews

#### 3.5.1 Linear Regression Chart

Both views of 1D chart and PDP are composed of three main parts: the CHART or PLOT, the PREDIC-TION legend and the INPUT FIELDS form. (See Figure 3.50). You can find a detailed explanation of each one below.



Figure 3.50: Linear regression chart parts

• Both the 1D Chart and the Partial Dependence Plot (PDP) allow you to view the impact of the input fields on the objective classes predictions, and their relationship. You can select the 1D chart or PDP by clicking on the selection buttons in the top bar menu.

The **1D chart**, allows you to select one input field for the x-axis. The y-axis represents the objective field. Only numeric fields can be selected for the x-axis. (See chart limitations in Subsection 3.8.1). You can extend the upper limit of the x-axis by clicking on the plus icon.

The blue band is formed by the upper bounds and lower bounds of the 95% prediction intervals. This means, for any given point at x-axis, its y value will be within this blue range with 95% probability. You can hide or show the band by clicking on the icon next to the field name of the y-axis.


Figure 3.51: 1D chart

The **PDP** allows you to select two different input fields for both axis and the values of the objective field are represented by differences in a color scale in the heatmap chart. You can select numeric or categorical fields for the axis. You can switch the axis by clicking on the option on top of the chart area.



Figure 3.52: Partial Dependence Plot

In both charts you can inspect the axis values in the gray area boxes next to the selector. You can freeze the view by pressing Shift and release it again by pressing Esc from your keyboard. When the view is frozen, an edit icon will appear you can edit the axis values and obtain a prediction for another preferred value. The resulting predicted probabilities are in the prediction legend to the right.

 The PREDICTION legend allows you to visualize the predicted values represented in the chart along with their corresponding colors. By default, in PDP, colors are shaded according to the prediction range shown in the chart area. This way, smaller differences in predictions are easier to perceive. However, you can choose to see the color shading according to the total range of values for the objective field by clicking on the icon next to the prediction bar Total. This Total option allows you to see the color scale for the total range of predictions. (See Figure 3.53)



Figure 3.53: Prediction legend

Again, freeze this view by pressing Shift, and release it again by pressing Esc from your keyboard.

• Below the chart legend, you can find the INPUT FIELDS form. (See Figure 3.54). You can configure the values for any numeric, categorical, text or items field. By changing their values, you can see the predictions changing in real-time.



Figure 3.54: Configure the values for other input fields

Moreover, the chart includes a reset option for your input fields values, and an export option to download your chart in PNG format explained below:

• After selecting the fields for the axis or configuring the input fields values, you can set them again to the default view by clicking the **reset** icon highlighted in Figure 3.55.



Figure 3.55: Reset the values for the input fields

• You can also **export** your chart in PNG format with or without legends. Freeze the view by pressing Shift from your keyboard and export the chart to get the classes percentages in the legend. Release the view by pressing Esc.



Figure 3.56: Export chart as image with or without legends

Note: there are some limitations in the number of input fields to visualize your linear regression in the chart (explained in Subsection 3.8.1).

## 3.5.2 Coefficient Table

The main goal of the linear regression algorithm is to learn the **coefficients** of the linear function for each of the dependent variables, i.e., for each of the input fields. See Section 3.2 for a detailed explanation of linear regression coefficients interpretation.

BigML allows you to inspect the learned coefficients for each one of the input fields in the coefficient table. The table **columns** represent coefficients and their statistics while the table **rows** represent the input field variables and the **Bias** (a.k.a. intercept term) of the linear regression. In the first row you will always find the Bias coefficients. You can sort the table rows by clicking on any of the columns labels.

Sources Datasets S	upervised - Unsupervised -	Predictions - Tasks		WhizzML 👻
	Predicting a	balone age from measurer	ments 🗘	ૐ- (≡)⁵- 0
4,177	DEFAULT NUMERIC	OBJECTIVE FIELD CLASS_NUMBER_OF_RINGS	BIAS YES	FIELD CODINGS YES
	VARIABLES: Filter bias or predic	ctors	SIGNIFICANCE LEVEL:	0.1 ×
Bias and predictors		1 Туре	t↓ Coefficients	t _i
Bias		123	3.89464	σ
Sex = M		ABC	0.05772	σ
Sex = I		ABC	-0.82488	σ
Sex = F		(II) ABC	0	
Length		123	-0.45833	σ
Diameter		123	11.07510	σ
Height		123	10.76150	σ
Whole_weight		123	8.97544	σ
Shucked_weight		123	-19.78690	σ
Viscera_weight		123	-10.58180	σ
Shell_weight		123	8.74181	σ

Figure 3.57: Table view for linear regression

For numeric fields, there is always one coefficient by field, however categorical, text and items fields have one coefficient by value (category, term or item). This is due to the required transformations, explained in Section 3.2, to convert categorical, text, and items fields to numeric fields (each single value is mapped to a separate variable in the formula). Missing values also get their own coefficients.

- For **numeric** fields, you always get one coefficient per field. If a field contains missing values, you will find an additional coefficient per field for the missing values. (See Subsection 3.2.2.)
- For **categorical** fields, you have one coefficient per class and an additional one for missing values per field. (See Subsection 3.2.3 and Subsection 3.4.6.)
- For text fields, there is one coefficient per term and an additional one for missing values per field.
- · For items fields, you get one coefficient per item and an additional one for missing values per field.

See an example of coefficients for a categorical field in Figure 3.58 where one single field, "Atmospheric condition", yields eleven different variables associated with different coefficients. There are twelve classes in the categorical field, with one set as the dummy class.

Sources Datasets Su	upervised - Unsupervised -	Predictions - Tasks			WhizzML 👻
<b>₽ ₽</b> . <b>8</b>	📈 US ca	r accidents linear regressio	on	Ŷ	
INSTANCES 4,999	DEFAULT NUMERIC	OBJECTIVE FIELD	ASHF	BIAS Yes	FIELD CODINGS YES
	VARIABLES: Fi	Iter bias or predictors			<u>A</u>
Bias and predictors		ţ,	Туре	t. Coefficients	3
Bias			123	0.78471	
Atmospheric Condition Atmosph	= Clear		ABC	0.11762	
Atmospheric Condition Atmosph	= Cloudy		ABC	0.10463	
Atmospheric Condition Atmosph	= Snow		ABC	0.11161	
Atmospheric Condition Atmosph	= Rain		ABC	0.13224	
Atmospheric Condition Atmosph	= Sleet, Hail (Freezing Ra		ABC	0.18139	
Atmospheric Condition Atmosph	= Fog, Smog, Smoke		ABC	0.10502	
Atmospheric Condition Atmosph	= Blowing Snow		ABC	0.02805	
Atmospheric Condition Atmosph	= Severe Crosswinds		ABC	0.20018	
Atmospheric Condition Atmosph	= Not Reported		АВС	0.34903	
Atmospheric Condition Atmosph	= Unknown		АВС	0.13544	
Atmospheric Condition Atmosph	= Other		ABC	0.04016	
Atmospheric Condition Atmosph	= Blowing Sand, Soil, Dirt		ABC	0	

Figure 3.58: Multiple field variables for categorical dummy encoded fields

Coefficients for **missing values** are always found at the end of the table. (See Figure 3.59)

Sources Datasets	Supervised - Unsupervised -	Predictions 👻 Tasks	_	WhizzML 🗸
	••• Predicting	abalone age from measu	urements 🗘	ૐ- (≡)⁵- 🛈
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD CLASS_NUMBER_OF_RIN		FIELD CODINGS YES
	VARIABLES: Filter bias or pred	ictors	SIGNIFICANCE LEVEL:	0.1 🔻
Bias and predictors		1 Туре	Coefficients	ţ
Bias		123	3.90015	σ
Sex = M		ABC	0.05522	σ
Sex = I		ABC	-0.82452	σ
Sex = F		(]) ABC	0	
Length		123	-0.47157	σ
Diameter		123	11.07550	σ
Height		123	10.73700	σ
Whole_weight		123	8.98378	σ
Shucked_weight		123	-19.78820	σ
Viscera_weight		123	-10.57760	s a
Shell_weight		123	8.73112	▲
Sex		missing	1.88682	σ
Diameter		missing	2.90388	σ
Show 25 🗘 variables		1 to 13 of 13 variables		

Figure 3.59: Missing numeric coefficients at the end of linear regression table

Next to each coefficient you will find one icon indicating if it is significant (see Figure 3.60) or non-significant (see Figure 3.61).



Figure 3.60: Significant icon



Figure 3.61: Non-significant icon

The significance of a coefficient is determined by comparing the *p*-value against the significance level selected in the top menu (see Figure 3.62). If the *p*-value is higher than the significance level, the coefficient will be non-significant. If the *p*-value is lower than the significance level, the coefficient will be significant. A good practice is to retrain the linear regression removing the non-significant coefficients. However, in most cases, the model performance should not be affected.

Sources Dataset	s Supervised - Unsupervised -	Predictions - Tasks		WhizzML 👻
a   ¥-  80-	••••• <u>//</u> con	crete_data   Training (80	%)	ૐ- (≡)⁵- ③
INSTANCES 824	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD		FIELD CODINGS
	VARIABLES: Filter bias or predictors		SIGNIFICANCE LEVEL: 0.1	CSV
Bias and predictors		† Туре	Coefficients 0.01	1
Bias		123	-34.87790 0.1	<u>م</u> ه
cement		123	0.12173	⊂
blast_furnace_slag		123	0.10651	σ
fly_ash		123	0.09088	σ
water		123	-0.11988	σ
superplasticizer		123	0.39687	σ

Figure 3.62: Select significance level

Sources Datase	ts Supervised - Unsupervised -	<ul> <li>Predictions - Task</li> </ul>	s	WhizzML 🗸
8 7. 8.	•••• <u>%</u> cor	ncrete_data   Training (8	30%)	
INSTANCES 824	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIEL		FIELD CODINGS
	VARIABLES: Filter bias or predictors		SIGNIFICANCE LEVEL:	.1 💌
Bias and predictors		tj Туре	Coefficients	tį.
Bias		123	-34.87790	σ
cement		123	0.12173	Significant coefficient
blast_furnace_slag		123	0.10651	
fly_ash		123	0.09088	
water		123	-0.11988	s o
superplasticizer		123	0.39687	σ

Figure 3.63: Significance icons for coefficient estimates

Next to each icon indicating the significance of a coefficient, you will find a  $\sigma$  symbol. If you mouse over it, a tooltip will display a summary of the stats for that coefficient. (See Figure 3.64.) First, you will find the *p*-value from the Wald test⁶. As mentioned in the previous point, this *p*-value is compared against the selected significance level to determine the coefficient's significance. Then, associated with the Z score chart, you will find the Z score value, the confidence interval for a 95% confidence and the standard error, or variance, of the coefficient estimate.

⁶https://en.wikipedia.org/wiki/Wald_test

Sources Datasets Supervised - Unsupervise	ed ▼ Predictions ▼ Tasks	WhizzML 👻
A T. B	concrete_data   Training (80%)	
INSTANCES         DEFAULT NUMERIC           824         NOT DEFINED	OBJECTIVE FIELD CONCRETE_COMPRESSIVE_STRENGTH	BIAS FIELD CODINGS
VARIABLES: Filter bias or predictor	rs SIGNIFI	
Bias and predictors	tj Type t	Coefficients
Bias	123	-34.87790 🛆 🛛
cement	123	0.12173 🛆 🛛
blast_furnace_slag	123	0.10651 🛆 💿
fly_ash	123	0.09088 P-VALUE (WALD TEST): 0
water	123	-0.11988
superplasticizer	123	0.39687
coarse_aggregate	123	0.02093
fine_aggregate	123	0.02267 STD. ERROR 1.14e-2 0
age	123	0.11037 🛆 🛛
Show 25 🗘 variables	1 to 9 of 9 variables	

Figure 3.64: Summary of stats per coefficient

You can download all the **stats** information by clicking in the download CSV icon in the top menu to the right. (See Subsection 3.7.1.)

Additional options for the table include a filtering option and an export option:

• You can filter the table first column by field name, class, term or item using the **search** box at the top of the table (see Figure 3.65.)

Sources Datasets	Supervised - Unsupervised -	Predictions 🕶 Tasks	_	WhizzML 🔻
a   17. <mark>1</mark> 0.	•••• 🥢 con	crete_data   Training (80%)	$\mathbf{Q}$	ૐ- (≡)⁵- ③
INSTANCES 824	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD	IGTH BIAS	FIELD CODINGS
	VARIABLES: Filter bias or predictors	s	IGNIFICANCE LEVEL: 0.1	CSV
Bias and predictors		1, Туре	Coefficients	ţ.
Bias		123	-34.87790	σ
cement		123	0.12173	
blast_furnace_slag		123	0.10651	σ
fly_ash		123	0.09088	σ
water		123	-0.11988	σ
superplasticizer		123	0.39687	σ
coarse_aggregate		123	0.02093	σ
fine_aggregate		123	0.02267	σ
age		123	0.11037	σ

Figure 3.65: Search and filter linear regression table

• You can also export the table in a CSV file by clicking on the icon highlighted in Figure 3.66.

Sources Datasets	Supervised - Unsupervised -	Predictions - Tasks		WhizzML 👻
₽ <b>₽</b> . <b>₽</b> . ••	•• 🥢 conc	rete_data   Training (80%)	Ŷ	(≡) [*] , (≡) [*]
INSTANCES 824	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD	BIAS YES	FIELD CODINGS
VAR	IABLES: Filter bias or predictors	SIGNIFIC	ANCE LEVEL: 0.1 V	Ţ
Bias and predictors		†↓ Туре †↓	Coefficients	ţ
Bias		123	-34.87790	σ
cement		123	0.12173	△ o
blast_furnace_slag		123	0.10651	σ
fly_ash		123	0.09088	σ
water		123	-0.11988	σ
superplasticizer		123	0.39687	σ
coarse_aggregate		123	0.02093	σ
fine_aggregate		123	0.02267	s o
age		123	0.11037	<b>△</b> σ

Figure 3.66: Export table in CSV file

## 3.6 Linear Regression Predictions

## 3.6.1 Introduction

The ultimate goal in building a linear regression is being able to make predictions with it. In BigML, you can make predictions for **single instances** or for **many instances in batch**. Each prediction comes with a measure, prediction interval, indicating the 95% confidence range for the predicted value.

The predictions tab in the main menu of the BigML Dashboard is where all your saved predictions are listed. (See Figure 3.67.) You can **search** your predictions by name clicking on the search option on the top menu. In the predictions list view, you can see, for each prediction, the **linear regression** icon used for the prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created).

Source	s Datasets Supervised Visupervised	i 🔻 🛛 Pre	dictions 👻 Tasks			V	/hizzML 🔻	
ŧ	Ē		Predictions				â q	
ាំ	Name	1	Objective	4	Prediction	η.		t↓.
~	diabetes   Training (80%) operating kind=probability, 8 inputs		plasma glucose		177.43172		5h 47m	nin
~	iris.csv operating kind=probability, 4 inputs		petal width		2.03351		6h 1m	nin
~	abalone_csv   Training (80%) with AutoOP operating kind=probability, 8 inputs		Class_number_of_rings		11.1189		3d 16	6h
~	abalone_csv   Training (80%) with AutoOP operating kind=probability, 8 inputs		Class_number_of_rings		14.21287		4d 1	5h
<u>.</u>	concrete_data   Training (80%) operating kind=probability, 8 inputs		concrete_compressiv		92.45896		2w4	4d

Figure 3.67: Predictions list view

When you first create an account at BigML, or every time that you start a new project, your list of predictions will be empty. (See Figure 3.68).

Sources	Datasets	Supervised -	Unsupervised -	Pre	dictions - Tasi	ks			Wh	hizzML	-
ŧ					Predictions				Ī	Î	0
សំ	Name			4	Objective	4	Prediction	ţ,		<b>60</b> 0 ⊞	ų.
				N	o predictions						
Show 10	predictions			N	lo predictions				К	< >	×

Figure 3.68: Empty predictions list view

Linear regression predictions are saved under the CLASSIFICATION & REGRESSION option in the menu (see Figure 3.69).

Sources	s Datasets Supervised 🗸	Unsupervised - Predictions	Tasks	١	WhizzML 👻
=	Ē	CLASSIFICATION & REGRESSION			Î Q
<del>ان</del> ا ش	Name	CLUSTER ANALYSIS	t,	Prediction	₩ ~ # \
2	diabetes   Training (80%)	Association discovery	e	177.43172	5h 51min
2	iris.csv operating kind=probability, 4 inputs	とう TOPIC MODELING 使 PCA PROJECTIONS		2.03351	6h 5min
2	abalone_csv   Training (80%) with A operating kind=probability, 8 inputs		nber_of_rings	11.1189	3d 16h
<u></u>	abalone_csv   Training (80%) with A operating kind=probability, 8 inputs	DOP Class_num	nber_of_rings	14.21287	4d 15h
<u></u>	concrete_data   Training (80%) operating kind=probability, 8 inputs	concrete_	compressiv	92.45896	2w 4d

Figure 3.69: Menu options of the predictions list view

Select the list for your single instances predictions or your batch predictions by clicking on the corresponding icons. (See Figure 3.70 and Figure 3.71.)



Figure 3.70: Single predictions icon



Figure 3.71: Batch predictions icon

## 3.6.2 Creating Linear Regression Predictions

BigML provides two options to predict with your linear regressions explained in the following subsections:

- PREDICT: to predict one single instance
- BATCH PREDICTION to predict multiple instances in batch.

#### 3.6.2.1 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the input fields used by the linear regression, so you can easily set the values and get an immediate response.

Follow these steps to create a single prediction:

1. Click PREDICT in the linear regression 1-click action menu. (See Figure 3.72)

Sources Datasets	Supervised - Unsupervise	ed 🕶 Predictions 👻 Tasl	ks		WhizzML 👻
8 ¥. 87.	•••• Predicti	ing abalone age from me	asurements	Q (3-	(≡)⁵- ()
4,177	DEFAULT NUMERIC NOT DEFINED	CLASS_NUMBER	PREDICT		ODINGS ES
	20	8	CREATE FUSION		
Y Class_number_of_r	ings 🔹	•	MOVE TO	CTION	

Figure 3.72: Predict using the 1-click action menu

Alternatively, click PREDICT in the **pop up menu** in the list view. (See Figure 3.73)

<u>;</u> ;;			<u>/</u> Lir	near Regressions			8	ð 🛍	Q,
11	Name			Objective 1	🛗 t,	<b>1</b>	<b>\$</b>	<b>i</b>	
I <mark>R</mark>	abalone dataset witho No extra information	out bias	/ 🗵	Class_number_of_rings	3h 1min	191.6 KB			
R	abalone_csv   Training bias			lass_number_of_rings	5h 6min	153.2 KB			
I <mark>P</mark>	<b>diabetes   Training (80</b> bias	CREATE BATCH PREDICTION	1	lasma glucose	6h 15min	20.4 KB	1	1	1
R	iris.csv bias	CREATE FUSION		etal width	6h 22min	4.5 KB	1		
I <mark>P</mark>	abalone_csv   Training	VIEW DETAILS	N	lass_number_of_rings	1d 1h	153.2 KB			
R	abalone_csv   Training			>:lass_number_of_rings	4d 23h	153.2 KB			
I <mark>P</mark>	concrete_data   Trainin No extra information	ng (80%) without bias		concrete_compressiv	4d 23h	46.1 KB			1
I <mark>P</mark>	abalone_csv   Training auto-optimized, model o	(80%) with AutoOP candidates=128, max. training tim		Class_number_of_rings	5d 5h	153.2 KB	2		1
I <mark>P</mark>	concrete_data   Trainin auto-optimized, model o	ng (80%) with AutoOP candidates=128, max. training tim		concrete_compressiv	5d 5h	46.1 KB			1
R	arrhythmia bias			class	6d 5h	363.0 KB			

Figure 3.73: Predict using the pop up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the linear regression as input fields. (See Figure 3.74.)

<b>.</b>	Predic	t using Predi	cting abalone age	from me	<b>G</b> -	(≡)∕~_ (
	Class_number_c	of_rings: 13.50	)	4.64		
					All inp	ut fields: 🕑
Sex			Length			<b></b>
F		¥	0.0	6	1.0	0.49
Diameter		€	Height			۲
0.0	0.8	0.39	0.0	0	1.41	0.69
Whole_weight		2	Shucked_weight			۲
0.0	3.53	1.72	0.0		1.86	0.9
/iscera_weight		<b></b>	Shell_weight			۲
0.0	0.95	0.46	0.0	0	1.26	0.61
New prediction name						

Figure 3.74: Linear regression prediction form

- 3. Set input values for your selected fields. BigML supports numeric, categorical, text and items fields as inputs.
- 4. **Get the prediction** at the top of the view along with the prediction interval. (See Figure 3.75.) BigML predictions are synchronous, i.e., when you send the input data, you get an immediate response. Moreover, single predictions from the BigML Dashboard are performed locally, so unless you save your prediction, it will not consume any credits and it will be updated instantly when you change your input values. Learn more about local predictions in Subsection 3.6.4.1.

Sources Datasets	Supervised 🗸	Unsupervised	- Predicti	ons - Tasks			WhizzML 👻
¥.		Predict	using Predic	cting abalone age	from me	<b>3</b> -	(≡)⁵, ③
	Clas	s_number_of	_rings: 13.50	)	4.64		
						All inp	ut fields: 🗹
Sex			<b></b>	Length			<b></b>
F			•	0.0	-0	1.0	0.49
Diameter			<b>S</b>	Height			<b>S</b>
0.0	0	0.8	0.39	0.0	0	1.41	0.69
Whole_weight			<b>S</b>	Shucked_weight			<b>S</b>
0.0	0	3.53	1.72	0.0	0	1.86	0.9
Viscera_weight			<b></b>	Shell_weight			<b>S</b>
0.0	6	0.95	0.46	0.0	-0	1.26	0.61
New prediction name							
Predicting abalone a	ge from measurem	ents				S S	ave

Figure 3.75: Get the linear regression prediction

5. Optionally, you can Save the linear regression prediction, so you will find it afterwards in the predictions list view. (See Figure 3.76.)

	(	Class_number_of	_rings: 13.50	)	4.64		
ex			<b>⊻</b>	Length		All inp	ut fields: 🕑
F			*	0.0		1.0	0.49
liameter			€	Height			٩
0.0	8	0.8	0.39	0.0	6	1.41	0.69
/hole_weight				Shucked_weight			٩
0.0	0	3.53	1.72	0.0	6	1.86	0.9
iscera_weight			€	Shell_weight			0
0.0	-6	0.95	0.46	0.0	-0	1.26	0.61

Figure 3.76: Save your linear regression predictions

Note: this option is only available from the BigML Dashboard for linear regressions with less than 100 fields. If you want to perform single instance predictions for a higher number of fields, use the BigML API⁷.

#### 3.6.2.2 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the linear regression you want to use to make predictions and a dataset containing the instances you want to predict. BigML will create a prediction for each instance in the dataset.

Follow these steps to create a batch prediction:

1. Click on BATCH PREDICTION option under the linear regression 1-click action menu (Figure 3.77)

Sources Datasets S	Supervised - Unsupervised -	Predictions - T	asks		WhizzML 👻
₽ ₽. ₩	Predicting	abalone age from n	neasurements	Q (5-	(≡)⁵-⊂ 🚺
INSTANCES	DEFAULT NUMERIC	OBJECTIVE	PREDICT		
4,177	NOT DEFINED	CLASS_NUMBER	BATCH PREDICTION		TES
			EVALUATE		
10 000	2D		CREATE FUSION		
			DELETE LINEAR REGRESSION		
Class_number_of_rings	•		MOVE TO		

Figure 3.77: Create batch prediction using 1-click action menu

Alternatively, click on CREATE BATCH PREDICTION in the pop up menu of the list view (Figure 3.78).

⁷https://bigml.com/api/predictions

Source	es Datasets <mark>S</mark>	upervised - Unsupervised -	Pi	redictions 🗸 Tasks				Whizzł	ИL <del>▼</del>
<i>:</i> /:			Lir	near Regressions			8	3 🛍	Q,
III	Name			Objective 1	🗰 †,	<b>a</b> 1,	\$	<b>(</b>	00
ılî	abalone dataset witho No extra information	ut bias 🧳	ø	Class_number_of_rings	3h 52min	191.6 KB			
ılî	abalone_csv   Training bias	PREDICT		lass_number_of_rings	5h 57min	153.2 KB			
ılî	diabetes   Training (80 bias	CREATE BATCH PREDICTION		lasma glucose	7h 6min	20.4 KB	1		1
ılî	<b>iris.csv</b> bias	CREATE FUSION		etal width	7h 13min	4.5 KB	1		
ılî	abalone_csv   Training No extra information	VIEW DETAILS		lass_number_of_rings	1d 2h	153.2 KB			
ılî	abalone_csv   Traininç No extra information	MOVE TO		>:lass_number_of_rings	5d	153.2 KB			

Figure 3.78: Create batch prediction using pop up menu

 Select the dataset containing all the instances you want to predict. The instances should contain the input values for the fields used by the linear regression as input fields. From this view you can also select another linear regression from the selector or even a model or ensemble by clicking on the icons on the top left menu. (See Figure 3.79.)

Sources Datasets Supervised - Unsupervised - Tasks	WhizzML 🕶
New Batch Prediction	<b>5</b> - i
Predicting abalone age from measurements       ×         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       •         *       *         *       *         *       *         *       *         *       *         *       *         *       *         *       *         * <t< th=""><th><u>م</u></th></t<>	<u>م</u>
191.5 KB 9 4,177 size fields instances	
Description:	
Prediction name:	
dataset with Predicting abalorom measurements	

Figure 3.79: Select dataset for batch prediction

3. After the linear regression and the dataset are selected, the batch prediction configuration options will appear along with a preview of the prediction output (a CSV file). (See Figure 3.80.) The default output format includes all your prediction dataset fields and adds an extra column with the class predicted. See Subsection 3.6.3 ofr a detailed explanation of all configuration options.

	d - Unsupervised - Predicti	ons 🔻 Tasks	_	WhizzML 👻
77	New Bato	h Prediction		<b>5</b> - (i
ର 🐔 🗶 🗶	₩ 22			
Predicting abalone age from	n measurements × ×	abalone_csv	or test	X V
Class_number_of_ri	Fri, 01 Mar 2019 15:35:22	IR	Tu	e, 12 Mar 2019 23:15:26
191.5 KB 9 size fields	4,177 s instances	19.2 KB size	9 fields	418 instances
Description:	$\overline{\mathbf{O}}$	Description:		୕
Sconfigure				$\overline{\mathbf{O}}$
Preview of the prediction file (using the	he type of each field) nole_weight,Shucked_weight,Visco	ra_weight,Shell_weight	,Class_number_of_rings	
Preview of the prediction file (using the Sex, Length, Diameter, Height, Wh	nole_weight,Shucked_weight,Visce 123,123,123 123,123,123 123,123,123 123,123,123	ra_weight,Shell_weight،	,Class_number_of_rings	
Preview of the prediction file (using the Sex, Length, Diameter, Height, Wh f_rings ABC, 123, 123, 123, 123, 123, 123, 123, 123	nole_weight,Shucked_weight,Visco 123,123,123 123,123,123 123,123,123 123,123,123 123,123,123 123,123,123	ra_weight,Shell_weight	,Class_number_of_rings Reset	

Figure 3.80: Configuration options for linear regression batch prediction

4. By default, BigML generates an output dataset with your batch predictions that you can later find in your datasets section in the BigML Dashboard. This option is active by default but you can deactivate it by clicking on the icon shown in Figure 3.81.

Sources Datasets Supervised - Un:	supervised - Predicti	ons 🚽 Tasks		WhizzML 🗸
A 17	New Batc	h Prediction		<b>5</b> - i
£1 F£1 <u>≈</u> ∠ ≤				
Predicting abalone age from measurem	nents x v	abalone_cs	v for test	×
Class_number_of_ri	Fri, 01 Mar 2019 15:35:22	IP		Tue, 12 Mar 2019 23:15:26
191.5 KB 9 size fields	4,177 instances	19.2 KB size	9 fields	418 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\odot$
Configure Preview of the prediction file (using the type of each of the prediction file (using the type of each of the prediction file (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the prediction file) (using the type of each of the ty	ht,Shucked_weight,Visce 3 3 3 3	ra_weight,Shell_weigh	t,Class_number_of_ring	⊙ gs,Class_number_o
Prediction name: abalone_csv for test with Predicting abalone	age from measure	u*)	Reset	Predict

Figure 3.81: Create a dataset from batch prediction

5. After you configure your batch prediction, click on the green button Predict to generate your batch prediction. (See Figure 3.82.)

Sources Datasets Supervised <del>-</del>	Unsupervised - Predicti	ons 👻 Tasks		WhizzML 🔻
1 T	New Batc	h Prediction		<b>5</b> - i
n fn 老 🗶 🕿				
Predicting abalone age from measu	irements × ×	abalone_csv	for test	×
Class_number_of_ri	Fri, 01 Mar 2019 15:35:22	1 ^R		Tue, 12 Mar 2019 23:15:26
191.5 KB 9 size fields	4,177 instances	19.2 KB size	9 fields	418 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
Configure				$\overline{\mathbf{O}}$
Preview of the prediction file (using the type o Sex, Length, Diameter, Height, Whole_we f_rings ABC, 123, 123, 123, 123, 123, 123, 123, 123	ight,Shucked_weight,Visce ,123 ,123 ,123 ,123	ra_weight,Shell_weight	,Class_number_of_ring	js,Class_number_o
Prediction name:				

Figure 3.82: Create batch prediction

6. When the batch prediction is created, you will be able to **download the CSV file** containing all your dataset instances along with a prediction for each one of them. (See Figure 3.83.)

Sources Datasets Supervised - Unsupervised - Predictions - Tasks	WhizzML 🔻
abalone_csv for test with Predicting abalone age fro	≪5- (≡)⁵- €
Predicting Abalone Age From Measurem 📀 🔲 Abalone_csv For Test	⊚
Configuration	$\overline{\mathbf{O}}$
Output preview Sex,Length,Diameter,Height,Whole_weight,Shucked_weight,Viscera_weight,Shell_weight,Class_number_of_ring f_rings,confidence interval,prediction interval F,0.545,0.425,0.125,0.768,0.294,0.1495,0.26,16,11.46291,0.16374,4.30509 F,0.55,0.44,0.15,0.8945,0.3145,0.151,0.32,19,13.1339,0.19,4.30616 M,0.43,0.35,0.11,0.406,0.1675,0.081,0.135,10,9.4649,0.16659,4.3052 M,0.5,0.4,0.13,0.6645,0.258,0.133,0.24,12,11.09861,0.15021,4.30459 I,0.355,0.28,0.085,0.2905,0.095,0.0395,0.115,7,8.23818,0.1567,4.30482 F,0.565,0.44,0.155,0.9395,0.4275,0.214,0.27,12,10.24577,0.13023,4.30394 F,0.56,0.44,0.14,0.9285,0.385,0.314,0.3,11,1141567,0.14026,4.30476	s,Class_number_o
Download batch prediction	

Figure 3.83: Download batch prediction CSV file

7. If you didn't disable the option to create a dataset explained in step 4, you will also be able to access the **output dataset** from the batch prediction view. (See Figure 3.84.)



Figure 3.84: Batch prediction output dataset

### 3.6.3 Configuring Linear Regression Predictions

BigML provides several options to configure your predictions such as setting default values for your missing numeric values (see Subsection 3.6.3.1), fields mapping (see Subsection 3.6.3.3), and output file settings (see Subsection 3.6.3.4.)

#### 3.6.3.1 Default Numeric Value

By using the **Default numeric value** before creating your batch prediction, you can easily replace all the missing numeric values by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 3.85.)

urces Datasets Supe	rvised - Unsupervised	d - Predictio	ons 👻 Tasks		WhizzML 🗸
1		New Batc	h Prediction		۵. (
ରେ 🖗	1 🗧 🛛				
Predicting abalone ag	e from measurements	×	abalone_csv	for test	* *
Class_number	_of_ri Fri, 01 Ma	r 2019 15:35:22	ılî		Tue, 12 Mar 2019 23:15:26
<b>191.5 KB</b> size		177 ances	19.2 KB size	9 fields	418 instances
Description:		◙	Description:		$\odot$
Configure					۲
Default numeric value: 😗	Select a default value v				
Excluded fields:					$\odot$
1 Make sure that the fields	in the Linear Regression	and Dataset ma	itch		
Fields mapping:				Defau	It fields 🕞
2 [OPTIONAL] Customize p	prediction output settings				

Figure 3.85: Configure Default numeric value for batch prediction

#### 3.6.3.2 Excluded Fields

You can specify which field or fields to exclude from the input data when creating your batch prediction. You search the field by typing the name and click on the field found to add to the list of exclusion. (See Figure 3.86)

ources Datasets Supervised	New Unsupervised ▼ Predict	tions 🔻 Tasks		WhizzML 🔻
1.7	New Bat	ch Prediction		<b>5</b> - (
ል 🙉 者 🔟	* 2			
Predicting abalone age from	measurements × ×	abalone_csv	for test	×
Class_number_of_ri	Fri, 01 Mar 2019 15:35:22	ıß	Τι	ie, 12 Mar 2019 23:15:26
191.5 KB 9 size fields	4,177 instances	19.2 KB size	9 fields	418 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
Configure Default numeric value: Select a	default value			۲
Excluded fields:				۲
Excluded fields: 🕖				
× Height	123 len			
Length				123
1 Make sure that the fields in the I Fields mapping:	inear Regression and Dataset n	naten	Apply this	s map 💿
	ion output cottinge			
2 [OPTIONAL] Customize predict	ion output settings			

Figure 3.86: Configure Default numeric value for batch prediction

#### 3.6.3.3 Fields Mapping

You can specify which input fields of the linear regression match with the fields in the dataset contaning the instances you want to predict. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** by clicking on the green switcher. Additionally, you can **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the batch prediction. (See Figure 3.87.)

ିଅରୁ Configure		۹
Default numeric value: Select a default value		
Excluded fields:	0	•
1 Make sure that the fields in the Linear Regression and Dataset		
Fields mapping:	Apply this map	۲
Linear regression fields	Dataset fields Auto-mapping by: NAME COM	
2. Length 123	2 Length 123 × •	
3. Diameter 123	3. Diameter 123 × v	
4. Height	4. Height 123 × v	
5. Whole_weight 123	5. Whole_weight 123 × v	
6. Shucked_weight 123	6. Shucked_weight 123 × v	
Excluded fields will not be taken into account to compute predictions		
2 [OPTIONAL] Customize prediction output settings		
Output settings	6	•

Figure 3.87: Configure the fields mapping for batch prediction

# Note: Fields mapping from the BigML Dashboard is limited to 200 fields. For batch predictions with a higher number of fields, map your fields using the **BigML API**⁸.

#### 3.6.3.4 Output Settings

Batch predictions return a CSV file containing all your instances and the final predictions. Tune the following settings to customize your prediction file (see Figure 3.88):

- **Separator**: this option allows you to choose the best separator for your output file columns. The default separator is the comma. You can also select the semicolon, the tab, or the space.
- New line: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".
- Output fields: by clicking on the list icon next to the separator selector, you can include or exclude all your dataset fields from your output file. You can also individually select the fields you want to include or exclude using the multiple output fields selector. Note: a maximum of 100 fields can be displayed in this selector, but all your dataset fields will be included in the output file by default unless you exclude them.
- **Headers**: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability column name, etc.). By default, BigML includes the headers.
- **Prediction column name**: customize the name for your predictions column. By default, BigML takes the name of the linear regression's objective field.
- **Confidence bounds**: this option allows you to include two additional columns with the confidence interval and prediction interval. By default they are not included in your ouput file.
- **Confidence interval column name**: customize the name for the confidence interval column if you include it in the output file. BigML sets "confidence interval" as the default name.

 $^{^{8} \}tt https://bigml.com/api/batchpredictions\#bp_batch_prediction_arguments$ 

• **Prediction interval column name**: customize the name for the prediction interval column if you include it in the output file. BigML sets "prediction interval" as the default name.

eparator:	New line:	_		
, (comma) 🔻	Unix, Linux or OS X (LF)	1		
Prediction column name:	Confidence interval column name:	Predictio	on interval column name:	
, b,c , -, -				
utput Fields:				
* Sex	ABC × Length	123	* Diameter	123
× Height	123 x Whole_weight	123	* Shucked_weight	123
× Viscera_weight	123 x Shell_weight	123	* Class_number_of_rings	123
iew of the prediction file (using the	type of each field)			
	e_weight,Shucked_weight,Viscera_weigh	t,Shell_wei	ght,Class_number_of_rings,	Class_number.
	e_weight.Shucked_weight.Viscera_weigh	t,Shell_wei	ght,Class_number_of_rings,	Class_number

Figure 3.88: linear regression output settings for batch predictions

#### 3.6.3.5 Prediction explanation

Prediction explanation helps understand why a linear regression makes a certain prediction. This is very useful in many applications, and the reasons behind a prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

When creating single linear regression prediction, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Save (see Figure 3.89).

Sources Datasets Supervised	Unsupervised	Predict	ions 🔻 Tasks			WhizzML 🔻
<b>T</b> *	Predict	using Predi	cting abalone age	from me	<b>G</b> -	(≡)∕~_ (≣
(	Class_number_of	_rings: 11.6	5	4.66		
					All inp	ut fields: 🗹
Sex		≤	Length			<b></b>
1		¥	0.0	6	1.0	0.51
Diameter		€	Height			<b>S</b>
0.0	0.8	0.38	0.0	6	1.41	0.69
Whole_weight		<b></b>	Shucked_weight			<b></b>
0.0	3.53	1.7	0.0		1.86	1.05
Viscera_weight		<b></b>	Shell_weight			<b></b>
0.0	0.95	0.24	0.0	6	1.26	0.6
New prediction name				explanatio	npute the <b>prediction</b> on when saving the prediction	
Predicting abalone age from measu	urements				E. s	ave

Figure 3.89: Prediction explanation

The prediction explanation represents the most important factors considered by the linear regression in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 3.90. The importances across all input fields should sum 100%.

₽ ₽.		Predicting abalone age from n	neasurements	
		Class_number_of_rings: 11.65	4.66	
PREDICTION EXPLANA	TION			
nput data		Importance		
Shucked_weight	1.05			57.49% +
Whole_weight	1.70			30.07% +
leight	0.69			9.61% +
hell_weight	0.60			2.08% +
/iscera_weight	0.24			0.00%
liameter	0.38			0.00%
ex	I			0.00%
.ength	0.51			0.00%



You can export the prediction explanation to a PNG image file, a CSV file or a JSON file by clicking the top right icons respectively.

## 3.6.4 Consuming Linear Regression Predictions

#### 3.6.4.1 Local Predictions

Local predictions are provided for single instances from the BigML Dashboard which are performed faster at no cost. Local predictions allow you to get a real-time prediction without consuming any credits or requiring any internet connection. This is possible because the linear regression is **saved in-memory**, so when the input values change, BigML is able to compute predictions in microseconds.

In addition to BigML Dashboard, you can fully use single and batch predictions via the BigML API and bindings. The following subsections explain both tools.

#### 3.6.4.2 Using Linear Regression Predictions via the BigML API

Linear regression predictions have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, and delete single and batch predictions.

In the example below, see how to create a single prediction using a linear regression and defining the input data once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"linearregression": "linearregression/5c79513a983efc522f000009",
                    "input_data": {"000003":0.61, "000004":1.58, "000005":1.15,
                    "000007":0.55}}}'
```

For more information on using linear regressions through the BigML API, please refer to the documentation⁹.

#### 3.6.4.3 Using Linear Regression Predictions via BigML Bindings

You can also create, configure, retrieve, list, update, and delete single and batch predictions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a linear regression with the Python bindings.

```
from bigml.api import BigML
api = BigML()
prediction = api.create_prediction(
    "linearregression/5c702c91983efc4cc6000016",
    {"age": 230, "cement": 326.81, "blast_furnace_slag": 205.33, "fly_ash":105.17})
```

For more information on BigML bindings, please refer to the bindings page¹⁰.

### 3.6.5 Descriptive Information

Each linear regression prediction has an associated **name**, **description**, **category**, and **tags**. You can find a brief description of each concept in the following subsections. The MORE INFO menu option displays a panel that provides editing options. (See Figure 3.91)

⁹https://bigml.com/api/linearregressions

¹⁰https://bigml.com/tools/bindings

Sources Datasets Supervised - Unsupervised -	Prediction	s 🔻 Tasks	WhizzM	IL <del>-</del>
Predicting a	ibalone age	from measurements	<b>८∌</b> ⊷ (≡) ^{\$} •	0
DETAILS				۹
Details:	$\overline{\mathbf{O}}$	Fields:	C	
INFO				۹
Name: Predicting abalone age from measurements		Category: Physical, Earth & Life Sciences	v	
Description:		Tags:		
Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope - a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age.		marine life abalone		•
Source Department of Computer Science, University of Tasmania				
PRIVACY				$\overline{\mathbf{O}}$

Figure 3.91: linear regression prediction descriptive information

#### 3.6.5.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: BigML uses the linear regression name "<linear regression name>".
- **Batch predictions**: BigML combines your prediction dataset name and the linear regression name: "<linear regression name> with <dataset name>".

Predictions names are displayed on the list and also on the top bar of a prediction view. Predictions names are indexed to be used in searches. Rename your predictions any time from the MORE INFO menu.

The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, but they will always have different identifiers.

#### 3.6.5.2 Description

Each prediction also has a **description** that it is very useful for documenting your Machine Learning projects. Predictions take their description from the linear regression used to create them.

Descriptions can be written using plain text and also markdown¹¹. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 3.92.)

¹¹https://en.wikipedia.org/wiki/Markdown

B I ≔ ⊘	0
age of aba cone, stai microscope	the age of abalone from physical measurements. The lone is determined by cutting the shell through the ning it, and counting the number of rings through a $\cdot$ - a boring and time-consuming task. Other ts, which are easier to obtain, are used to predict
Description:	
determined b rings through	age of abalone from physical measurements. The age of abalone is y cutting the shell through the cone, staining it, and counting the number of a microscope a boring and time-consuming task. Other measurements, ier to obtain, are used to predict the age.
Source	
Department of	f Computer Science, University of Tasmania

Figure 3.92: Markdown editor for linear regression prediction descriptions

Descriptions cannot be longer than 8192 characters.

#### 3.6.5.3 Category

A **category** taken from the linear regression used to create it is associated with each prediction. Categories are useful to classify predictions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 3.5.

#### 3.6.5.4 Tags

A prediction can also have a number of **tags** associated with it. These tags help to retrieve the prediction via the BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the linear regression used to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

#### 3.6.6 Linear Regression Predictions Privacy

The link displayed in the **Privacy** panel is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared by using a secret link. (See Figure 3.93.)

Sources Datasets	Supervised - Unsupervised -	Predictions - Tasks	WhizzML 🗸
₽ ₽₽- ••••	abalone_csv	for test with Predicting abalone age fi	ro (=) ⁵ , (i)
DETAILS			◙
INFO			$\odot$
PRIVACY			۲
Private link   Only you can	view this batchprediction logged into your	account.	
	https://strato.	.dev.bigml.com/dashboard/batchprediction/5c88	539c983efc4a190000b2



## 3.6.7 Moving Linear Regression Predictions to Another Project

When you create a prediction, it will be assigned to the same **project** where the original linear regression is located. You cannot move predictions between projects as you do with other resources.

## 3.6.8 Stopping Linear Regression Predictions

Single predictions are **synchronous** resources, so you cannot cancel them during the creation since you get the result immediately.

By contrast, batch predictions are **asynchronous** resources, so you can stop their creation before the task is finished. Use the DELETE BATCH PREDICTION option from the **1-click action menu** (Figure 3.94) or from the **pop up menu** on the list view.



Figure 3.94: Stop linear regression batch prediction from 1-click action menu

A modal window will be displayed asking you for confirmation. If you stop the prediction during its creation you won't be able to resume the same task again, so if you want to create the same prediction you will have to start a new task.

Sources Datasets	Are you sure you want to delete this prediction?	WhizzML 👻
a 😨 🚥	If you delete this prediction, you will no longer have access to its data and you will need to recreate it.	<u>ن</u> (=) ⁴ ر (=)
Predicting A	Cancel Delete	
Configuration		

Figure 3.95: linear regression delete prediction confirmation

## 3.6.9 Deleting Linear Regression Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the **1-click action menu** (see Figure 3.96) or using the **pop up menu** on the predictions list view (see Figure 3.97).

Sources Datasets Supervised - Unsupervised - Tasks	WhizzML 🗸
Batch Predicting abalone age from measurements	(≡) [€] , (Ξ)
Predicting Abalone Age From Measurem	
Configuration	$\odot$
Output preview Sex,Length,Diameter,Height,Whole_weight,Shucked_weight,Viscera_weight,Shell_weight,Class_number_of_rings f_rings M,0.455,0.365,0.095,0.514,0.2245,0.101,0.15,15,9.21992 M,0.35,0.265,0.09,0.2255,0.0995,0.0485,0.07,7,7.48674 F,0.53,0.42,0.135,0.677,0.2565,0.1415,0.21,9,11.09457 M,0.44,0.365,0.125,0.516,0.2155,0.114,0.155,10,9.65131 I,0.33,0.255,0.08,0.205,0.0895,0.0395,0.055,7,6.73625 I,0.425,0.3,0.095,0.3515,0.141,0.0775,0.12,8,7.81351 F,0.53,0.415,0.15,0.7775,0.237,0.1415,0.33,20,13,53673	s,Class_number_o
Download batch prediction	

Figure 3.96: Linear regression delete prediction from 1-click menu

Source	es D	atasets Supervised - Predictions - Tasks		Whi	zzML 👻
ŧ	F	Batch Predictions		Q	<b>.</b> .
ណ	ill	Name	Objective 1	Instances $\uparrow_{\downarrow}$	- 🛗 - İ,
~	ılî	abalone_csv with Predicting abalone age from measurements operating kind=probability, use all fields	Class_number_of_rings	4K+	27min
<mark>9</mark>	ılî	Batch Predicting abalone age from measurements operating kind=probability, use all fields	Class_number_of_rings	4K+	30min
<u></u>	ılı	iris.csv with iris.csv with sepal length as wei operating kind=probability, use all fields	etal width	150	3h 27min
~	ılî	iris.csv with iris.csv 3 fields with sepal lengt	petal width	150	3h 30min
<u></u>	ılı	abalone_csv for test with Predicting abalone age from measurements operating kind=probability, use all fields	Class_number_of_rings	418	20h 1min
<u> </u>	ılı	abalone_csv for test with Predicting abalone age from measurements operating kind=probability, use all fields	Class_number_of_rings	418	20h 41min

Figure 3.97: Linear regression delete prediction from pop up menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

Sources Datasets	Are you sure you want to delete this prediction?	
A	If you delete this prediction, you will no longer have access to its data and you will need to	డం (≡)*- ౹
Predicting #	recreate it. Cancel Delete	

Figure 3.98: Linear regression delete prediction confirmation

## 3.7 Consuming Linear Regressions

Similarly to other models in BigML, linear regressions are white-boxed models, so you can **download** them and used them locally to make predictions. You can also create and consume your linear regressions programmatically via the **BigML API and bindings**. The following subsections explain those three options.

## 3.7.1 Downloading Linear Regressions

You can download your linear regression in several programming languages including JSON PML, Python or Node.js. By downloading your linear regression you will be able to compute **predictions locally**, free of latency and at no cost. Click on the download icon in the top menu (see Figure 3.99), and select your preferred option (see Figure 3.100)

Sources Dataset	s Supervised - Unsupervised -	<ul> <li>Predictions          <ul> <li>Tasks</li> </ul> </li> </ul>	Download Actionable Linear WitzzML - Regression	
8 ¥• 8•	•••• Predicting	g abalone age from measurements	Ġ.	(≣)⁵₊ (≣
4,177	NOT DEFINED	CLASS_NUMBER_OF_RINGS	YES	YES

Figure 3.99: Click download icon

Actionable Linear Reg	Predicting abalone age from measurements	(3) (3) ×	(≡) [≸]
4,177 Complete actionable linear r	egression:		CODI YES
Select the language	*		
Python Node.js			
JSON PML	a language using the selector above.		
16		Close	
16		Close	

Figure 3.100: Select language to download linear regression

## 3.7.2 Using Linear Regressions Via the BigML API

linear regression have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them for predictions.

In the below example, see how to create a linear regression using an existing dataset once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/linearregression?\$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"dataset": "dataset/5c7057a9983efc4cc2000005"}'
```

For more information on using linear regressions through the BigML API, please refer to the documentation¹².

### 3.7.3 Using Linear Regressions Via the BigML Bindings

You can also create and use linear regressions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a linear regression with the Python bindings.

```
from bigml.api import BigML
api = BigML()
linearregression = api.create_linear_regression(
        'dataset/5c87d50d983efc4a02000085', {"objective_field":"plasma glucose"})
```

For more information on BigML bindings, please refer to the bindings page¹³.

## 3.8 Linear Regression Limits

There are some limits that apply for the creation of any BigML resource. These are limits based on the number of classes, terms and items that can be considered to create your models. This is explained in Subsection 3.8.0.1.

Additionally, some specific limits apply for your linear regressions **visualization**, i.e. to the linear regression chart and the coefficient table views, depending on the number of input fields in your dataset. See Subsection 3.8.2 and Subsection 3.8.1 for a detailed explanation.

Note: chart limits and coefficient table limits just affect to the visualization of the model, i.e., despite your dataset reach those limits, you can still creating the linear regression, evaluating it and using it to make predictions.

#### 3.8.0.1 Field Limits

Linear regression, similarly to other BigML models, has the following limitations according to the type of field:

- Classes: for categorical, a maximum number of 1,000 distinct classes per field is allowed.
- **Terms**: BigML can handle up to 1,000 terms in total. If multiple text fields are defined, then the token limit per field is evenly divided by the number of text fields evenly, e.g., a dataset with two text fields would result in 500 terms per text field. BigML selects those terms with most significant frequency, discarding both those that appear either too often or too infrequently. A maximum of 256 characters per term is allowed.
- Items: a maximum number of 10,000 distinct items per field is allowed.

### 3.8.1 Chart Limits

There are some circumstances under which your chart cannot be displayed:

As the 1D chart only supports numeric fields for the x-axis, if your linear regression only contains categorical, text, or items fields, the 1D chart cannot be displayed. When you try to click on the 1D chart icon you will see a warning message. (See Figure 3.101)

¹²https://bigml.com/api/linearregressions

¹³https://bigml.com/tools/bindings

Sources Datasets	Supervised  Unsupervised	✓ Predictions ✓ Tasks		WhizzML 👻
	•••	Loan_risk_data	Ŷ	
ur resource does not contain y numeric field to be included in the x-axis.	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE FIELD CREDIT_AMOUNT	BIAS	FIELD CODINGS YES
<b>Y</b> purpose	ABC -	$\mathcal{G}$	PREDICTION	
used car		- min.		max.
retraining repairs				
radio/tv				
other			INPUT FIELDS	All fields: 🗹

Figure 3.101: Warning message when the linear regression does not have any numeric input field

 If your linear regression contains more than 800 predictors, the prediction interval bounds cannot be displayed in the chart. You will see the warning message shown in Figure 3.102. You can still see your coefficients by downloading the CSV file. For how to calculate the number of predictors in a linear regression, see Subsection 3.2.3.



Figure 3.102: Warning message when the linear regression has more than 800 predictors

## 3.8.2 Coefficient Table Limits

If your linear regression contains more than **1,000 input fields**, the coefficient table cannot be displayed. You will need to **download the CSV** if you want to see your linear regression coefficients. You will get the message shown in Figure 3.103:



Figure 3.103: Warning message when the coefficient table limits are reached

## 3.9 Descriptive Information

Each linear regression has an associated **name**, **description**, **category**, and **tags**. The following subsections provide a brief description for each concept. In Figure 3.104, you can see the options the MORE INFO menu provides to edit them.



Figure 3.104: Edit linear regression descriptive information

## 3.9.1 Linear Regression Name

Each linear regression has a name that is displayed in the linear regression list view and also on the top bar of the linear regression view. linear regression's names are indexed to be used in searches. When you create a linear regression, it gets a default name. Change it using the MORE INFO menu option on the right corner of the linear regression view. The name of a linear regression cannot be longer than **256**
characters. More than one linear regression can have the same name even within the same project, but they will always have different identifiers.

# 3.9.2 Description

Each linear regression also has a **description** that it is very useful for documenting your Machine Learning projects. linear regressions take the description of the datasets used to create them by default.

Descriptions can be written using plain text and also markdown¹⁴. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 3.105.)

<i>, , , , , , , , , ,</i>	Ø ⟨⟩
age of al cone, sta microscop	ng the age of abalone from physical measurements. The palone is determined by cutting the shell through the aining it, and counting the number of rings through a pe a boring and time-consuming task. Other ents, which are easier to obtain, are used to predict
escription:	
determined rings throug	he age of abalone from physical measurements. The age of abalone is by cutting the shell through the cone, staining it, and counting the number of h a microscope – a boring and time-consuming task. Other measurements, asier to obtain, are used to predict the age.
Source	

Figure 3.105: Markdown editor for linear regression descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

# 3.9.3 Category

A **category**, taken from the dataset used to create it, is associated with each linear regression. Categories are useful to classify linear regressions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A linear regression category must be one of the 24 categories listed on Table 3.5.

¹⁴https://en.wikipedia.org/wiki/Markdown

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 3.5: Categories used to classify linear regression by BigML

# 3.9.4 Tags

A linear regression can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide linear regressions with some extra information. A linear regression inherits the tags from the dataset used to create it. Each tag is limited to a maximum of 128 characters. Each linear regression can have up to 32 different tags.

### 3.9.5 Counters

For each linear regression, BigML also stores a number of counters to track the number of other resources that have been created using the linear regression as a starting point. In the linear regression view, you can see a menu option that displays counters for evaluations, single and batch predictions, and the fusions created. It also allows you to quickly jump to all the resources of one type. (See Figure 3.106)

Sources	Datasets	Supervised - Unsupervised -	Predictions - Tasks	_	WhizzML 🔻
8 4		•••• Predicting at	palone age from measurements	<b>\$</b>	(≣)⁵₊ ()
INSTAN	2	VIEW 2 PREDICTIONS FROM THIS LINEAR RE	OBJECTIVE FIELD	BIAS	FIELD CODINGS
4,177	<b>E</b> 4	VIEW 4 BATCH PREDICTIONS FROM THIS LINE	CLASS_NUMBER_OF_RINGS	YES	YES
	<b>1</b>	VIEW 1 EVALUATION FROM THIS LINEAR REG			
	2	NO FUSIONS THAT INCLUDE THIS LINEAR RE			

Figure 3.106: Counters for linear regressions

# 3.10 Linear Regression Privacy

Privacy options for a linear regression can be defined in the **More Info** panel, displayed in Figure 3.107. There are two levels of privacy for BigML linear regressions:

- **Private**: only accessible by authorized users (the owner and those who have been granted access by him or her).
- · Shared: accessible by any user with whom the owner shares the secret link.

Sources Datasets Supervised	Unsupervis	ed • Predictions •	Tasks	_	WhizzML 👻
A Tr. 60,	Predict	ting abalone age fro	om measurements	ф 👶-	(≡) ^{\$} , (i),
DETAILS					$\overline{\mathbf{O}}$
INFO					$\odot$
PRIVACY					۲
Private link   Only you can view this linear	regression logged in	to your account.			
	https://s	strato.dev.bigml.com/dasl	hboard/linearregression/5c7951	3a983efc522f000009	
Secret link   Only people with the secret li	nk can view this linea	r regression.			
SHARED	https://s	strato.dev.bigml.com/sha	red/linearregression/53h5HA9Dx	Lmav5KGQi5epKvJKhd	
	> <iframe< td=""><td>src="https://strato.dev.bi</td><td>gml.com/embedded/linearregre</td><td>ssion/53h5HA9DxLmav5K.</td><td></td></iframe<>	src="https://strato.dev.bi	gml.com/embedded/linearregre	ssion/53h5HA9DxLmav5K.	

Figure 3.107: linear regression privacy

# 3.11 Moving Linear Regressions to Another Project

When you create a linear regression, it will be assigned to the same project where the original dataset is located.

linear regressions can only be assigned to a single project. However, you can move linear regressions between projects. The menu option to do this can be found in two places:

1. From the linear regression list, in either the chart view or table view, click the MOVE TO... option within the **1-click action menu** and select another project or create a new one. (See Figure 3.108.)

Sources Datasets	Supervised - Unsupervised -	Predictions - T	asks		WhizzML 🔻
a 17. 87. ••	••• Predicting ab	alone age from n	neasurements	ф <u></u> б,	(≡)⁵, (i)
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE CLASS_NUMBER	PREDICT		ODINGS
	RIABLES: Filter bias or predictors		CREATE FUSION		CSV
Bias and predictors	NEW PROJECT		MOVE TO		ţ,
Bias	PROJECTS		3.90015		Δσ
Sex = M	DEMO		0.05522		σ
Sex = I	PROJECT LNR		-0.82452		σ
Sex = F	(	ABC ABC	0		
Length		123	-0.47157		σ

Figure 3.108: Change project from 1-click action menu

2. In the linear regression list view, click the MOVE TO... option within the **pop up menu** and select another project or create a new one. (See Figure 3.109.)

Source	es Datasets <mark>S</mark>	NEW Supervised - Unsupervised -	- Pr	redictions 👻 Tasks				WhizzN	1∟ -
<u>:/</u> :		<u></u>	🖉 Lir	near Regressions			8	8 🛍	Q,
ill	Name			Objective 1	<b>**</b> 1,	â ţ	<b>\$</b>		
ılî	iris.csv 3 fields with se bias, weight field: 0000	epal length as weight field	/ 🗵	petal width	11h 27min	4.5 KB		1	
ılî	iris.csv with sepal len bias, weight field: 0000			etal width	11h 30min	4.5 KB		1	
ılî	abalone dataset with No extra information	CREATE BATCH PREDICTION		lass_number_of_rings	1d 9h	191.6 KB			
ılî	abalone_csv   Training bias	6 CHEATER COICH		lass_number_of_rings	1d 11h	153.2 KB			
ılî	diabetes   Training (80 bias	VIEW DETAILS		lasma glucose	1d 12h	20.4 KB	1	1	1
ılî	iris.csv bias	MOVETO		NEW PROJECT		4.5 KB	1		
ılî	abalone_csv   Training No extra information	g (80%) without bias		PROJECTS		3.2 KB			
ılî	abalone_csv   Training	g (80%) without bias		DEMO PROJECT LNR		3.2 KB			
ılî	concrete_data   Trainin No extra information	ng (80%) without bias		concrete_compressiv	6d 6h	46.1 KB			1

Figure 3.109: Change project from pop up menu

# 3.12 Stopping Linear Regressions

You can stop the creation of a linear regression before the task is finished by clicking the DELETE LINEAR REGRESSION option from the **1-click action menu** (see Figure 3.110), or from the **pop up menu** in the linear regression list view (see Figure 3.111).



Figure 3.110: Stop linear regression creation from 1-click action menu

Source	es Datasets S	upervised - Unsupervised	<ul> <li>Predictions - Tasks</li> </ul>				Whizz∿	/iL <del>-</del>
<u>;;;</u>			Linear Regressions			Q	ŝ â	Q,
ill	Name		Objective	tin ti	₿ ţ	<b>\$</b>	<b>(</b>	
ılî	1001fields In progress: 40%		8 -	2min	35.6 MB			
ıl <mark>e</mark>	Loan_risk_data	PREDICT	redit_amount	4h 11min	135.8 KB			
ılî	Loan_risk_data	CREATE BATCH PREDICTION	redit_amount	4h 13min	135.8 KB			
ılî	202fields with stagge bias, weight field: 0000	CREATE FUSION	outh	9h 6min	17.9 MB			
ılî	502fields with stagge bias, weight field: 0000	VIEW DETAILS	outh	13h 30min	17.9 MB			
ılî	498fields with specta bias, weight field: 0000	MOVE TO	>outh	13h 34min	17.8 MB			
ılî	<b>498fields</b> bias, weight field: 0001e	-8	youth	18h 7min	17.8 MB			
ılî	US car accidents linea	-	Fatalities in crash Fata	1d 4h	771.2 KB			

Figure 3.111: Stop linear regression creation from pop up menu

A modal window will be displayed asking you for confirmation. If you stop the linear regression during its creation you won't be able to resume the same task. If you want to create the same linear regression, you will have to start a new task.

Source	es Datasets	Supervised - Unsupervised - Predictions - Tasks		WhizzN	∕IL ▼
		Are you sure you want to delete this linear regression?	3 8		0,
ill -		, , ,	-		
		If you delete this linear regression, you will no longer have access to its data and you will need to recreate it.			
		Cancel Delete			
		ter as weight field v1 youth 9h 11min 17.9 MB			

Figure 3.112: Confirmation message to delete a linear regression

# 3.13 Deleting Linear Regressions

You can delete your linear regressions by clickin in the DELETE LINEAR REGRESSION option from the **1-click action menu** (see Figure 3.113) or using the **pop up menu** on the linear regression list (see Figure 3.114).

Sources Datasets	Supervised Unsupervised	<ul> <li>Predictions -</li> </ul>	lasks	N	/hizzML 👻
	••• Predicting	g abalone age from r	neasurements	ф <u></u> .	(≡)∕, (i)
4,177	DEFAULT NUMERIC NOT DEFINED	OBJECTIVE CLASS_NUMBER	PREDICT BATCH PREDICTION		ODINGS ES
Bias and predictors	RIABLES: Filter bias or predictors	†↓ Туре	CREATE FUSION CREATE FUSION CREATE FUSION CREATE FUSION CREATE FUSION CREATE FUSION		<b>ev</b>
Bias		123	3.90015		Δσ
Sex = M		ABC	0.05522		Δσ
Sex = I		ABC	-0.82452		Δσ
Sex = F		() ABC	0		

Figure 3.113: Delete linear regression from 1-click action menu

Source	es Datasets <mark>S</mark>	upervised - Unsupervised -	Predictions - Tasks	;			WhizzN	AL <del>-</del>
<u></u>		2	Linear Regressions			8	8 🛍	Q,
ill	Name		Objective	t, 🛗 t,	ê ţ	<b>\$</b>	s (	00
ılî	<b>1001fields</b> bias	Ø	youth	15min	35.6 MB			
ılî	Loan_risk_data bias	PREDICT	redit_amount	4h 24min	135.8 KB			
ılî	Loan_risk_data bias	CREATE BATCH PREDICTION	redit_amount	4h 26min	135.8 KB			
ılî	202fields with stagge bias, weight field: 0000		outh	9h 20min	17.9 MB			
ılî	502fields with stagge bias, weight field: 0000		outh	13h 44min	17.9 MB			
ılî	498fields with specta bias, weight field: 0000		outh	13h 47min	17.8 MB			
ılî	<b>498fields</b> bias, weight field: 0001e	e8	youth	18h 20min	17.8 MB			

Figure 3.114: Delete linear regression from pop up menu

A modal window will be displayed asking you for confirmation. Once a linear regression is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

Source	s Datasets	Supervised  Unsupervised  Predictions  Tasks		Whizzl	ML -
		Are you sure you want to delete this linear regression?			0
ill			<b>\$</b>		
		If you delete this linear regression, you will no longer have access to its data and you will need to recreate it.			
		Cancel Delete			

Figure 3.115: Confirmation message to delete a linear regression

# 3.14 Takeaways

This chapter explains linear regressions in detail. Here is a list of key points:

- A linear regression is a supervised Machine Learning algorithm used to solve regression problems.
- A linear regression is built from a dataset available in BigML and used to make an evaluation, a single prediction, or a batch prediction.
- You can create a linear regression with just one click or configure it as you wish. BigML provides several **configuration** options before creating your linear regression.
- To create a linear regression you need a dataset containing at least one numeric field.
- · Categorical fields must be converted to numeric values in order to train a linear regression model.
- · If you do not specify any objective field, BigML will take the last numeric field in your dataset.
- BigML allows you to include your numeric fields' missing values as valid values to train your linear regression model.
- The **chart view** provides a visual way to analyze a field impact on predictions given certain values for the rest of the fields.
- You get a prediction interval along with the predicted value.
- BigML displays all your linear regression coefficients in a **table view** which you can also download as a CSV file.
- You need to evaluate your linear regression model's performance using data that the model has not seen before.
- The ultimate goal in building a linear regression is being able to make predictions with it.
- BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the linear regression, so you can easily set the input data and get an immediate response.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the linear regression you want to use to make predictions and a dataset containing the instances for which you want to obtain predictions.
- You can configure your batch predictions output file settings.
- · You can download your linear regression to perform local predictions.
- You can add **descriptive information** to your linear regressions (name, description, tags, and category).
- You can move your linear regressions between projects.
- You can **share** your linear regressions with other people using the secret link.
- You can **stop** your linear regression creation by deleting them.
- You can permanently **delete** an existing linear regression.



# **Logistic Regressions**

# 4.1 Introduction

There are multiple Machine Learning problems that require predicting a categorical value, such as "true or false", "churn or not churn", "fraud or not fraud", "high risk, low risk or medium risk", etc. These are called classification problems, and there can be multiple categories (or classes) to predict.

Logistic regression is a supervised Machine Learning technique that can be used to solve classification problems. These problems can also be solved with other Machine Learning methods, such as models, ensembles or deepnets. We explain these methods in Chapter 1, Chapter 2 and Chapter 5 respectively. The main difference is that logistic regression assumes your input fields can be mapped to predict your objective field following linear patterns. For this reason, logistic regressions work better in those cases for which the problem can be linearly solved.

For each class of the objective field, logistic regression computes a probability modeled as a logistic function value, whose argument is a linear combination of the field values. See Section 4.2 for more details on the logistic regression formula.

This chapter contains comprehensive description of BigML's logistic regressions including how they can be created with 1-click (Section 4.3), all configuration options available (Section 4.4), and the different visualizations provided by BigML (Section 4.5). See Section 4.6 for an explanation of how logistic regressions can be used to make predictions. You can also export your logistic regressions in different formats to make local predictions faster at no cost (Section 4.6). The process to evaluate your logistic regressions' predictive performance in BigML is explained in a different chapter (Chapter 7).

In BigML, the third tab of the main menu on the Dashboard allows you to access all of your available **supervised** models. Select LOGISTIC REGRESSIONS from the drop-down menu (Figure 4.1), you will reach the logistic regression list view.

Sourc	es	Dat	asets Supervised -	Unsupervised -	Predictions 👻 Tasks				WhizzN	/IL <del>-</del>
X		ភា ៩ភា	MODELS ENSEMBLES	<u>j</u> L	ogistic Regressions				Q	Ø,
ill	Nan	*	LINEAR REGRESSIONS		1 Objective	t 🛗 🕅		++	t, 🗈 t,	
ıß	diat L2 n	X	LOGISTIC REGRESSIONS V	sing values	diabetes	4m	25.6 KB	1		
ılî	<b>gori</b> L2 n	Ø	FUSIONS	0%) sing values	label	4m	94.9 MB			1
ılî	graj L2 n		TIME SERIES	(80%) sing values	label	4m	9.0 MB	1	2	1
ıß	lc_c L2 n		OPTIML	sing values	quality	6m 2w	1.1 MB			1
~	diat									

Figure 4.1: Logistic regressions under Supervised tab

The logistic regression list view (Figure 4.2) lists all your available logistic regressions. For each logistic regression, the view shows the link to the **Dataset** used to create it, its **Name**, **Objective** (objective field name), **Age** (time elapsed since it was created), **Size**, and number of **evaluations**, **predictions**, and **batch predictions** that have been created using that logistic regression. The SEARCH menu option in the top right corner of the logistic regression list view allows you to **search** your logistic regressions by name.

Sourc	es Datasets Supervised - Unsupervise	ed 🔻 Pr	edictions -	Tasks				Whizz	ML 🗕
S		Log	istic Regres	sions				Î <	Ø
ill	Name	ţ	Objective	t _i	<b>***</b> 1,	<b>A</b> 1	<b>\$</b>	<b>I</b>	
ılî	diabetes L2 regularized (c=1), bias, auto-scaled, missing values		diabetes		4m	25.6 KB	1		
ılı	gorilla-horse-dir all resnet18   Training (80%) L2 regularized (c=1), bias, auto-scaled, missing values		label		4m	94.9 MB			1
ılî	grape-strawberry-dir resnet18   Training (80%) L2 regularized (c=1), bias, auto-scaled, missing values		label		4m	9.0 MB	1	2	1
ılî	Ic_quality   Training (80%) L2 regularized (c=1), bias, auto-scaled, missing values		quality		6m 2w	1.1 MB			1
ılı	diabetes training set L2 regularized (c=1), bias, auto-scaled, missing values		diabetes		2у	20.4 KB			
Show 1	IO V logistic regressions	1 to 5 of	5 logistic regressio	ns				K < 1	> >

Figure 4.2: Logistic regression list view

By default, when you first create an account at BigML, or every time that you start a new project, your list of logistic regressions will be empty. (See Figure 4.3.)

Sources Datasets Supervised	Unsupervised      ▼ Predictions      ▼ Tasks	WhizzML 🕶
	Logistic Regressions	â < 🕉
Name Name	1 Objective	
	No logistic regressions	
Show 10 C logistic regressions	No logistic regressions	I< < > >I

Figure 4.3: Empty Dashboard logistic regressions view

Finally, in Figure 4.4 you can see the icon used to represent a logistic regression.



Figure 4.4: Logistic regression icon

# 4.2 Understanding Logistic Regressions

As mentioned in the introduction of this chapter, logistic regression is a supervised learning algorithm to solve classification problems. Logistic regression works better in those cases for which the features are roughly linear and the problem can be linearly solved. This is mainly due to the fact that the logistic regression generates linear decision boundaries to separate the objective field classes. You can find a

#### detailed explanation of this behavior in the following blog post¹.

The reason behind this linear behavior can be found in the logistic regression formula which consist of a logistic function whose argument is a linear combination of the input field values. You can see the logistic regression formula below (Figure 4.5), where dependent variable,  $p_i$ , is the **probability** for each of the *i* classes of the objective field, and the independent variables, (X1, X2, ...Xk) represent the *k* variables for the **input fields** in your dataset, which are multiplied by the logistic regression **coefficients**  $(b_{0,i}, b_{1,i}, b_{2,i}, ... b_{k,i})$ .

where

$$p_i = \frac{1}{1 + e^{-f_i(X)}}$$

$$f_i(X) = b_{0,i} + b_{1,i}X_1 + b_{2,i}X_2 \dots + b_{k,i}X_k)$$

Figure 4.5: Logistic regression formula

The logistic regression tries to learn the *k* **coefficients**  $(b_{0,i}, b_{1,i}, b_{2,i}, ..., b_{k,i})$  of the linear function,  $f_i(X)$ , using maximum likelihood estimation techniques. BigML logistic regression is an optimized implementation of the liblinear library² which uses the Trust-Region Newton Optimization method to estimate the coefficients.

Each class of the objective field will have a different set of coefficients associated, e.g., if the objective field has two classes, two different functions,  $p_1$  and  $p_2$ , will be learned from the training data, one by class (see Figure 4.6).

$$p_1 = \frac{1}{1 + e^{-(b_{0,1} + b_{1,1}X_1 + b_{2,1}X_2 \dots + b_{k,1}X_k)}}$$
$$p_2 = \frac{1}{1 + e^{-(b_{0,2} + b_{1,2}X_1 + b_{2,2}X_2 \dots + b_{k,2}X_k)}}$$



A positive coefficient ( $b_k > 0$ ) for a field, indicates a positive correlation with the predicted class, while **negative coefficients** ( $b_k < 0$ ) indicate a negative relationship. Higher absolute coefficient values for a field results in a greater **impact on predictions** of that field. This should not be **misinterpreted** as field importance due to several reasons:

- Field importance in Logistic Regression can be defined as the contribution of a field to the final class probability which depends not only on the field coefficient but also on the interactions with the rest of the input fields. Since the model assumes independence between the different inputs, coefficients can be considered as absolute measures of field importance only when all inputs are independent, but this is often not the case. In many real datasets, the impact of a particular field is also dependent on the values of other fields.
- Different field magnitudes make coefficients incomparable. Coefficients for fields with different
  magnitudes, e.g., salary and age, are not comparable since they will tend to be higher for fields
  with smaller scales. Changing the scale of the field would change the coefficient value. BigML
  automatically scales your numeric fields. (See Subsection 4.4.8.)
- Fields can be **multi-collinear**. If two fields are highly correlated, they can be effectively substituted for each other during model training, so the field importance could be split between the two. The greater number of fields in the dataset, the more likely the fields are multi-collinear.

BigML provides a table containing all your logistic regression coefficients. (See Subsection 4.5.2.)

¹https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees ²https://www.csie.ntu.edu.tw/~cjlin/liblinear/

When the logistic regression has learned the coefficients, you can use the model to make **predictions** for new instances. The logistic regression always returns a probability per class of the objective field. The class with the highest probability will be the predicted class. Taking into account the previous formulas in Figure 4.6, for a given set of input values, (X1, X2, ...Xk), you will get two **probabilities**, one per class, e.g.,  $p_1 = 85\%$  and  $p_2 = 15\%$ . In BigML, when there are more than two classes, the probabilities are normalized so the sum of all probabilities for each instance prediction is equal to 100%.

By definition, the **input fields** (X1, X2, ...Xk) in the logistic regression formula need to be **numeric** values. However, BigML logistic regressions can support any type of fields by applying a set of transformations to categorical, text, and items fields. Moreover, BigML can also handle missing values for any type of field. The following subsections detail both behaviors.

# 4.2.1 Input Field Transformations

Apart from numeric fields, BigML logistic regressions are optimized to support categorical, text and items fields by applying a set of transformations in order to convert them in numeric values:

• **Categorical fields** are **One-hot** encoded by default, i.e., each class is mapped to a separate 0-1 numeric variable. For a given instance, the variable corresponding to the instance class, has its value set to 1, while the other variables are set to 0.

For example, imagine you are trying to predict the probability of customer churn = [True, False] given two input fields: number of calls (numeric), numCalls, and the tariff plan (categorical), tariffPlan = [B, N, P], which includes three different classes, B (basic), N (normal), P (professional). The logistic regression will create one variable for the numeric field and another three variables for the categorical field, one by class. Letting i be the objective field classes (True, False), the logistic regression formula will be:

$$p_i = \frac{1}{1 + e^{-f_i(X)}}$$

where

$$f_i(X) = b_{0,i} + b_{1,i} numCalls + b_{2,i}B + b_{3,i}N + b_{4,i}P$$

For a new customer with values numCalls = 240 and tariffPlan = N then:

$$f_i(X) = b_{0,i} + b_{1,i}240 + b_{2,i}0 + b_{3,i}1 + b_{4,i}0$$

BigML also provides three other types of coding, **Dummy**, **Contrast** and **Other** coding, that you can configure for each of your categorical fields. See Subsection 4.4.10 for a complete explanation of categorical fields encoding.

- For text fields, each term is mapped to a corresponding numeric variable, whose value is the number of occurrences of that term in the instance. Text fields without term analysis enabled are excluded from the model (read the Sources with the BigML Dashboard document to learn more about text analysis [11]).
- For **Items fields**, each different item is mapped to a corresponding numeric field, whose value is the number of occurrences of that item in the instance.

### 4.2.2 Missing Values

BigML logistic regressions can handle missing values for any type of field. For categorical, text, and items fields, missing values are always included as another category, term or item by default.

For numeric fields, missing values are also included by default, but you can deactivate this option by configuring your logistic regression (see <u>Subsection 4.4.5</u>). Alternatively, you can replace your missing numeric values by another valid value like the field's mean, median, maximum, minimum or zero (see

Subsection 4.4.4). If none of the mentioned options has been enabled for building your logistic regression, the instances containing missing values for numeric fields in your dataset will be ignored by the model.

For missing values, a separate variable is created to build the logistic regression. Once the logistic regression is created, you can find an additional coefficient for each field at the end of the coefficient table. (See Figure 4.7.) Learn more about the coefficient table in Subsection 4.5.2.

If the dataset does not contain missing values for a field, the coefficient for missing values will be zero, except in the case of text fields which can be different from zero. This is due to the fact that BigML has a limit of 1,000 terms for text fields, so there may be instances not containing any of the terms considered to build the model and appear as missing values instead. (See Subsection 4.8.0.1 to know more about term limits for text fields.)

Sources Datasets	Logistic Regressions 🔻	Clusters Anomalies	Associations Predictions Tas	sks Scripts ▼
	<b>₩</b> - •••• <u>/</u> Us	car accidents dataset's	logistic regression	
INSTANCES           4,999	DEFAULT NUMERIC NOT DEFINED		JECTIVE FIELD INTERCEPT SC NJURY SEVERITY 0	ALE NORMALIZE
		Filter bias or predictors		GY
Bias and predictors	Type	Fatal Injury (K)	Incapacitating Injury (/	A) 🗘 Injured, Severity Unkno
Crash Date.day-of-week	missing	Missing values	0	0
Fatalities in crash	missing	0	0	0
Roadway	missing	0	0	0
Age	missing	0.03463	-1.05278	0.17137
Alcohol Results	missing	-0.85629	-0.18594	-0.52391
Person Type	missing	0	0	0
Drug Involvement	missing	0	0	0
Race	missing	-8.66498	3.0044	0.27518
Gender	missing	0	0	0
Crash Date.month	missing	0	0	0
Show 10 ᅌ variables		76 to 85 of 87 variab	oles	<b>  ζ 5 6 7 8 9 &gt; &gt;  </b>

Figure 4.7: Missing numeric coefficients at the end of logistic regression table

### 4.2.3 Logistic Regressions with Images

BigML logistic regressions do not take images as input directly, however, they can use image features as those fields are numeric.

BigML extracts image features at the source level. Image features are sets of numeric fields for each image. They can capture parts or patterns of an image, such as edges, colors and textures. For information about the image features, please refer to section Image Analysis of the Sources with the BigML Dashboard³[11].

³https://static.bigml.com/pdf/BigML_Sources.pdf

Sources Datase	ts Supervised <del>▼</del> U	nsupervised - Predictio	ons 🕶 Tasks		WhizzML 🔻
a ¥. 0.	••••	grape-straw	berry resnet18	ู่ เมื่อ-	*** ⊘** (≡)** ()
<b>X</b>	4 label	ABC × V		Search by name	•
Name	↑, Туре	t̂j Count t̂j	Missing 🕆 E	Frrors 1, Histogram	
image_id	ima	<b>ge</b> 70	0	o 👋 齸	<b>\$</b> a <b>\$</b>
filename	pat	<b>h</b> 70	0	0	
label	512 image featur		0	0	
Show 10 🗸 fields	✔1 514 🛛 🔄 512	1 to 3 of 3	3 fields		

Figure 4.8: A dataset with images and image features

As shown in Figure 4.8, the example dataset has an image field *image_id*. It also has image features extracted from the images referenced by *image_id*. Image feature fields are hidden by default to reduce clutter. To show them, click on the icon "Click to show image features", which is next to the "Search by name" box. In Figure 4.9, the example dataset has 512 image feature fields, extracted by a pre-trained CNN, *ResNet-18*.

Sources Datasets	Supervised - Ur	supervised - Predictio	ons 🕶 Tasks		WhizzML 🔻
	•••	grape-strawl	Derry res	ide image features	\$
	\rm label	ABC × V		Search by nam	e 🗶
Name	†↓ Туре	t̂↓ Count t̂↓	Missing 🕆 Err	ors t _i Histogram	
image_id	imag	e 70	0	o 🜌 🌖	🐲 🍥 🌔
ilename	patl	70	0	0	
abel	ABO	70	0	0	
mage_id.resnet18 feature 1	<b>≋</b> petra	ned 70	0	° <b>Hululu</b>	adduraa a
mage_id.resnet18 feature 2	petra	ned 70	0	1 A A	hthemason
mage_id.resnet18 feature 3	petra	ned 70	0	° adadh	u.IIII
mage_id.resnet18 feature 4	petra	ned 70	0	° uluuu	unalituaa.
mage_id.resnet18 feature 5	petra	ned 70	0	° Intellinte	
mage_id.resnet18 feature 6	<b>≋</b> petra	ned 70	0	-	literen en tek
image_id.resnet18 feature 7	<b>≋</b> petra	ned 70	0		III
Show 10 🗸 fields	✓! 514 ⊠≣ 512	1 to 10 of 5	15 fields		<pre> &lt; 1 2 3 4 5 &gt; &gt;</pre>

Figure 4.9: A dataset with image feature fields shown

From image datasets like this, logistic regressions can be created and configured using the steps described in the following sections. All other operations including prediction, evaluation applies too.

# 4.3 Creating Logistic Regressions with 1-Click

To create a logistic regression in BigML you have two options: either the 1-click option which uses the default values for all available configuration options, or you can tune the parameters in advanced by using the configuration options explained in Section 4.4. This section explains how to create a logistic regression with 1-click.

From the dataset view, select the 1-CLICK LOGISTIC REGRESSION option in the 1-click action menu. (See Figure 4.10.)

Sources Datasets	Logistic Regressions -	Clust	ers Ano	malies Asso	ociations Predictions Tasks	Scripts -
a s ^r 67. ••	••	0	Loan	risk data data	aset 🍫 👶	=)/~ (
, i ^{ge}					1-CLICK MODEL	3
••••	Type	٥	Count	Missing	1-CLICK ENSEMBLE	
hecking_status	ABC		1,000	0	1-CLICK CLUSTER	
duration	123		1,000	0	1-CLICK ANOMALY	
					1-CLICK ASSOCIATION	
credit_history	ABC		1,000	0	1-CLICK TRAINING   TEST	
					REQUEST EXPORT (CSV)	

Figure 4.10: Create 1-click logistic regression from dataset 1-click action menu

Alternatively, you can use the 1-CLICK LOGISTIC REGRESSION option in the **pop up** menu from the dataset list view. (See Figure 4.11.)

Sourc	Datasets	Logistic Regressions 🔻	Clusters /	Anomalies	Associations	s Prec	dictions	Tasks		Scri	ots 🔻
				Datas	sets						Q
9	Name		\$	₩ ≎	₿ ≎	<u>sh</u> •	£ស 🕈	∑ ≎	<b>%</b>	*****	₽ ≎
54	Loan risk data data	aset		1h 56min	135.8 KB	0	0	1	0	0	0
54	Country St			5h 2min	12.0 KB						
54		1-CLICK ENSEMBLE		5h 2min	75.3 KB			1			
54	Arrythmia			5h 2min	536.2 KB			2			
54	oo carace	1-CLICK ANOMALY		5h 2min	741.0 KB			1			
S°.		1-CLICK ASSOCIATION 1-CLICK TRAINING   TEST		4m	27.0 KB			1	1		
Show	10 ᅌ data 📭	VIEW DETAILS		1 to 6 of 6 datas	sets					K < 1	> >1
		DELETE DATASET									

Figure 4.11: Create 1-click logistic regression from dataset popup menu

Either option builds a logistic regression using the default values for all available configuration options. (See Section 4.4.)

Note: for some datasets, the 1-click option may be disabled. This can be due to the fact that your dataset does not contain any categorical field or the field taken as the default objective field is not categorical. If you do not specify any objective field BigML takes the last valid field in your dataset as the objective field by default. To change the objective field, either configure your logistic regression or select the objective field from your dataset (both options are explained in Subsection 4.4.1.)

# 4.4 Logistic Regression Configuration Options

While the 1-click creation menu option (see Section 4.3) provides a convenient and easy way to create a BigML logistic regression, you can also have more control over the logistic regression creation and configure a number of parameters that affect the way BigML creates logistic regressions. Click the CONFIGURE LOGISTIC REGRESSION menu option in the **configuration menu** of your dataset view. (See Figure 4.12.)



Figure 4.12: Configure logistic regression

Note: for some datasets, the configuration option may be disabled. This can only be due to the fact that the dataset does not contain any categorical field to be selected as the objective field. In case the field to be predicted in your dataset is a numeric field, you can discretize it so it becomes categorical. Read the Datasets documentation [9] to learn how to discretize a numeric field.

# 4.4.1 Objective Field

The objective field, or "target field", is the field you want to predict. Logistic regressions only support **categorical** fields as the objective field.

BigML takes the **last categorical field** in your dataset as the objective field by default. If you want to change the objective field, you have two options: you can select another field from the configuration panel to build the logistic regression, or you can change it permanently from your dataset view.

• Select the **Objective field** from the logistic regression **configuration panel**. This option will only affect the logistic regression you are building that time. (See Figure 4.13.)

Sources Datasets Log	gistic Regressions 🔻	Clusters Anor	malies Associat		
≗   S ²   S ¹⁰ - • • • •		US car a	accidents datas	set	🍫 🕗 (≡)🕶 🤅
Objective field:		Default n	umeric value: የ	Missing numerics: 🔞	Eps: 🕜
Injury Severity	ABC	Select a	default value	N/A	0.0001
State	AB	c			
Atmospheric Condition	AB				$\overline{\mathbf{O}}$
Roadway Person Type	AB				
	AB	C			
Drug Involvement	AB				
Drug Involvement Race	AB	c c		Reset	+ Create logistic regression
Drug Involvement	AB				Create logistic regression
Drug Involvement Race Gender Injury Severity	AB AB AB		Missing		۹ :
Drug Involvement Race Gender Injury Severity Name	AB AB AB		Missing	Errors Histogram	۹ :
Drug Involvement Race Gender	AB AB AB AB AB	c Count		Errors Histogram	Q 1

Figure 4.13: Configure the objective field to create the logistic regression

• Change the **default objective field** for the dataset. This option will save your objective field preference for any model you build. Click on the edition icon next to the field name when you mouse over it, a pop up window will be displayed. Then click on the Objective field icon and Save it. (See Figure 4.14.)

Sources Datasets Logistic F	Regressions ▼ Clusters Anomalies As	sociations Pred	lictions Tasks Scripts ▼
	Update field 000001 details	aset	°⊗ - ∽ (≡) ^{×™} ()
Name	Name	Errors	Q X
State	Atmospheric Condition	0	
Atmospheric Condition	•	0	
Crash Date	Description	0	
Fatalities in crash		0	
Roadway	This field is the objective field	0	dial and a
Age	✓ Save X Cancel	0	nal HUUUUunaa.
Alcohol Results	<b>123</b> 2,293 2,706	0	h
Person Type	<b>ABC</b> 4,999 0	0	

Figure 4.14: Change the default objective field

# 4.4.2 Automatic Optimization

You can turn on the **Automatic optimization** option so BigML will automatically tune the parameters of your logistic regression (see Figure 4.15).

a 17, 60,	•	Sent	iment	🍫 - 🦽 - (≡	) <b>%</b>
OGISTIC REGRESSION CO	NFIGURATION			(	2
bjective field: sentiment	ABC	Automatic optimiza	Max. training time: 00:30:00 ©	Logistic regression candidates:	•
Default numeric value: Select a default value	Missing numerics:	Eps:	No stats:		
	ion				⊽

Figure 4.15: Automatic optimization

The high number of possible combinations for parameter values makes it difficult to find the optimum configuration since the combinations that lead to a poor result outnumber the ones that result in a satisfying performance. Hand-tuning different configurations is a time-consuming process that requires a high level of expertise and intuition. To combat this problem, BigML offers first-class support for automatic logistic regression parameter optimization.

Behind the scenes, BigML uses the same technology for logistic regression parameter optimization as the one used for OptiML. If you want to know more about the technical details, please read the Chapter 2 of the document OptiML with the BigML Dashboard [10].

When you turn on the **Automatic optimization** option, all the logistic regression parameters will be disabled (because they will be automatically optimized), except the **Default numeric value**, the **Missing numerics**, and the **Weights** parameters which you can manually configure (see Figure 4.16).

Sources Datasets Supervised - Un:	supervised   Predictions	Tasks	WhizzML 🗸
≙ Ţ, <mark>8</mark> 0,	Sentimer	nt	°‰_ ♂~ (≡) [*] ~ ③
LOGISTIC REGRESSION CONFIGURATION			? pui
Objective field: sentiment ABC v	Automatic optimization	Max. training time: 00:30:00 ©	128
Default numeric value: Missing numerics: Select a default value   N/A	Eps:	No stats:	
Advanced configuration			$\odot$
Logistic regression name:			
Sentiment		R	eset // Create logistic regression

Figure 4.16: Configure the default numeric value and the missing numerics

Since the optimization process can take some time, BigML offers two configurable parameters to limit the time to create the optimized logistic regression: a training duration (see Subsection 4.4.2.1) and the logistic regression candidates (see Subsection 4.4.2.2).

#### 4.4.2.1 Training duration

The scale parameter to regulate the logistic regression runtime. It's set as an integer from 1 to 10. It indicates the user preference for the amount of time they wish the optimization to take. The higher the number, the more time that users are willing to wait for possibly better logistic regression performance. The lower the number, the faster that users wish the logistic regression training to finish. The default value is set to 5.

The training duration is set in a scale. The actual training time depends on the dataset size, among other factors.

Sources Datasets Supervised - Un:	supervised 👻 Predictions 🕤	<ul> <li>Tasks</li> </ul>	WhizzML 🔫
	diabete	s	ılề₂ 🎭 ♂₂ (≡)⁵₂ 🛈
LOGISTIC REGRESSION CONFIGURATION			POT
Objective field:	Automatic optimization	Training duration:	Logistic regression candidates:
Default numeric value:     Missing numerics:       Select a default value <ul> <li>N/A</li> </ul>	Eps: 0.	No stats:	Ø
Advanced configuration			$\odot$
Logistic regression name: diabetes		Reset	Create logistic regression

Figure 4.17: Training duration

#### 4.4.2.2 Logistic regression candidates

The maximum number of different logistic regressions (i.e., logistic regressions using a unique configuration) to be trained and evaluated during the optimization process. The default number is 128 candidates which is usually enough to find the best logistic regression, but you can set it from 4 up to 200. Only the top-performing logistic regression will be returned. If training duration is very low (see Subsection 4.4.2.1) given the dataset size, it is possible that not all the logistic regression candidates will be tried out.

Sources Datasets	Supervised - Unsup	pervised - Prediction	ns 🕶 Tasks	WhizzML 🔻
a 17. 07	•	diabo	etes	ılề₊ 🎭 🍊₊ (≡)⁵₊ 🛈
OGISTIC REGRESSION COM	FIGURATION			
Dbjective field: diabetes	C V	utomatic optimization	Training duration:	5 Logistic regression candidates:
Default numeric value:       Select a default value	Missing numerics:	Eps:	0.0001 O	
Advanced configurati	on			•
ogistic regression name:				
diabetes			Re	set 🥂 Create logistic regression

Figure 4.18: Logistic regression candidates

# 4.4.3 Weights

It is not unusual for a dataset to have some categories that are common and others very rare. For example, in datasets used to predict fraud, usually fraudulent transactions are very scarce compared to regular ones. When this happens, logistic regression tends to predict the most frequent values simply because the overall model's performance metrics improve with that approach. However, in cases such as fraud prediction, you may be more interested in predicting rare values rather than successfully predicting frequent ones. In that case, you may want to assign more **weights** to the scarce instances so they are equivalent to the abundant ones.

BigML provides three different options to assign specific **weights** to your instances, **balance objective**, **objective weights**, and **weight field** explained in the following sections (see Figure 4.19).

Sources Datasets Supervised - I	Jnsupervised   Predictions	Tasks			WhizzML 🔻
a 17, 80,	US car ac	cidents		🍖 👶	(≡)∕, (i)
OGISTIC REGRESSION CONFIGURATION					<b>?</b>
Dbjective field: Injury Severity ABC •	Default numeric value:           Select a default value         *	Missing numerics:	Eps:	0.0001	No stats:
Advanced configuration					۲
Weights:					۲
Select a weight option				WEIGHTS No weights	
Sc Objective weights					◙
Regularization:					຺
Field codings:					຺
Sampling:			4	1,999 instances	$\overline{\mathbf{v}}$

Figure 4.19: Weight options for logistic regression

#### 4.4.3.1 Balance Objective

When you set the **balance objective** weight (see Figure 2.21), BigML automatically balances the classes of the objective field by assigning a higher weight to the less frequent classes, with the most frequent class always having a weight of 1. For example, take the following frequencies for each class:

[False, 2000; True, 50]

By enabling the **Balance objective** option, BigML will automatically apply the following weights:

[False, 1; True, 40]

In this example, the class "True" is getting forty times more weight as it is forty times less frequent than the most abundant class.

#### 4.4.3.2 Objective Weights

The **Objective weights** option allows you to manually set a specific weight for each class of the objective field. BigML oversamples your weighted instances replicating them as many times as the weight stablishes. If you do not list a class, it is assumed to have a weight of 1. Weights of 0 are also valid, but if all classes have a weight of 0, the logistic regression creation will produce an error.

This option can be combined with the **Weight field** (see Subsection 4.4.3.3). When combining it with the **Weight field**, both weights are multiplied. For example if you assign a weight of 3 for the "True" class and the weight field assigns a weight of 2 for a given instance labeled as "True", that instance will have a total weight of 6.

#### 4.4.3.3 Weight Field

The **Weight field** option allows you to assign individual weights to each instance by choosing a special weight field. The selected field must be numeric and it must not contain any negative or missing values. The weight field will be excluded from the input fields when building the logistic regression. You can select an existing field in your dataset or you may create a new one in order to assign customized weights.

The weights of your weight field will impact in the same way as the **Objective weights**. If an instance has a weight of 3 it will be replicated three times in the dataset to train the model.

### 4.4.4 Default Numeric Value

Logistic regressions can include missing values as valid values for any type of fields as explained in Subsection 4.2.2. However, there can be situations for which you don't want to include them in your model. For those cases, the **Default numeric value** parameter is an easy way to replace missing numeric values by another valid value. You can select to replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 4.20.)



Figure 4.20: Select a default numeric value to replace missing numeric values

Note: if your dataset does not contain missing values for your numeric fields, this parameter will not have impact on your logistic regression. By contrast, if your dataset contains missing numeric values and you neither select a Default numeric value or enable the Missing numerics configuration option, instances with missing numeric values will be ignored to build the model. (See next Subsection 4.4.5.)

### 4.4.5 Missing Numerics

By default, missing values for your numeric fields are included as valid values to build your logistic regression. However, as explained in the previous subsection, there can be cases for which you don't want them to be included in syour model. The **Missing numerics** option allows you to select if you want to include or exclude the missing numeric values to build your logistic regression. (See Figure 4.21.)

	gistic Regressio	ns - Clusters	Anomalies Associ	iations Predictio	ons Tasks	Scripts 👻
<b>₽ ₽ ₽</b> •••••		111	US car accidents data	aset	🍖 👶-	(≡)∕~ (i)
Objective field:	•	ABC V	Default numeric value: 3 Select a default value	Missing nume	erics: 🖓 Eps: 🖗	0.0001
්සූ Configure						$\overline{\mathbf{O}}$
Logistic regression name:						
US car accidents dataset's lo	ogistic regress	ion				
	ogiatic regresa	ion		Reset	Z [*] Create logistic	regression
	ogiatic regress			Reset	Create logistic	regression ×
	≎ Typ		punt Missing	Errors		
Name		e ≎ Co	punt Missing		٩	×
Name State Atmospheric Condition		e ≎ Co ABC 4,9		Errors	Q. Histogram	×

Figure 4.21: Include missing numeric values in your logistic regression

#### Note: missing values are always included for categorical, text, and items fields. (See Section 4.2.)

As explained in Subsection 4.2.2, by including missing numeric values, you will get an additional coefficient for your numeric fields denoting the missing values. You will find them at the end of the **coefficients table** as you can see in Figure 4.22. If a numeric field has no missing values, those coefficients will be zero.

Sources Datasets Lo	gistic Regressions 👻	Clusters Anomalies As	ssociations Predictions Tas	ks Scripts ▼
	•••• <u>)</u> US c	ar accidents dataset's lo	gistic regression	🌐 💠 🍊 🛈
4,999	NOT DEFINED		CTIVE FIELD INTERCEPT SC	ALE NORMALIZE FALSE
		Filter bias or predictors		GY
Bias and predictors	\$ Туре	Fatal Injury (K)	Incapacitating Injury (A)	) 🗘 Injured, Severity Unkno
Crash Date.day-of-week	missing	Missing values	0	0
Fatalities in crash	missing	0	0	0
Roadway	missing	0	0	0
Age	missing	0.03463	-1.05278	0.17137
Alcohol Results	missing	-0.85629	-0.18594	-0.52391
Person Type	missing	0	0	0
Drug Involvement	missing	0	0	0
Race	missing	-8.66498	3.0044	0.27518
Gender	missing	0	0	0
Crash Date.month	missing	0	0	0
Show 10 文 variables		76 to 85 of 87 variables		<b>  &lt; &lt;</b> 5 6 7 8 9 > >

Figure 4.22: Missing numeric coefficients at the end of logistic regression table

**Missing numerics** and **Default numeric value** parameters are substitutes. (See Subsection 4.4.4.) Therefore, if you enable the default numeric value parameter, the **Missing numerics** icon will automatically be disabled. (See Figure 4.23.)

bjective field: Injury Severity ABC	Default numeric value:     No missing numerics:     Eps:       Maximum     N/A	0.0001
Configure		۲
Scales:		$\overline{\mathbf{O}}$
Regularization:		◙
Sampling:	4,999 instances	$\overline{\mathbf{O}}$
Advanced sampling:	Custom settings	$\overline{\mathbf{O}}$

Figure 4.23: Missing numerics is disabled when there is a Default numeric value selected

Note: if your dataset contains missing numeric values and you do not either enable Missing numerics or select a Default numeric value, instances containing missing values will be ignored when building the logistic regression.

#### 4.4.6 Eps

This parameter sets the **stopping criteria** for the solver. If the difference between the current results and the last iteration results is smaller than **Eps**, then the solver is finished. You can set positive float values greater than 0 and smaller than 1. The default value is 0.0001. Higher values make the model to be faster built but they may result in a worse predictive performance.

Sources Datasets Lo	gistic Regressions	s ▼ Clusters Anor	nalies Associati	ons Predictions		
a   S ^q   8₀, ••••		US car a	accidents datas	et	<b>%</b>	(≡)/- (≡)/- (
Dbjective field:		Default n	umeric value: 🔞	Missing numeri	cs: 🗘 Ep	s: 😧
Injury Severity	A	BC Select a d	default value	N/A	G	
්සූ Configure						•
ogistic regression name:						
ogistic regression name: US car accidents dataset's	ogistic regressio	n		Reset	Ž ⁺ Create	e logistic regression
	logistic regressio	n		Reset	Z [*] Create	e logistic regression
US car accidents dataset's	logistic regressio ≎ Type	n \$ Count	Missing			
US car accidents dataset's	≎ Туре		Missing	Errors H	٩	
US car accidents dataset's l	↓ Type	≎ Count	Ū.	Errors H	listogram	

Figure 4.24: Eps parameter for logistic regression

# 4.4.7 Stats

This parameter allows you to include or exclude the **statistical tests** to assess the quality of the model's fit to the data. (See Figure 4.25.) BigML does not include these statistics by default to speed up the model creation because the **time required** to compute them can be considerably **high**.

Sources Datasets	Supervised - I	Jnsupervised 🔻	Predictions	<ul> <li>Tasks</li> </ul>	_	_	WhizzML 🔻
â ŢŢ, ₽₽, ••	••	US	Car Accident	ts dataset fi	Itered		‱, ⊘, (≡) <b>%</b> (i
LOGISTIC REGRESSION CC	ONFIGURATION						<b>?</b>
Dbjective field: Injury Severity	ABC v	Default num Select a def		Missing nu	umerics:	Eps:	No stats:
ogistic regression name:							
	et filtered's logistic	regression			Reset	<u></u> c	reate logistic regression
US Car Accidents datase	et filtered's logistic	regression			Reset	<u>/</u> c	reate logistic regression
US Car Accidents datase	et filtered's logistic		Count	Missing	Reset		
US Car Accidents datase	≎ Тур		Count 4,999	Missing		Q Histogram	

Figure 4.25: Include statistics in logisticregression

When **stats** are enabled, BigML automatically configures your categorical fields with **dummy coding**. BigML selects the dummy class in alphabetical order, therefore for fields with classes such us "0"/"1", "yes"/"no" or "true/"false", the dummy class will always be "0", "no" or "false". You can see the field codings configuration by clicking in the wheel next to your categorical fields. (See Figure 4.26.) This automatic configuration is to avoid multicollinearity in your model, otherwise all the statistics computed will be null. You can configure the categorical fields with any of the other field codings, **contrast** or **other**, but you cannot configure one-hot coding again if **stats** are enabled. See Subsection 4.4.10.1 to read more about field codings.

Sources Datasets	Supervised 👻 l	NEW Unsupervised ▼	Predictions	▼ Tasks				WhizzML 🔻
	•		US car acci	dents datas	et	<b>%</b>	<b>.</b> .	(≡)⁵- 〔
LOGISTIC REGRESSION CO	NFIGURATION							•
Objective field: Injury Severity	ABC ¥	Select a defa		Missing ne	umerics:	Eps:	0.0001	Stats:
Advanced configuration	ion							▣
Logistic regression name: US car accidents dataset'	s logistic regress	ion			Reset	Crea	ate logistic	regression
Name	\$ Тур	e 🗘	Count	Missing	Errors	Histogram		
State	!	ield coding config Dummy	ured as	0	0			l
Atmospheric Condition	۲		4,999	0	0			_
Crash Date	!	Y-MM-DD	4,999	0	0			
Fatalities in crash		123	4,999	0	0			
Roadway	-	ABC 🍖	4,999	0	0	dud	alı.	_
Age		123	4,929	70	0			

Figure 4.26: Check your categorical fields encoding by clicking in the green wheel icon

BigML computes two kinds of statistical tests, a test that measures the quality of the coefficients as a whole, and a set of tests that measure the significance of the individual coefficient estimates:

- Statistical test to measure the quality of the whole model:
  - Likelihood ratio: tests whether the coefficients as a whole have any predictive power to
    predict a certain class of the objective field over an intercept-only model.
- Statistical tests to measure the significance of the individual coefficient estimates:
  - Standard error: the variance of the coefficient estimates.
  - Z scores: how many standard deviations are the coefficient estimates from the mean. Letting β be the coefficient estimate and SE the standard error, then the formula for each i coefficient is:

$$\mathsf{Z} \operatorname{score}_i = \frac{\beta_i}{SE(\beta_i)}$$

- *p*-value: determines the statistical significance of each coefficient in the logistic regression formula. The *p*-value is associated with the  $\chi^2$  of the Wald test⁴ with one degree of freedom at the  $Z^2$  value. The Wald test indicates how far the estimated parameters are from zero to test their significance.

The *p*-value needs to be compared against a significance level, usually 0.1, 0.05 or 0.01. A *p*-value higher than the significance level, indicates that the null hypothesis can be accepted, hence the coefficient is non-significant. A *p*-value lower than the significance level indicates

⁴https://en.wikipedia.org/wiki/Wald_test

that the null hypothesis can be rejected, so the coefficient estimate is significant. A good practice is to retrain the logistic regression removing the non-significant coefficients.

Note: note that *p*-values are not extracted directly from Z score values, they are associated with the  $\chi^2$  of the Wald test.

- Confidence intervals: give the size of the 95% confidence interval for each coefficient estimate. That is, for a coefficient estimate  $\beta$ , and an interval value n, the value of the coefficient is  $\beta \pm n$  with a confidence of 95%.

Once your logistic regression has been created, you will be able to visualize the **stats** in the coefficient table. (See Subsection 4.5.2.1.)

To avoid lengthy computation times, stats from large input datasets will be computed from a subsample of the dataset such that the number of coefficients X rows is less than or equal to 1E+8.

It is possible that some statistic values contain **null values**. Wald test statistics can not be computed for zero-value coefficients, and so their corresponding entries are null. Moreover, if the coefficients' information matrix is **ill-conditioned**, e.g. if there are fewer instances of the positive class than the number of coefficients, or if there is a perfect correlation between the input fields (multicollinearity), the standard error, Z score, *p*-value, and confidence intervals will also have a null value.

# 4.4.8 Bias and Auto-Scaling

You can include or exclude the **Bias** from the model, a.k.a. the intercept term of the logistic regression formula. (See formula in Section 4.2.) For most cases, including the bias results in a better model. By default it is included. (See Figure 4.27.)

You can also scale the **numeric fields** of your dataset, to ensure each field will have equivalent influence despite differences in magnitudes, e.g., salary and age. By enabling the **Auto-scale** parameter, BigML automatically transforms your numeric fields so their standard deviations equal 1. Auto-scaling fields will allow you to compare the different coefficients learned by the model as explained in Section 4.2. Fields are auto-scaled by default. (See Figure 4.27.)



Figure 4.27: Bias parameter and auto-scaling parameter for numeric fields

### 4.4.9 Regularization

The main goal for having a regularization factor is to avoid overfitting, i.e., tailoring the model to the training data at the expense of generalization.

In BigML you can choose between **L1 or L2 Regularization**. L1 norm causes more coefficients to be zero, while using the L2 norm forces the values of all coefficients towards zero. Usually L2 yields better results, so it is the default option to create your logistic regression.

You can also tune the **Strength (c)** parameter, which is the inverse of the regularization strength, so higher values indicate less regularization. It must be a positive integer greater than 0. Too high values for strength will make the algorithm perfectly fit the training data boundaries, so the logistic regression will perform poorly when trying to predict new instances not seen before by the model. Too low values for strength will result in vague decision boundaries not following the data patterns, hence resulting also in a bad predictive performance. The default value is 1 which usually works well for most cases. (See Figure 4.28.)

S	US car accidents data	set	🍖 🧔-	(≡)⁄~
jective field: jury Severity	Default numeric value:         •           •         Select a default value         •	Missing numerics: <b>2</b> N/A	Eps: 😧	0.0001
Configure				(
Scales:				$\odot$
Regularization:				۲
Regularization: Strength (c):		REGULARIZATION	STRENGTH (C)	0
Sa L1			4,999 instances	$\overline{\mathbf{O}}$
			Default settings	◙
Advanced sampling:				

Figure 4.28: Regularization parameters

# 4.4.10 Field Codings

**Categorical fields** must be converted to numeric values in order to train a logistic regression model. By default, they are **One-hot** encoded, but BigML allows you to configure three other types of coding for each one of your categorical fields: **Dummy coding**, **Contrast coding**, and **Other coding**. See the following subsections for a detail explanation of each option. (Learn more about input fields transformations in Subsection 4.2.1.)

Note: if stats are enabled, then the default coding is Dummy for all your categorical fields, selecting the dummy class by alphabetical order. See <u>Subsection 4.4.7</u> for more details.

#### 4.4.10.1 One-hot Coding

Categorical fields are **One-hot** encoded by default. That is, a separate 0-1 numeric field is created for each category and an additional one for missing values. For a given instance, the variable corresponding to the instance's categorical value has its value set to 1, while the other variables are set to 0. See an example of **One-hot** coding scheme for a field containing three classes in Table 4.1:

Classes	C0	C1	C2	C3
Class 1	1	0	0	0
Class 2	0	1	0	0
Class 3	0	0	1	0
MISSING	0	0	0	1

Table 4.1: One-hot coding

#### 4.4.10.2 Dummy Coding

The main goal of using dummy coding⁵ is to compare a class selected as the reference or **control class** with the rest of classes. The control class is assigned a value of 0 for each variable. The control class is called **dummy class** in BigML and it is usually a class with a representative number of instances compared to the other classes in the dataset. See an example of dummy coding schema for three different classes, being the "Class 1" the dummy class, in Table 4.2:

Classes	C0	C1	C2
Class 1	0	0	0
Class 2	1	0	0
Class 3	0	1	0
MISSING	0	0	1

Table 4.2: Dummy coding example for 3 classes

To set **Dummy coding** for a field:

1. Click on the configuration icon next to the field name. (See Figure 4.29.)

 $^{{}^{5} \}tt https://en.wikipedia.org/wiki/Categorical_variable \#Dummy_coding$ 

Sources Datasets	Models ▼ Clusters	Anomalies As	sociations Prec	dictions Tas	iks	Scripts 🔻
a s ^a 8	•	Loan r	isk data datase	et	<b>%</b> -	(=) ^{NEW} /→ ()
Objective field:		Default n	umeric value: 👔	Missing nu	merics: O Eps:	0.0001
class	ABC	Select a d	default value	N/A	$\Theta$	0.0001
ିଲ୍ଲ Configure						$\odot$
ogistic regression name:						
Loan risk data dataset's lo	gistic regression			Reset	Ž ⁺ Create lo	ogistic regression
<b>•</b>					٩	×
Name Fie	ld codings	Count	Missing	Errors	Histogram	
checking_status	🇞 🗹 🗛	1,000	0	0		
duration	123	1,000	0	0	www	
credit_history	item	s 1,000	0	0		
ourpose	АВ С	1,000	0	0		
credit_amount	123	1,000	0	0		
savings_status	AB C	1,000	0	0		
employment	See ABC	1,000	0	0		
nstallment_commitment	123	1,000	0	0		

Figure 4.29: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field has not a previous configuration for field codings, it will be disable. **Enable** field coding configuration by clicking on the green switcher shown in Figure 4.30.

a s 19. •	Configure field coding for purpose field	×	
Dbjective field:	DISABLED		ps:
class		_	
	Dummy coding:		
Configure	Class	Select	
ogistic regression name:	business	۲	
Loan risk data dataset's	domestic appliance	0	
<b>X</b>	education	0	3
lame	furniture/equipment	0	
hecking_status	new car	0	
uration	othor	^	
	Configured coding: NONE	Cancel Save	
redit_history			

Figure 4.30: Enable field coding configuration

3. Select the class you want to set as the dummy class. (See Figure 4.31.)

	lodels ▼ Clusters Anomalies Assoc Configure field coding for <b>purpose</b> field	iations Predictions Tasks	Scripts -
8   S ⁴   ⁶ 7-   •			· ↓ (=) → ①
Objective field:	DISABLED CONTRACTOR CONTRACTOR		ps: 🕜
class	Dummy coding:	and the second	9
Configure	Class	Select	$\overline{\mathbf{O}}$
_ogistic regression name:	business	•	
Loan risk data dataset's	domestic appliance	Select this class as dummy	te logistic regression
<b>X</b>	education	Ŏ	8
lame	furniture/equipment	0	
hecking_status	new car	0	
luration	.*b.~v	^	
redit_history	Configured coding: NONE	Cancel Save	
purpose	ABC 1,000		

Figure 4.31: Select the dummy class

4. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **DUMMY**". (See Figure 4.32.)

Sources Datasets M	odels <del>•</del> Clusters Anomalies A	ssociations Predictions Tasks	Scripts +
	Configure field coding for <b>purpose</b> field	×	(=) ^{NEW} ()
Objective field:	DISABLED COM ENABLED		ps: 0
class	Dumn	ny coding selected	0.0001
	Dummy coding:		
Configure	Class	Select	$\overline{\mathbf{v}}$
Logistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	•	22
Name	furniture/equipment	0	
checking_status	new car	0	
duration	othor	~	
credit_history	Configured coding: DUMMY	Cancel Save	
purpose	<b>ABC</b> 1,000	0 0	

Figure 4.32: Field codings: dummy

Note: you cannot select several field codings for the same field simultaneously.

5. Close the modal window by clicking outside or by clicking Cancel .

Sources Datasets	Models - Clusters Anomalies Asso		Scripts -
a s [.] 6	Configure field coding for purpose field	×	r_ (≣)/~ ()
Objective field:	disabled Correction Enabled		ips: 🕜
class	Contrast c	oding selected	0.0001
	Contrast coding:		
Configure	Class	Value	$\overline{\mathbf{v}}$
Logistic regression name:	business	-0.25	
Loan risk data dataset's	domestic appliance	1	te logistic regression
	education	0.5	
Name	furniture/equipment	-0.25	26
checking_status	new car	-1	
	athar	•	
duration	Configured coding: CONTRAST	Cancel	
credit_history	gg		

Figure 4.33: Close modal window

Note: if the Cancel button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

6. After configuring the field codings for a field, the configuration icon will become green. (See Figure 4.34.)

Sources Datasets	Models -	Clusters A	nomalies As	sociations Pr	edictions Ta	sks	_	Scripts 👻
a s ^a 67.	••••		Loan r	isk data datas	et		🍖 🚓	(≡) [∕] ▼ (i)
Objective field:			Default n	umeric value: የ	Missing nu	merics: 🔞	Eps: 🔞	0.0001
class		ABC	* Select a	default value 🔻	N/A		0	0.0001
Configure								⊚
ogistic regression nar	ne:							
Loan risk data datase	t's logistic regr	ession			Reset	j*	Create logistic	regression
<b>V v</b>						٩		×
Name Fie	ld codings configu	ired ^{(pe}	Count	Missing	Errors	Histogram		
hecking_status	🗞 🗹	BC	1,000	0	0			
duration		123	1,000	0	0	անուն		
credit_history		items	1,000	0	0			
ourpose	ŝ	ABC	1,000	0	0		_	
credit_amount		123	1,000	0	0	III		
avings_status	ф.	ABC	1,000	0	0			
employment	°	ABC	1,000	0	0		_	

Figure 4.34: Field codings configured

7. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 4.35.)

) s [.] 7. •			· · · · · · · · · · · · · · · · · · ·
ojective field:	DISABLED CON ENABLED		ps: 🔞
lass	Dummy coo	ding selected	0.0001
	Dummy coding:		
Configure	Class	Select	$\bigcirc$
gistic regression name:	business	0	
oan risk data dataset's	domestic appliance	0	te logistic regression
2	education	٥	
me	furniture/equipment	0	
ecking_status	new car	0	
ration	other	^	
	Configured coding: DUMMY	Cancel	
edit_history			

Figure 4.35: Disable field coding configuration

After creating your logistic regression, your dummy class will be identified with the dummy icon in the **coefficients table** view (see Subsection 4.5.2). (See Figure 4.36.)

Sources Datasets Logistic Regressions	Clusters Anomalies	Associations Prediction	ns Tasks	Scripts 🗸
a s ^a ir <mark>8</mark> /	Loan risk data dataset's l	logistic regression	🔋 💠 🛷	(≡) [∕] ~ (€)
1,000 DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	BJECTIVE FIELD INTER	O NO	FALSE
Bias and predictors	Type	bad	good	<
credit_history = credits/all	items	0.11584	-0.11584	
credit_history = no	items	0.11584	-0.11584	
purpose = radio/tv	ABC	-0.25654	0.25654	
purpose = new car	ABC	0.60878	-0.60878	
purpose = furniture/equipment	ABC	-0.14651	0.14651	
purpose = used car	Control Class (B C (Dummy Coding)	-0.93013	0.93013	
purpose = business	АВС	0	0	
purpose = education	АВС	0.63471	-0.63471	
purpose = repairs	АВС	0.36551	-0.36551	
purpose = domestic appliance	АВС	0.02578	-0.02578	
Show 10 🗘 variables	11 to 20 of 79 varia	bles	K K 1 2	3 4 5 > >

Figure 4.36: Dummy class in table view

#### 4.4.10.3 Contrast Coding

Contrast coding⁶ allows you to set different values for different classes. Instead of the 0-1 values of **One-hot** coding, you will be able to set any integer or float value for each of the classes, plus an additional one for missing values. The sum of all values must equal 0. The values of the classes need to be set based on certain hypothesis, e.g., higher values for a class assume this class has more influence on the objective field than the others. A positive value indicates a positive relationship between the class and the objective field while a negative value indicates a negative relationship. A coefficient of 0 will exclude the class from the model. In the Table 4.3 you can see an example of contrast coding schema for three different classes.

Classes	C0
Class 1	0.5
Class 2	-0.25
Class 3	-0.25
MISSING	0

Table 4.3: Contrast coding example for 3 classes

To set **Contrast coding** for a field, follow these steps:

⁶https://en.wikipedia.org/wiki/Categorical_variable#Contrast_coding

1. Click on the configuration icon next to the field name. (See Figure 4.37.)

Sources Datasets	Models 🕶	Clusters A	nomalies A	ssociations Pr	edictions Ta	asks		Scripts 🔻
a s ^a 9. •	•••		Loan	risk data datas	et		🎭 🦽	(≡) [/] ▼ (i
Dbjective field:			Default	numeric value: 🔞		numerics: 👔	Eps: 🕜	0.0001
class		ABC	* Select a	default value v	N/A		0	
රීසූ Configure								$\odot$
ogistic regression name	:							
Loan risk data dataset's	s logistic regre	ession				et 🧷	Create logistic	regression
<b>•</b>						Q		э
lame	Field codings	Туре	Count	Missing	Errors	Histogram	_	
hecking_status	° <u>o</u> €	ABC	1,000	0	0			
luration		123	1,000	0	0	uludi.		
redit_history		items	1,000	0	0			_
ourpose	°0	ABC	1,000	0	0			
redit_amount		123	1,000	0	0	llu		-
avings_status	¢œ	ABC	1,000	0	0			
mployment	¢œ	ABC	1,000	0	0			
nstallment_commitment		123	1,000	0	0			

Figure 4.37: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field has not a previous configuration for field codings, it will be disable. **Enable** field coding configuration by clicking on the green switcher shown in Figure 4.38

	odels <del>–</del> Clusters Anomalies Asso		Scripts 👻
a s' 7. •	Configure field coding for purpose field	×	·
Objective field:	DISABLED ENABLED		.ps: 0
class			0.0001
	Dummy coding:		
Configure	Class	Select	$\overline{\mathbf{O}}$
Logistic regression name:	business	۲	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	0	×
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	^	
	Configured coding: NONE	Cancel Save	
credit_history			

Figure 4.38: Enable field coding configuration

3. Select the Contrast coding option. (See Figure 4.39.)

Sources Datasets	vlodels <del>–</del> Clusters Anomalies Associat		Scripts 🔻
a s° 17	Configure field coding for purpose field	Contrast	< : _ (5- (=)  ()
Objective field:	DISABLED ENABLED		.ps: 0
class			2
	Dummy coding:		
්සූ Configure	Class	Select	
ogistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
<b>V</b>	education	0	
lame	furniture/equipment	0	
hecking_status	new car	0	
luration	athar	^	
	Configured coding: NONE	Cancel Save	
credit_history			

Figure 4.39: Field codings: contrast coding

4. Set the values you want for your classes based on your hypothesis. All classes values must sum 0. (See Figure 4.40.) By using the BigML API, multiple contrast codings can be given for a field as long as all the codings are orthogonal to ensure there are no co-dependent coefficients. Check the corresponding documentation⁷.

 $^{^{7}} h \texttt{ttps://bigml.com/api/logistic$  $regressions \texttt{#lr_coding_categorical_fields}$ 

	odels • Clusters Anomalies As Configure field coding for purpose field	isociations Predictions lasks	Scripts -
pjective field:	DISABLED CON ENABLED		:▲ (∅+ (=)- ps: 0
lass			0.0001
	Contrast coding:	and the second	
Configure	Class	Value	۲
gistic regression name:	business	-0.25	
pan risk data dataset's	domestic appliance	1	te logistic regression
	education	0.5	
<u>a</u> *	furniture/equipment	-0.25	3
ne cking_status	new car	-1	
uning_status	other	0	_
ation			
dit_history	Configured coding: NONE	Cancel	

Figure 4.40: Set the contrast coding values for each class

Note: you cannot select several field codings for the same field simultaneously.

5. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **CONTRAST**". (See Figure 4.41.)

	odels  Clusters Anomalies Assoc	ciations Predictions Tasks	Scripts -
Image: Comparison of the section o	DISABLED CONTRACTOR ENABLED		r_
class	Contrast co	oding selected	0.0001
	Contrast coding:		
Sonfigure	Class	Value	$\overline{\mathbf{O}}$
ogistic regression name:	business	-0.25	
Loan risk data dataset's	domestic appliance	1	te logistic regression
<b>V</b>	education	0.5	×
lame	furniture/equipment	-0.25	~
hecking_status	new car	-1	
	athar	0	
uration	Configured coding: CONTRAST	Cancel Save	
predit_history	gg		

Figure 4.41: Contrast coding saved

6. Close the modal window by clicking outside or by clicking Cancel .

Sources Datasets	Models <del>▼</del> Clusters Anomalies As	sociations Predictions Tasks	Scripts -
a s 19	Configure field coding for purpose field	:	× (=) ··· (=)
Objective field:	DISABLED COM ENABLED		.ps: 🕜
class	Dummy	coding selected	0.0001
	Dummy coding:		
Configure	Class	Select	$\overline{\mathbf{v}}$
Logistic regression name:	business	0	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	•	×
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	^	
credit_history	Configured coding: DUMMY	Cancel	
purpose			

Figure 4.42: Close modal window

Note: if the Cancel button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

7. After configuring the field codings for a field, the configuration icon will become green. (See Figure 4.43.)

Sources Datasets	Models 🔻	Clusters Ar	nomalies As	sociations Pre	dictions Tasl	ks		Scripts 👻
	••••		Loan r	isk data datas	et		<b>‰</b> . ⊘.	(≡) [/] √
Objective field:			Default n	umeric value: 👔	Missing nur	nerics: 🕜	Eps: 🕜	0.0001
class		ABC	* Select a d	lefault value 🔻	N/A		0	
Configure								◙
Logistic regression name	e:							
Loan risk data dataset	s logistic regre	ssion			Reset	<u></u> c	reate logistic r	egression
<b>•</b>						٩		×
Name	l codings configure	ed ^{ype}	Count	Missing	Errors	Histogram		
checking_status	🍇 🗹	BC	1,000	0	0			
duration		123	1,000	0	0	www.	ll	
credit_history		items	1,000	0	0			
purpose	00 00	ABC	1,000	0	0			
credit_amount		123	1,000	0	0	llu		
savings_status	ogo	ABC	1,000	0	0			
employment	^o a	ABC	1,000	0	0			

Figure 4.43: Field codings configured
8. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 4.44.)

	odels  Clusters Anomalies As Configure field coding for purpose field	sociations Predictions Tasks	
	DISABLED C ENABLED		
Objective field:	Contras	at coding selected	:ps: 🕜
	Contrast coding:		
Configure	Class	Value	$\overline{\mathbf{O}}$
ogistic regression name:	business	-0.25	
Loan risk data dataset's	domestic appliance	1	te logistic regression
<b>V</b>	education	0.5	×
ame	furniture/equipment	-0.25	
hecking_status	new car	-1	
uration	other	0	
redit_history	Configured coding: CONTRAST	Cancel Save	1111

Figure 4.44: Disable field coding configuration

After creating your logistic regression, you will be able to see your **Contrast coding** values in the **coefficients table** view (see Subsection 4.5.2) by clicking on the icon. (See Figure 4.45.)



Figure 4.45: Contrast icon in table view

A modal window will be displayed with your codings values and you can download them in CSV or JSON format by clicking on the corresponding icons. (See Figure 4.45.)

	Checking_status	EXP	× س	DEVELOPERS Dashboard
	Class		Value	👩   PROD 🥽 DEV
Sources Datasets	no checking		-0.5	Scripts *
a s ^a ir f	<0		1	> 3, (≡)/- 0
	0<=X<200		-0.25	NORMAL 17E
1,000	>=200		-0.25	FALSE
	[MISSING]		0	
Bias and predictors		≎ Type ≎ D	aa ≎ g	¢ boo

Figure 4.46: Contrast modal window in table view

### 4.4.10.4 Other Coding

Other coding⁸ allows you to set different values for different classes. It works the same way as contrast coding (see Subsection 4.4.10.3), but in this case the values do not need to sum 0. In the Table 4.4 you can see an example of other coding schema for three different classes.

Classes	C0
Class 1	2
Class 2	-0.4
Class 3	3
MISSING	1

Table 4.4: Other coding

To set **Other coding** for a field, follow these steps:

1. Click on the configuration icon next to the field name. (See Figure 4.47.)

 $^{^{8} \\ \}texttt{https://en.wikipedia.org/wiki/Categorical_variable\#Contrast_coding}$ 

Sources Datasets	Models - Cluster	s Anomalies A	ssociations Pre	dictions Ta	isks	Scripts 🔻
a s ^a 67	••	Loan	risk data datas	et	<b>%</b> _ (	≫_ (=)∕- î
Objective field:		Default	numeric value: 🔞	Missing nu	umerics: 😧 Eps: 🕄	0.0001
class	AB	Select a	a default value 🔻	N/A	$\odot$	0.0001
Configure						$\odot$
Logistic regression name:						
Loan risk data dataset's l	ogistic regression				Ż [†] Create logi	stic regression
<b>•</b> •					٩	×
Name	eld codings	Count	Missing	Errors	Histogram	_
checking_status	🍇 🗹 🛛 🗛	C 1,000	0	0		
duration	12	1,000	0	0	u IIII kana	
credit_history	ite	nis 1,000	0	0		
ourpose	🎭 🔼 🗚	c 1,000	0	0		_
credit_amount	12	3 1,000	0	0	<b>.</b>	
savings_status	AB	c 1,000	0	0		
employment		C 1,000	0	0		_
nstallment_commitment	12	3 1,000	0	0		

Figure 4.47: Field coding configuration

2. A modal window will be displayed so you can configure the field codings for that field. If the field has not a previous configuration for field codings, it will be disable. **Enable** field coding configuration by clicking on the green switcher shown in Figure 4.48

	odels – Ciusters Anomalies As:		Scripts 👻
a s' 17. •	Configure field coding for purpose field	×	·
Objective field:	DISABLED		ps: 🔞
class		_	0.0001
	Dummy coding:		
්සූ Configure	Class	Select	$\overline{\mathbf{a}}$
Logistic regression name:	business	۲	
Loan risk data dataset's	domestic appliance	0	te logistic regression
	education	0	22
Name	furniture/equipment	0	
checking_status	new car	0	
duration	athar	^	
	Configured coding: NONE	Cancel Save	
credit_history			

Figure 4.48: Enable field coding configuration

3. Select the Other coding option. (See Figure 4.39.)

- <b>-</b>	Configure field coding for purpose field	Other	· ▲ (=)-
jective field:			ips: 🕜
lass	Dummy coding:	and the second	0.000
Configure	Class	Select	
	business	۹	
gistic regression name:	domestic appliance	0	
oan risk data dataset's	education	0	te logistic regressio
3 -	education	0	
ne	furniture/equipment	0	
cking status	new car	0	
oning_otatao	other	~	_
ation	Configured coding: NONE	Cancel Save	
dit_history			

Figure 4.49: Field codings: other coding

4. Set the values you want for your classes based on your hypothesis. You can set any float or integer value. (See Figure 4.50.) By using the BigML API, multiple other codings can be given for a field. Check the corresponding documentation⁹.

 $^{^{9} \}tt https://bigml.com/api/logistic$  $regressions {\tt \#lr_coding_categorical_fields}$ 

	Adels  Clusters Anomalies Assoc Configure field coding for purpose field	iations Predictions Tasks X	Scripts *
Objective field:	DISABLED ENABLED		: ▲ (5- (=)/~ (3)
class	Other coding:	and the second s	0.0001
Configure	Class	Value	$\odot$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	
Name	furniture/equipment	0	*
checking_status	new car	-2	
	athar	0	
duration	Configured coding: NONE	Cancel	d1111
credit_history	Computed Coding. NONE	Calicer	
purpose	ABC 1,000	0 0	L.,

Figure 4.50: Set the other coding values for each class

Note: you cannot select several field codings for the same field simultaneously.

5. Click Save . Make sure you saved your configuration by looking at the bottom message "Configured Coding: **OTHER**". (See Figure 4.51.)

	odels  Clusters Anomalies Assoc Configure field coding for purpose field	iations Predictions Tasks	Scripts +
	disabled 🛛 😸 ENABLED		:_ (∋)' <del>-</del> ()
Objective field:			.ps: 🕜
class	Other cod	ing selected	0
	Other coding:	<b>_</b>	
Configure	Class	Value	$\overline{\mathbf{v}}$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	×
Name	furniture/equipment	0	
checking_status	new car	-2	
duration	athas	0	
duration	Configured coding: OTHER	Cancel	
credit_history			

Figure 4.51: Other coding saved

6. Close the modal window by clicking outside or by clicking Cancel.

Sources Datasets	Models - Clusters Anomalies Ass		Scripts -
a s 7	Configure field coding for purpose field		× (=)/- ①
Objective field:	DISABLED CON ENABLED		.ps: 🔞
class	Other	coding selected	0.0001
	Other coding:		
🎭 Configure	Class	Value	$\odot$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	2
Name	furniture/equipment	0	
checking_status	new car	-2	
duration	athar		
	Configured coding: OTHER	Cancel	
credit_history			

Figure 4.52: Close modal window

Note: if the Cancel button is red, it indicates there are changes you have not saved yet so you will lose them by closing the modal window.

7. After configuring the field codings for a field, the configuration icon will become green. (See Figure 4.53.)

Sources Datasets	Models 🔻	Clusters Ar	iomalies As	sociations Pre	dictions Tas	ks		Scripts 👻
a s ^a 67-	••••		Loan r	isk data datase	et		🍖. G.	(≡) [∕] ~ (i)
Objective field:			Default n	umeric value: 🔞	Missing nu	merics: 🕜	Eps: 😰	0.0001
class		ABC	* Select a c	efault value	N/A		0	
Configure								⊚
Logistic regression nam	ne:							
Loan risk data datase	t's logistic regre	ssion			Reset	<u>_</u> * (	Create logistic r	egression
<b>V</b>						Q		x
Name	d codings configur	ed ^{ype}	Count	Missing	Errors	Histogram		
checking_status	🍇 🗹	BC	1,000	0	0			
duration		123	1,000	0	0	ահուր		
credit_history		items	1,000	0	0			
purpose	°0	ABC	1,000	0	0		_	
credit_amount		123	1,000	0	0	llu		
savings_status	<b>\$</b> \$	ABC	1,000	0	0			
employment	ŝœ	ABC	1,000	0	0			

Figure 4.53: Field codings configured

8. To remove the field coding configuration for that field, click Disable from the switcher and click Save again. (See Figure 4.54.)

	odels  Clusters Anomalies Asso Configure field coding for purpose field	ciations Predictions Tasks X	
a s [.] 7. •			·_ ∅- (=) ⁼
Objective field:			ips: 🕜
class	Otherco	ding selected	
	Other coding:	<b>_</b> _	
Configure	Class	Value	$\odot$
Logistic regression name:	business	1	
Loan risk data dataset's	domestic appliance	3	te logistic regression
	education	0.4	×
Name	furniture/equipment	0	
checking_status	new car	-2	
duration	athar	n	
	Configured coding: OTHER	Cancel Save	
credit_history			

Figure 4.54: Disable field coding configuration

After creating your logistic regression, you will be able to see your **Other coding** values in the **coefficients table** view (see Subsection 4.5.2) by clicking on the icon. (See Figure 4.55.)

Sources Datasets	Logistic Regressions  Clusters Anomalies Associations Pre		Predictions Tasks		S	Scripts ▼			
a s ^r ir	₩. ···· <u>/</u>	Loan risk data datas	et's logis	tic regres	ssion	Ê	¢	<b>ॐ</b> ~ (≡)∕~	•
INSTANCES	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS		TIVE FIELD		OEPT SCAL	E	NORMALIZE FALSE	
		BLES: Filter bias or predi	ctors						CSV V
Bias and predictors	6		\$	bad		\$	good		
Bias		View Classes Values 23 (Other Coding)		0			0		
checking_status				-0.1729			0.1729		
duration		123		0.0302			-0.0302		
credit_history = existing		items		-0.59306			0.59306		

Figure 4.55: Other coding in coefficients table

A modal window will be displayed with your coding values and you can download them in CSV or JSON format by clicking on the corresponding icons. (See Figure 4.55.)

	PLOYMENTS GALLERY T LAE	EXP 3S	TERESA WI	HAT'S NEW DEVELOPERS Dashboard
	Checking_status		CSV	PROD COM DEV
Sources Datasets	Class		Value	Scripts -
	no checking		2	
	<0		1	
INSTANCES	0<=X<200		0.4	FALSE
	>=200		-3	
	[MISSING]		0	CSV
Bias and predictors				1 ÷
Bias		123	0	0
checking_status	O ^r			

Figure 4.56: Other coding modal window

## 4.4.11 Sampling Options

Sometimes you do not need all the data contained in your dataset to build your logistic regression. If you have a very large dataset, sampling may be a good way of getting faster results. BigML allows you to sample your dataset before creating the logistic regression, so you do not need to create a separate dataset first. You can find a detailed explanation of the sampling parameters available in the following subsections. (See Figure 4.57.)

### 4.4.11.1 Rate

The **Rate** is the proportion of instances to include in your sample. Set any value between 0% and 100%. Defaults to 100%.

### 4.4.11.2 Range

Specifies a subset of instances from which to sample, e.g., choose from instance 1 until 200. The **Rate** you set will be computed over the **Range** configured. This option may be useful when you have temporal data, and you want to train your logistic regression with historical data, and test it with the most recent one to check if it can predict based on time.

## 4.4.11.3 Sampling

By default, BigML selects your instances for the sample by using a random number generator, which means two samples from the same dataset will likely be different even when using the same rates and row ranges. If you choose deterministic sampling, the random-number generator will always use the same seed, thus producing repeatable results. This lets you work with identical samples from the same dataset.

### 4.4.11.4 Replacement

Sampling with replacement allows a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once. By default, BigML generates samples without replacement.

### 4.4.11.5 Out of bag

This argument will create a sample containing only out-of-bag instances for the currently defined rate, so the final total number of instances for your sample will be one minus the rate configured for your sample (when replacement is false). This can be useful for splitting a dataset into training and testing subsets. It is only selectable when a sample rate is less than 100%.

	Sources Datasets Logisi	ic Regressions 👻 C	lusters Anoma	lies Associations	Predictions Ta	isks	Scripts 👻
	≜ S ^q ₩,		Loan ris	k data dataset		<b>%</b> G-	(≡) [/] √
	Objective field:	ABC	Pefault num     Select a def		Nissing numerics: 🕢	Eps: 🕑	0.0001
	Configure						۲
	Scales:						$\overline{\mathbf{O}}$
	Regularization:						$\odot$
Rate	Sampling:					1,000 instances	۲
	Rate:	100%				SAMPLING RATE	0
Range	Advanced sampling:					Custom settings	۲
	Range: 1,000 instances	1,000	RANGE 1 - 1,000	SAMPLING Random	REPLACEMENT YES	OUT OF BAG	0
Sampling-	. 🗃 🏛 🛍						
eplacement Out of bag							

Figure 4.57: Sampling parameters for logistic regression

### 4.4.12 Creating Logistic Regressions with Configured Options

After finishing the configuration of your options, you can change the default logistic regression name in the editable text box. Then you can click on the Create logistic regression button to create the new logistic regression, or reset the configuration by clicking on the Reset button.

ective field:	Automatic optimizatio	n		
	•			
ault numeric value: Missing nur	nerics: Eps:	No stats:		
Advanced configuration				
Weights:				⊽
Scales:				$\overline{\mathbf{O}}$
Regularization:				⊽
Field codings:				$\overline{\mathbf{O}}$
Sampling:			42,535 instances	$\overline{\mathbf{O}}$
Dataset advanced sampling:			Default settings	$\overline{\mathbf{O}}$
Advanced ordering:			Default settings	♥
istic regression name:			~	
an data logistic regression		Reset	Create logistic r	egressio

Figure 4.58: Create logistic regression after configuration

## 4.4.13 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to the

Reset button (See (Figure 4.58)). This is to show how to create the logistic regression programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 4.59) below:

				0
bjective field:		Automatic optimization		
efault numeric v			×	
Select a default val	API reques API endpoint:	st preview /logisticregression		
	1 - { 2	"objective_field": "000010".	_	
Advanced o	3	"name": "Loan data logistic regression", "missing_numerics": true.		
Weights:	5 6	<pre>"compute_stats": false, "eps": 0.0001,</pre>		
Scales:	7 8 9	"balance_fields": true, "bias": true, "regularization": "l2",		
Regularizati	10 11	"c": 1, "optimize": false,		
Field coding		Check the full list of argument	nts in the API documentation	
Sampling:			42,535 instances	
Advanced ord				

Figure 4.59: Logistic regression API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for logistic regressions, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

# 4.5 Visualizing Logistic Regressions

After creating your logistic regression, you will be able to analyze your results with BigML unique visualization: a **1D and 2D chart**, to examine the impact of the input fileds in the objective field, and a **coefficients table**, for more advanced users, to interpret the resulting coefficients for each input field. The following subsections explain both visualizations in detail.

Switch from the chart view to the table view by clicking on icons in the top bar menu of the logistic regression view. (See Figure 4.60.)

a s ^e il ^e	°∎• •••• <u>∕</u> L	oanRisks dataset.	's logistic regression	1	• ૐ- (≡)√- 0
INSTANCES	DEFAULT NUMERIC N	ISSING NUMERICS	OBJECTIVE FIELD	BIAS	FIELD CODINGS
Table	EFINED	TRUE	CLASS	YES	NO

Figure 4.60: Switch chart and table views

## 4.5.1 Logistic Regression Chart

The chart view is composed of three main parts: the CHART itself, which you can view in one dimension (1D) or two dimensions (2D), the PREDICTION legend and the INPUT FIELDS form. (See Figure 4.61.) You can find a detailed explanation of each one below.



Figure 4.61: Logistic regression chart parts

• The CHART allows you to view the impact of the input fields on the objective classes predictions. You can select the 1D chart or 2D chart by clicking on the green switcher in the top bar menu.

The **1D chart**, allows you to select one input field for the x-axis. The y-axis represents the probabilities for each of the predicted classes. Only numeric fields can be selected for the x-axis.(See chart limitations in Subsection 4.8.1.) You can extend the upper limit of the x-axis by clicking on the plus icon.

Probability	v			PREDIC	TION
00% <b>7</b>				bad	Probabili
30%				0%	100
30%				bad	64.06
/0%				good	
60%			0		
50%				All classes	¥
10%				INPUT F	ELDS All fields: 🗹
30%			0	checking_status	<b></b>
				0<=X<200	v
20%				credit_history	S
10%					
0%	30 -	40 50	60 70	+	

Figure 4.62: 1D chart

The **2D chart** allows you to select two different input fields for both axis and the class probabilities are represented in different colors in a heat map. You can select numeric or categorical fields for the axis. You can switch the axis by clicking on the option on top of the chart area.

purpose	*	used car	£3	PREDIC	TION
purpose		0000 001	Cx	good	Probability
used car				50.1%	87.2%
retraining				bad	34.0
repairs				good	65.9
radio/tv					
other				All classes	Ŧ
new car				INPUT FI	ELDS All fields:
furniture/equipment				checking_status	٥
				0<=X<200	v
education					
education domestic appliance				credit_history	٩
	30 40	) 50	60 70	credit_history	٦



In both charts you can inspect the axis values in the grey area next to the selector. You can freeze the view by pressing Shift and release it again by pressing Escape from your keyboard. When

the view is frozen, an edition icon will appear you can edit the axis values and obtain a prediction for another preferred value. The resulting predicted probabilities are in the prediction legend to the right.

To know more about how to interpret the influence of the input fields on predictions, read this post¹⁰.

 The PREDICTION legend allows you to visualize the classes represented in the chart along with their corresponding colors. The main probability color bar at the top is the probability for the predicted class. By default, in the 2D chart, colors are shaded according to the range of probabilites shown in the chart area. That way, smaller differences in predictions are easier to perceive. However, you can choose to see the color shading according to the total range of class probabilities (from 0% to 100%) by clicking on the icon next to the probability bar Total. (See Figure 4.64.) You can also select to see only one of the classes using the class selector int he bottom of the legend.



Figure 4.64: Prediction legend

Freeze this view by pressing Shift, and release it again by pressing Escape from your keyboard.

• Below the chart legend, you can find the INPUT FIELDS form. (See Figure 4.65.) You can configure the values for any numeric, categorical, text or items field. By changing their values, you can see the class probabilities changing in real-time. You can also select or disable your input fields, so they will be treated as missing values. If you configured your model to deactivate the **Missing numerics** option, you will not be able to disable your numeric fields. (See Subsection 4.4.5.)

¹⁰https://blog.bigml.com/2016/09/26/predicting-airbnb-prices-with-logistic-regression/

		22.5000	1		PREDICTION
age	Y	22.5000	* Q	bad	Probability
70				50.1%	75.1%
65				bad	72.89
60 -				good	- 27.11
55 -					
50 .				All classes	Ŧ
45					NPUT FIELDS All fields: 🗹
40				checking_status	<b></b>
35 -				0<=X<200	¥
30 -					
25 .				credit_history	<b></b>

Figure 4.65: Configure the values for other input fields

Moreover, the chart includes a reset option for your input fields values, and an export option to download your chart in PNG format explained below:

• After selecting the fields for the axis or configuring the input fields values, you can set them again to the default view by clicking the **reset** icon highlighted in Figure 4.66.

age	<b>*</b>	22.5000	£3	PREDICTION	
5			bad	Probabili	
0.			50.1%	75.1	%
5			bad		72.
			good	-	27.
55 -					
			All classes	v	
15				INPUT FIELDS All fie	elds:
			checking_status		I
95 -			0<=X<200		
25 -			credit_history		I
0	30 40	50 60	70		

Figure 4.66: Reset the values for the input fields

• You can also export your chart in PNG format with or without legends. Freeze the view by pressing

Shift from your keyboard and export the chart to get the classes percentages in the legend. Release the view by pressing Escape.

age	¥	22.5000	۲. bad	EXPORT PDP AS IMAGE	
75 70 70 70 70 70 70 70 70 70 70 70 70 70			50.1%		75.1%
65			bad	_	72.89
60 -			good	_	27.11
55 -					
50 .			All classes	\$	
45 -				INPUT FIELDS	All fields: 💽
40			checking	g_status	۲
35 -			0<=X<2	200	¥
30 -			credit_hi	istory	
20 -					L
10 20	30 40	50 6	0 70		

Figure 4.67: Export chart as image with or without legends

Note: there are some limitations in the number of classes of the objective field and the number of input fields to visualize your logistic regression in the chart (explained in Subsection 4.8.1).

## 4.5.2 Coefficient Table

The main goal of the logistic regression algorithm is to learn the **coefficients** of the logistic function for each of the dependent variables, i.e., for each of the input fields. A different set of coefficients is associated with each class of the objective field. See Section 4.2 for a detailed explanation of logistic regression coefficients interpretation.

BigML allows you to inspect the learned coefficients for each one of the input fields in the coefficient table. The table **columns** represent the objective field classes while the table **rows** represent the input field variables and the bias (a.k.a. intercept term) of the logistic regression. In the first row you will always find the **Bias** coefficients. If your objective field has a high number of classes, you may need to scroll horizontally to see all of them. You can order the table rows by clicking on any of the columns labels.



Figure 4.68: Table view for logistic regression

For numeric fields, there is always one coefficient by field, however categorical, text and items fields have one coefficient by value (category, term or item). This is due to the required transformations, explained in Section 4.2, to convert categorical, text, and items fields in numeric fields (each single value is mapped to a separate variable in the formula). Missing values also get their own coefficients.

- For **numeric** fields, you always get one coefficient per field. If you train the logistic regression with **Missing numerics**, you will find an additional coefficient per field for the missing values. (See Subsection 4.4.5.)
- For categorical fields, you have one coefficient per classand an additional one for missing values per field. (See Subsection 4.4.10.) Note: if you configure the field with Contrast or Other coding there will be just one coefficient for that field (see Subsection 4.4.10).
- · For text fields, there is one coefficient per term and an additional one for missing values per field.
- For items fields, you get one coefficient per item and an additional one for missing values per field.

See an example of coefficients for a categorical field in Figure 4.69 where one single field, "Atmospheric condition", yields twelve different variables associated with different coefficients, one per each one of the field's classes.

Sources Datasets Logist	tic Regressions - Clusters	s Anomalies Associations	s Predictions Tasks	NEW Scripts ▼
	•••• 🗾 US car acci	dents' dataset's logistic re	egression 📋 💠	ૐ- (≡)∕- 0
	NULT NUMERIC MISSING M	UMERICS OBJECTIVE FIEL		FIELD CODINGS NO
	VARIABLES: Filter	bias or predictors		CSV
Bias and predictors	Type	Seatal Injury (K)	Incapacitating Injury (A)	Injured, Severit
Bias	123	0	0	0
Atmospheric Condition = Clear	ABC	0.32739	-0.7529	-0.85787
Atmospheric Condition = Cloudy	ABC	0.24365	-0.69657	-0.57577
Atmospheric Condition = Snow	ABC	0.21401	-0.82362	0.13515
Atmospheric Condition = Rain	ABC	0.14261	-1.00879	0.19305
Atmospheric Condition = Sleet, Hail (F	reezing Ra AB C	0.15334	0.2617	-0.26642
Atmospheric Condition = Fog, Smog,	Smoke ABC	0.07901	0.08599	-0.22839
Atmospheric Condition = Blowing Sno	ABC	0.0776	-1.37713	-0.11355
Atmospheric Condition = Severe Cros	swinds ABC	-0.01984	0.52796	-0.01327
Atmospheric Condition = Not Reporte	d ABC	0.00427	-0.726	-0.02405
Atmospheric Condition = Unknown	ABC	0.03207	-0.06046	-0.00356
Atmospheric Condition = Other	ABC	0.00516	-0.28787	-0.00481
Atmospheric Condition = Blowing Sar	nd, Soil, Dirt ABC	0.00098	-0.00298	-0.00021

Figure 4.69: Multiple field variables for categorical one-hot encoded fields

Coefficients for **missing values** are always found at the end of the table. (See Figure 4.70.) For fields without missing values in the original dataset, those coefficients should be zero (see Subsection 4.2.2.)

Sources Datasets	Logistic Regressions -	Clusters Anomali		Predictions Tasks	Scripts -
₽ 5° ıı		S car accidents data	iset's logistic regr	ession	i 🏟 🚳 🛈
INSTANCES           4,999	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS	OBJECTIVE FIELD	INTERCEPT SCALE	FALSE
		S: Filter bias or predict	ors		GY
Bias and predictors	Type	Fatal Injury (K	\$	Incapacitating Injury (A)	Injured, Severity Unkn
Crash Date.day-of-week	missing	Missing values		0	0
Fatalities in crash	missing	0		0	0
Roadway	missing	0		0	0
Age	missing	0.03463		-1.05278	0.17137
Alcohol Results	missing	-0.85629		-0.18594	-0.52391
Person Type	missing	0		0	0
Drug Involvement	missing	0		0	0
Race	missing	-8.66498		3.0044	0.27518
Gender	missing	0		0	0
Crash Date.month	missing	0		0	0
Show 10 🔿 variables		76 to 85 of 8	37 variables		K < 5 6 7 8 9 > >

Figure 4.70: Missing numeric coefficients at the end of logistic regression table

Additional options for the table include a filtering option and an export option:

• You can filter the table first column by field name, class, term or item using the **search** box at the top of the table (see Figure 4.71.)

Sources Datasets Logistic Regressions	Clusters Anoma	lies Associations	Predictions Tasks	Scripts 👻
a s in 17 j	JS car accidents dat	aset's logistic regre	ession	
4,999 DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS	OBJECTIVE FIELD	INTERCEPT SCALE 0	NORMALIZE FALSE
Bias and predictors	The bias of predic	tors	<ul> <li>incapacitating Injury (A)</li> </ul>	↓ Injured, Severit
Bias		0	0	0
Atmospheric Condition = Clear	ABC	0.32739	-0.7529	-0.85787
Atmospheric Condition = Cloudy	ABC	0.24365	-0.69657	-0.57577
Atmospheric Condition = Snow	ABC	0.21401	-0.82362	0.13515
Atmospheric Condition = Rain	ABC	0.14261	-1.00879	0.19305
Atmospheric Condition = Sleet, Hail (Freezing Ra	ABC	0.15334	0.2617	-0.26642
Atmospheric Condition = Fog, Smog, Smoke	ABC	0.07901	0.08599	-0.22839
Atmospheric Condition = Blowing Snow	ABC	0.0776	-1.37713	-0.11355
Atmospheric Condition = Severe Crosswinds	ABC	-0.01984	0.52796	-0.01327
Atmospheric Condition = Not Reported	ABC	0.00427	-0.726	-0.02405
Show 10 0 variables	1 to 10 of	e 87 variables	IK K	1 2 3 4 5 > >

Figure 4.71: Search and filter logistic regression table

• You can also **export** the table in a CSV file by clicking on the icon highlighted in Figure 4.72.

Sources Datasets Logistic Regressions	Clusters Anom	alies Associations	Predictions Tasks	Scripts 🗸
a S° in ™ )	US car accidents da	taset's logistic regre	ession	
4,999 DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD	INTERCEPT SCALE 0	NORMALIZE FALSE
VARIAE	Filter bias or predi	ctors		Ţ
Bias and predictors	Type 🗘 F	atal Injury (K)	Incapacitating Injury (A)	Injured, Severity
Bias	123	0	0	0
Atmospheric Condition = Clear	ABC	0.32739	-0.7529	-0.85787
Atmospheric Condition = Cloudy	ABC	0.24365	-0.69657	-0.57577
Atmospheric Condition = Snow	ABC	0.21401	-0.82362	0.13515
Atmospheric Condition = Rain	ABC	0.14261	-1.00879	0.19305
Atmospheric Condition = Sleet, Hail (Freezing Ra	ABC	0.15334	0.2617	-0.26642
Atmospheric Condition = Fog, Smog, Smoke	ABC	0.07901	0.08599	-0.22839
Atmospheric Condition = Blowing Snow	ABC	0.0776	-1.37713	-0.11355
Atmospheric Condition = Severe Crosswinds	ABC	-0.01984	0.52796	-0.01327
Atmospheric Condition = Not Reported	ABC	0.00427	-0.726	-0.02405
Show 10 C variables	1 to 10 o	f 87 variables	IC ( 1	2 3 4 5 > >

Figure 4.72: Export table in CSV file

Note: there are some limitations in the number of classes of the objective field and the number of input fields to visualize your logistic regression in the coefficient table (explained in Subsection 4.8.1).

#### 4.5.2.1 Coefficient Table with Stats Computation

When **stats** are enabled, BigML displays them in the coefficient table. See a detailed explanation of how to include the stats in your model in Subsection 4.4.7.

In the resulting coefficient table you will find three new elements: a new row at the top of the table containing the **likelihood ratio**, an icon per coefficient indicating its **significance** and a **summary of the stats** per coefficient. You can find a detailed explanation below:

• The **likelihood ratio** tests if the coefficients as a whole have any predictive power. It is the difference in the log likelihood between the fitted model and an intercept-only model. You will find it in the first row of the table. (See Figure 4.73.)

Sources Datasets	Supervised - Unsu	NEW) pervised ▼ Prediction	New) ns ▼ Tasks			WhizzML 🔻
a   17.   10.	•••• <u>)</u>	transport dataset -	sample (80.00%)'s lo	gistic	<u>م</u> د	<b>5</b> - (≡) ⁵ - C
INSTANCES 3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS	OBJECTIVE FIELD ED_DISCRETIZED	BIAS YES	FI	ELD CODINGS YES
	VARIABLES: Filter bias or p	oredictors		FICANCE LEVEL:	0.1 *	ন্ট
Bias and predictors		🗢 Туре	12 years	\$	13-18 years	
ikelihood ratio 👔			378.39908	Δσ	378.39908	σ
Bias		123	-0.58822	σ	0.58822	o م
iemale = 1		ABC	-0.19065	σ	0.19065	σ
iemale = 0		() ABC	0		0	
race = white		ABC	0.70347	σ	-0.70347	σ
		ABC			0	
ace = black			0		0	
race = black race = hispanic		ABC	0.09679	σ	-0.09679	σ
				<u></u> σ		<u>α</u> σ
race = hispanic		ABC	0.09679		-0.09679	

Figure 4.73: Likelihood ratio

If this difference is significant, i.e., the *p*-value is lower than the significance level, then the coefficients as a whole have more predictive power than an intercept-only model. The icon shown in Figure 4.74 indicates if the likelihood ratio is **significant** or not given the selected significance level.

Sources Datasets	Supervised - Unsup	NEW) pervised ▼ Prediction	NEW) Is ▼ Tasks			WhizzML 🔻
₽   ₩.   ₩.   •	•••• <u>)</u>	transport dataset -	sample (80.00%)'s lo	ogistic	$\mathbf{\Phi}$	ૐ- (≡)⁵- ⓒ
3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS	OBJECTIVE FIELD ED_DISCRETIZED	BIAS YES		FIELD CODINGS YES
	ARIABLES: Filter bias or p	redictors	SIGN	IIFICANCE LEVEL:	0.1 💌	ন্ট 🗟
Bias and predictors		Type	12 years	Significant	13-18 years	
Likelihood ratio 👔			378.39908	Δσ	378.39908	σ
Bias		123	-0.58822	Δσ	0.58822	o o
female = 1		ABC	-0.19065	σ	0.19065	σ
female = 0		(I) ABC	0		0	
race = white		ABC	0.70347	σ	-0.70347	Δ σ
race = black		(I) ABC	0		0	
race = hispanic		ABC	0.09679	σ	-0.09679	σ
bytest		123	-1.04493	σ	1.04493	σ

Figure 4.74: Significant likelihood ratio

You can select your preferred significance level by using the selector shown in Figure 4.75.

Sources Datasets	Supervised - Unsu	NEW) pervised ▼ Predictions	NEW) s ▼ Tasks	_	WhizzML 🔻
a 17. 80.	•••• <u>)</u>	transport dataset - s	ample (80.00%)'s l	ogistic 🗘	ૐ- (≡)⁵- ③
INSTANCES 3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD ED_DISCRETIZED	BIAS YES	FIELD CODINGS YES
	VARIABLES: Filter bias or p	predictors	SIG	NIFICANCE LEVEL: 0.1 *	ন্দ্র 🖾
Bias and predictors		\$ Туре	12 years	0.01 ars	4
Likelihood ratio 😮			378.39908	0.05 0.1 )8	Δ •
Bias		123	-0.58822	Δ σ 0.58822	Δ •
female = 1		ABC	-0.19065	Δ σ 0.19065	σ
female = 0		() ABC	0	0	
race = white		ABC	0.70347	ο -0.70347	σ
race = black		ABC ABC	0	0	

Figure 4.75: Select significance level

You can see the *p*-value associated to the likelihood ratio by mousing over the sigma icon. (See Figure 4.76.)

Sources Datasets	Supervised - Unsup	ervised - Prediction	<mark>NEW)</mark> is ▼ Tasks			WhizzML 🔻
	•• <u> </u>	ransport dataset -	sample (80.00%)'s log	istic	<b>(</b>	<b>ૐ</b> - (≡) ⁵ - ③
3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD ED_DISCRETIZED	BIAS YES	FI	ELD CODINGS YES
	ABLES: Filter bias or pr	edictors	SIGNIF	ICANCE LEVEL:	.1 🔻	4 <u>T</u> ≎ 💼
Bias and predictors		🗘 Туре	12 years	\$	13-18 years	
Likelihood ratio 👔			378.39 08		78.39908	σ
Bias		123	-0.588; ? <b>p-v</b> a	lue: 0.00000	.58822	σ
female = 1		ABC	-0.19065	Δσ	0.19065	σ
female = 0		() ABC	0	(	D	
race = white		ABC	0.70347	Δσ	-0.70347	σ
race = black		() ABC	0	(	D	
race = hispanic		ABC	0.09679	Δ σ	-0.09679	σ
bytest		123	-1.04493	σ	1.04493	σ

Figure 4.76: Likelihood ratio p-value

• Next to each coefficient you will find one icon indicating if it is significant (see Figure 4.77) or non-significant (see Figure 4.78).



Figure 4.77: Significant icon



Figure 4.78: Non-significant icon

The significance of a coefficient is determined by comparing the *p*-value against the significance level selected in the top menu (see Figure 4.75). If the *p*-value is higher than the significance level, the coefficient will be non-significant. If the *p*-value is lower than the significance level, the coefficient will be significant. A good practice is to retrain the logistic regression removing the non-significant coefficients. However, in most cases, the model performance should not be affected.

Sources Datasets	Supervised - Unsup	ervised  Prediction:	NEW) s ▼ Tasks			WhizzML	- •
≙ <b>'</b> ₽₹- <b>'</b> ₽ _* •	••• <u>)</u> 1	transport dataset - s	sample (80.00%)'s log	jistic	$\mathbf{\Phi}$	<b>८</b> , (≡) ⁵ ,	(
INSTANCES 3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS	OBJECTIVE FIELD ED_DISCRETIZED	BIAS YES		FIELD CODINGS YES	
	ARIABLES: Filter bias or p	redictors	SIGNIF	ICANCE LEVEL:	0.1 🔻	4 <u>∎</u> 2	csv
Bias and predictors		🗘 Туре	12 years	\$	13-18 years		;
Likelihood ratio 👔			378.39908	σ	378.39908	$\bigtriangleup$	σ
Bias		123	-0.58822	σ	0.58822	$\Delta$	σ
female = 1		ABC	-0.19065	σ	0.19065	$\bigtriangleup$	σ
female = 0		()) ABC	0		0		
race = white		ABC	0.70347	σ	-0.70347	$\Delta$	σ
race = black		() ABC	0	ignificant coefficie	0 ot		
race = hispanic		ABC	0.09679		-0.09679		σ
bytest		123	-1.04493	σ	1.04493	$\Delta$	σ

Figure 4.79: Significance icons for coefficient estimates

Next to each icon indicating the significance of a coefficient, you will find a σ symbol. If you mouse over it, a tooltip will display a summary of the stats for that coefficient. (See Figure 4.80.) First, you will find the *p*-value from the Wald test¹¹. As mentioned in the previous point, this *p*-value is compared against the selected significance level to determine the coefficient's significance. Then, associated with the Z score chart, you will find the Z score value, the confidence interval for a 95% confidence and the standard error, or variance, of the coefficient estimate.

¹¹https://en.wikipedia.org/wiki/Wald_test

Sources Datasets	Supervised - Unsu	pervised  Predictions	▼ Tasks			WhizzML	•
a 17. 67.	•••• <u>)</u>	transport dataset - s	ample (80.00%)'s log	jistic	$\mathbf{\Phi}$	(≡) ⁵ .	6
INSTANCES 3,036	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD ED_DISCRETIZED	BIAS Yes		FIELD CODINGS YES	
	VARIABLES: Filter bias or p	oredictors			0.1 🔻	₫ <u></u>	CSV U
Bias and predictors		\$ Туре	12 years	\$	13-18 years		
Likelihood ratio 👔			378.39908	σ	378.39908		σ
Bias		123	-0.58822	σ	0.58822	Δ	σ
female = 1		ABC	-0.19065	Δ σ	0.19065	$\Delta$	σ
female = 0		(I) ABC	P-VALUE (WALD TEST): 0	.02618	0		
race = white		ABC	0.70347 s	5% 🛆 🖸	-0.70347	$\Delta$	σ
race = black		(I) ABC	0 -0.	2.224	0		
race = hispanic		ABC	CONFIDENCE INTERVAL 0.09679 -3.598-1	+2.26e-2	-0.09679		σ
bytest		123		8.57e-2	1.04493	$\Delta$	σ
dadcoll = 0		(]) ABC	0		0		

Figure 4.80: Summary of stats per coefficient

Note: remember that your categorical fields will automatically be configured with dummy coding when stats are enabled to avoid multicollinearity and the dummy class selected will be the first one in alphabetical order. Learn more about field codings and stats configuration in Subsection 4.4.10 and Subsection 4.4.7 respectively.

You can download all the **stats** information by clicking in the download CSV icon in the top menu to the right. (See Subsection 4.7.1.)

# 4.6 Logistic Regression Predictions

## 4.6.1 Introduction

The ultimate goal in building a logistic regression is being able to make predictions with it. In BigML, you can make predictions for **single instances** or for **many instances in batch**. Each prediction comes with a measure indicating the probability of the predicted class, a percentage ranging from 0% up to 100%. For each prediction, BigML also provides the probabilities for the rest of classes in the objective field.

The predictions tab in the main menu of the BigML Dashboard is where all your saved predictions are listed. (See Figure 4.81.) You can **search** your predictions by name clicking on the search option on the top menu. In the predictions list view, you can see, for each prediction, the **logistic regression** icon used for the prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created).

Sources	Datasets	Supervised -	Unsupervised -	Predict	tions 🔻 Tasks				Whizzł	ML 🔻
1	e				Predictions				Ô	Q,
ា	Name			\$	Objective	\$	Prediction	\$	ė.	<b>†</b>
	No predictions									
Show 10	predictions			Nop	redictions found				I< <	> >

Figure 4.81: Predictions list view

When you first create an account at BigML, or every time that you start a new project, your list of predictions will be empty. (See Figure 4.82.)

Sources	Datasets	Supervised -	Unsupervised - Pre	dic	tions 🝷 Tasks	_	WhizzM	il 🕶
=	-				Predictions		Ô	0
សំ ា	Name			\$	Objective	\$ Prediction	\$	(
				N	o predictions			
Show 10	predictions			No p	redictions found		I< < >	×

Figure 4.82: Empty predictions list view

Logistic regression predictions are saved under the CLASSIFICATION & REGRESSION option in the menu (see Figure 4.83.)

	CLASSIFICATION & REGRESSION V CLUSTER ANALYSIS		Q
Name Name	ANOMALY DETECTION	Prediction	¢ 🛗 <
	ASSOCIATION DISCOVERY		
	TOPIC MODELING		

Figure 4.83: Menu options of the predictions list view

Select the list for your single instances predictions or your batch predictions by clicking on the corresponding icons. (See Figure 4.84 and Figure 4.85.)



Figure 4.84: Single predictions icon



Figure 4.85: Batch predictions icon

## 4.6.2 Creating Logistic Regression Predictions

BigML provides two options to predict with your logistic regressions explained in the following subsections:

- PREDICT: to predict one single instance
- BATCH PREDICTION to predict multiple instances in batch.

#### 4.6.2.1 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the input fields used by the logistic regression, so you can easily set the values and get an immediate response.

Follow these steps to create a single prediction:

1. Click PREDICT in the logistic regression 1-click action menu. (See Figure 4.86.)

Sources Datasets	Logistic Regressions 🔻	Clusters Anomal	ies Associati	ons Predictions	Tasks	Scripts 👻
a s ^a il ^a	¶n, •••• <u>∕</u> L	.oan risk data datas	et   Training (8	30%)'s logisti	🇊 🗘	(≣) → (آ)
INSTANCES	DEFAULT NUMERIC	MISSING NUMERICS	OBJECTIVE	PREDICT		LIZE
800	NOT DEFINED	TRUE	CLASS	BATCH PREDIC	CTION	E
				EVALUATE		
				🔎 DELETE LOGIS	TIC REGRESSION	· · · · · · · · · · · · · · · · · · ·
				MOVE TO		

Figure 4.86: Predict using the 1-click action menu

Alternatively, click PREDICT in the **pop up menu** in the list view. (See Figure 4.87.)

Sources	Datasets Logistic Regres	clusters	Anomalies	Associations	Predictions	Tasks		Scrip	pts 🔻
		<u> </u>	ogistic Re	gressions					Q,
II	Name		Objective	\$	<b>***</b>	₿ ≎	<u>○</u>	\$	\$
ıl <mark>î</mark>	Loan risk data dataset   Training (80%	%)'s logistic regression	class		4h 10min	108.6 KB	1	0	0
ılı	US car accidents da		Injury Seve	erity	5h 44min	741.0 KB		1	
ılî	Arrythmia diagnosis	BATCH PREDICTION	Arrhythmia	1	5h 45min	536.2 KB			
III.	Loan risk data datas	TAILS	class		5h 47min	135.8 KB			
ılî	Loan risk data datas	DELETE LOGISTIC REGRESSION	class		1d 3h	135.8 KB		2	
ılî	Arrythmia diagnosis uataset s rogist		Arrhythmia	1	1d 6h	536.2 KB			
ılî	Loan risk data dataset's logistic regi	ression	class		1d 7h	135.8 KB		2	
ılı	US car accidents dataset's logistic n	egression	Injury Seve	erity	1d 11h	741.0 KB			
ıl <mark>e</mark>	Arrythmia diagnosis dataset's logist	ic regression	Arrhythmia	1	1d 11h	536.2 KB			
ılı	Titanic Survival dataset's logistic reg	gression	Survived		1d 11h	75.3 KB			
Show 10	Iogistic regressions	1 to 1	10 of 11 logistic re	gressions			Κ	< 1 2	××

Figure 4.87: Predict using the pop up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the logistic regression as input fields. (See Figure 4.88.)

Sources Datasets S	upervised  Unsupervised	Predictions	<ul> <li>Tasks</li> </ul>		WhizzML	•
<b>*</b> *	Predict u	ising Loan risk	data dataset - san	nple	(≡) ⁵ ,	G
P		class: bad		72.80%		(
	72.80%					jsoi
			27.20%	-		
	vad		400 ⁰			
Select a positive class	Probability threshold:	0	50%		All input fields:	✓
checking_status		🗹 di	uration		ा स	_
0<=X<200		v	0	0	⁸⁹ 44	4
credit_history		🗹 p	urpose		[	2
all paid		¥	business			¥
ob		🗹 n	um_dependents		<u>ा</u> (	2
high qualif/self emp/mgmt		Ŧ	0	8	2	1
own_telephone		🗹 fo	oreign_worker		(	۷
none		•	no			Ŧ
New prediction name						

Figure 4.88: Logistic regression prediction form

3. Select the fields to be used for your prediction (Figure 4.89.) Non-selected fields will be considered as missing values during the prediction.

Sources Datasets	Supervised - Unsuperv	rised - Predict	tions 👻 Tasks		WhizzML 🗸
<b>T</b> *	Pre-	dict using Loan	risk data dataset - s	sample	<b>(</b> ∌- (≡) [*] , (
<u>r</u> a		class: bad		72.80%	(
	72.80%				
			27.20%		
	trad		6000		
Select a positive class	<ul> <li>Probability threshold:</li> </ul>		50%		
					All input fields: 🗹
checking_status		<b>S</b>	duration		4P 🕑
0<=X<200		v	0	-0	⁸⁹ 44
credit_history		<b></b>	purpose		<b>e</b>
all paid		v	business		v
job		<b></b>	num_dependents		াচ 🗹
-			0		2
high qualif/self emp/mgmt		Y		- C	1
own_telephone		۲	foreign_worker		۲
none		v	no		¥
New prediction name					
Loan risk data dataset	- sample (80.00%)				Save

Figure 4.89: Logistic regression predictions form

If your logistic regression was not trained with **Missing numerics** (see Subsection 4.4.5) you won't be able to disable your numeric fields for the prediction and a warning message will appear instead: "This field cannot be disabled because your model has been trained without missing numerics".

- 4. Set input values for your selected fields. BigML supports numeric, categorical, text and items fields as inputs.
- 5. Get the prediction at the top of the view along with the predicted class probability. (See Figure 4.90.) BigML predictions are synchronous, i.e., when you send the input data, you get an immediate response. Moreover, single predictions from the BigML Dashboard are performed locally, so unless you save your prediction, it will not consume any credits and it will be updated instantly when you change your input values. Learn more about local predictions in Subsection 4.6.4.1.1.

Sources Datasets Supe	rvised - Unsupervised	Predict	ions - Tasks			/hizzML 🔻
<b>Z</b> ≁	Predict	using Loan	risk data dataset - s	ample	G-	(≡)≁_ (
2		class: bad		72.80%		
	72.80%				PNG	
			27.20%			
	wad		900 ⁰			
Select a positive class v Pr	obability threshold:	0	50%		All input	fields: 💽
hecking_status		۲	duration			್ರಾ 💽
0<=X<200		¥	0		89	44
redit_history		≤	purpose			۲
all paid		¥	business			Ŧ
ob		€	num_dependents			4 <u>1</u> 2 💽
high qualif/self emp/mgmt		¥	0	6	2	1
wn_telephone		€	foreign_worker			۲
none		v	no			v
lew prediction name						
Loan risk data dataset - samp	le (80.00%)				Sav	0

Figure 4.90: Get the logistic regression prediction

6. Below the prediction you can see a histogram view containing the rest of your **class probabilities** (Figure 4.91). You can download all the probabilities in PNG format, in CSV or JSON file by clicking on the corresponding icons. (See Subsection 4.6.4.1.)

<b>7</b> .	Predict	using Loan r	isk data dataset - s	sample	<b>(</b> ∌- (≡) ⁵ - (
2		class: bad		72.80%	
	72.80%	_			
			27.20%		
			۵		
	bad		4000		
	<ul> <li>Probability threshold:</li> </ul>		50%		
Select a positive class	Probability threshold:		5076		All input fields:
hecking_status		<b>S</b>	duration		্যান চিনাল বিষ্ণু বিষ্ণ
			0		89
0<=X<200		Ψ		-0	44
redit_history			purpose		٩
all paid		v	business		v
aii paid			Dusiness		
b		<b></b>	num_dependents		4P (
high qualif/self emp/mgmi	1	v	0		2
	•			0	1
wn_telephone		€	foreign_worker		٩
none		v	no		
lew prediction name					
Loan risk data datase	t cample (80,00%)				Save

Figure 4.91: All classes probabilities distribution

7. Optionally, you can Save the logistic regression prediction, so you will find it afterwards in the predictions list view. (See Figure 4.92.)

Sources Datasets Superv	sed - Unsupervised -	Predictio	ons 👻 Tasks		WhizzML 👻
	Predict u	sing Loan r	isk data dataset - s	ample	<b>65</b> - (≡) ⁵ - <b>6</b>
[ ²		lass: bad		72.80%	(
	72.80%		27.20%		
	web		Soob		
Select a positive class Y Prol	pability threshold:	0	50%		All input fields: 🕑
okecking_status 0<=X<200		*	duration 0		89 44
credit_history		≤	purpose		۲
all paid		•	business		v
ob		<b></b>	num_dependents		TP 🗹
high qualif/self emp/mgmt		¥	0	0	2
own_telephone			foreign_worker		<b></b>
none		Ŧ	no		v
New prediction name					
Loan risk data dataset - sample	(80.00%)				Save

Figure 4.92: Save your logistic regression predictions

Note: this option is only available from the BigML Dashboard for logistic regressions with less than 100 fields. If you want to perform single instance predictions for a higher number of fields, use the BigML API¹².

### 4.6.2.1.1 Logistic Regression Prediction with Images

BigML logistic regressions can be trained from images using extracted image features (Subsection 4.2.3). Because image features are automatically generated numeric fields, creating logistic regression predictions with images is the same as creating other logistic regressions. The only thing different is input fields of images.

Note: When the input fields contain images, in order to create the single prediction, BigML will extract image features automatically to match what were used in the dataset to train the logistic regression.

¹²https://bigml.com/api/predictions

Sources	Datasets	Supervised 🗸	Unsupervised •	Predictions	<ul> <li>Tasks</li> </ul>			WhizzM	LŦ
<b>T</b> .			Predict	using grape-str	awberry re	snet18		<	(
<u>s</u> z				label: ?		?			۲
Select a p	positive class v	Probability th	reshold:	0	50%				-
image_id	l			<b></b>				All input fields:	M
	Select image			* 🗋					
New pree	1659.jpg closed, im	age, 2 fields (1 pat	h, 1 image)						
grape-s	696969.jp open, imag	eg ge, 2 fields (1 path,	1 image)				۵.	Predict	
	predict-in closed, im	nages.zip age, 6 sources, 23	6 fields (234						

Figure 4.93: Select a single image source in the image input field

The logistic regression in Figure 4.93, "grape-strawberry resnet18", was created from a dataset containing image features extracted from a pre-trained CNN, *ResNet-18*. Creating a prediction using the logistic regression will be directed to the **prediction form** which presents all input fields used by the logistic regression. One of them is the image field. Because this is a single prediction, an image is input by using a single image source. Clicking on the input field box, single image sources available will be in the dropdown list. There is also a search box which can be used to locate specific ones.

Sources Datase	ts Supervised <del>-</del> Unsupervis	ed - Predictions -	Tasks	WhizzML 🔻
***	Prec	lict using grape-strawb	perry resnet18	(≡) ⁵ √ (Ξ)
₫2		label: ?	?	۲
Select a positive class	s v Probability threshold:	50	%	
image_id		<b>V</b>		All input fields: 🗹
Select ima	ge			
	06969.jpeg pen, image, 2 fields (1 path, 1 image)			
	edict-images.zip osed, image, 6 sources, 2 List componen		e	Predict
	ape-strawberry.zip osed, image, 70 sources, 237 fields (1 c	12		

Figure 4.94: List the components of a composite source

Oftentimes single image sources were used for creating a composite source, they become component sources of the composite source. Or an image was uploaded as a part of an archive file (zip/tar) which created a composite source. In those cases, the composite source will be shown in the dropdown list, along with an icon "List components". In the example in Figure 4.94, predict-images.zip is a composite source, click on the icon to show its component sources.

Sources	Datasets	Supervised -	Unsupervised -	Predictions -	Tasks	_		WhizzML 👻
₽.			Predict us	sing grape-strav	wberry resne	t18		ૐ- (≡)⁵- ③
<u>1</u> 2			la	abel: ?		?		۲
Select a p	oositive class 🔻	Probability thr	reshold:	-0	50%			
								All input fields: 🥑
image_id				≤				
	Select image			^ D				
	🗧 🔛 pred	lict-images.zip						
			c	2				
New pree grape-s	156015						e.	Predict
grape-3	1659.jp	9						Trealer
	And Person in the	age, 2 fields (1 path,	1 image)					
	328320. open, im	J <b>P9</b> age, 2 fields (1 path,	1 image)					

Figure 4.95: Select a component of a composite source

After the component sources of the composite are listed, scroll the dropdown list to find the desired one, then click to select it, as shown in Figure 4.95. There is also a search box to locate specific component sources.

Sources Datasets	Supervised 👻 Unsuper	vised - Predicti	ons 🔻 Tasks		WhizzML 🔻
*** **	2	Predict using blue	uebird-gps-ta	gged	<b>€</b> 5- (≡) ⁵ - <b>()</b>
P		label: ?		?	۲
Select a positive class	Probability threshold:		50%		
					All input fields: 🗹
image_id		۷	gpslongitude		<b>₽ ≤</b>
Select image		* <b>D</b>	- 136.32	0	-97.10325
gpslatitude		ন্দু 🗹	gpsaltitude		ন্দু 🗹
23.57	58.08	37.57352	0	6	2217 <b>709</b>
New prediction name					
bluebird-gps-tagged					Predict

Figure 4.96: Logistic regression image prediction form, more fields

In addition to images, logistic regressions may use other fields, which will be in the prediction form too. As shown in Figure 4.96, all the fields can be selected, and their input values be set by dragging the knobs on the sliders or by entering precise values in their input boxes.

Once all fields are selected, click on the green button Predict to create a prediction.

Sources Datasets Superv	ised - Unsupervised - Predict	ions ▼ Tasks	WhizzML 🗸
	grape-strav	vberry resnet18	
₫°	label: strav	vberry 97.21%	۲
lin .	97.21%		
		2.79%	
	strentert	diate.	
	9		
Select a positive class	ability threshold:	50%	
			All input fields: 🕑
image_id	€		
696969.jpeg	× •		
New prediction name			
grape-strawberry resnet18			Predict

Figure 4.97: Logistic regression image single prediction

After a new prediction is created, as shown in Figure 4.97, the predicted class is at the top of the form along with its probability. The prediction interface is the same as ones created by non-image logistic regression. Everything described earlier in this section (Subsection 4.6.2.1) applies.

#### 4.6.2.2 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the logistic regression you want to use to make predictions and a dataset containing the instances you want to predict. BigML will create a prediction for each instance in the dataset.

Follow these steps to create a batch prediction:

1. Click on BATCH PREDICTION option under the logistic regression 1-click action menu (Figure 4.98)

Sources Datasets	Logistic Regressions	Clusters Anomal	lies Associati	ons Predictions	Tasks	Scripts <del>-</del>
₽ St IV	° )	Loan risk data datas	set   Training (	80%)'s logisti	<b>i</b>	ૐ- (≡)∕- 0
INSTANCES	DEFAULT NUMERIC	MISSING NUMERICS	OBJECTIVE	PREDICT		LIZE
800	NOT DEFINED	TRUE	CLASS	BATCH PREDIC		Æ
				EVALUATE		
			_	J DELETE LOGIS	TIC REGRESSION	
				MOVE TO		
Probability					OBJECTIVE FIELD	,

Figure 4.98: Create batch prediction using 1-click action menu

Alternatively, click on CREATE BATCH PREDICTION in the pop up menu of the list view (Figure 4.99).
Sources	Datasets	Logistic Regressions -	Clusters An	iomalies Associatio	ons Predictions	Tasks		Scri	pts 🔻
			Log	istic Regressions					Q
ill	Name		\$	Objective	≎ ∰ ≎	<b>ê</b>	08 \$	\$	1
ılî	Loan risk data data	aset   Training (80%)'s logis	tic regression 🗴	class	1min	108.6 KB	0	0	0
ıl <mark>e</mark>	Diabetis diagnosis			Diabetes	1min	21.6 KB			
ılî	2012 POTUS Winne	CREATE BATCH PR	EDICTION	Winner	2min	210.3 KB			
ılî	Arrythmia diagnosi	is 🖉 view details		Arrhythmia	2min	428.3 KB			
ıl <mark>o</mark>	US Car Accidents of	di Z DELETE LOGISTICI	REGRESSION	Injury Severity	2min	548.1 KB			
ılî	Loan risk data data	aseri maining (ov %) s iogis	acregression	class	2min	108.6 KB			
ihow 10	Iogistic regressi	ions	1 to 6 of	6 logistic regressions				K < 1	> >

Figure 4.99: Create batch prediction using pop up menu

 Select the dataset containing all the instances you want to predict. The instances should contain the input values for the fields used by the logistic regression as input fields. From this view you can also select another logistic regression from the selector or even a model or ensemble by clicking on the icons on the top left menu. (See Figure 4.100.)

1 ¹ 2 K ²	New B	atch Prediction	۵. (
Loan risk data dataset   Tra	ining (80%)'s logistic reg × 💌	Search dataset	Å
108.6 KB 21 size fields	Thu, 26 May 2016 10:07:52 800 instances	I Please enter 1 more character	
Description:	$\odot$		
rediction name:			
Batch Prediction of Loan risk data	dataset   Training with data	Reso	et Predict

Figure 4.100: Select dataset for batch prediction

3. After the logistic regression and the dataset are selected, the batch prediction configuration options will appear along with a preview of the prediction output (a CSV file). (See Figure 4.101.) The default output format includes all your prediction dataset fields and adds an extra column with the class predicted. See Subsection 4.6.3 ofr a detailed explanation of all configuration options.

61 Fr 1			
Loan risk data dataset   Training (8	10%)'s logistic reg × ×	Loan risk data dataset	× *
	Thu, 26 May 2016 10:07:52	- Al	Thu, 26 May 2016 10:05:2
108.6 KB 21 size fields	800 instances	135.8 KB 21 size fields	1,000 instances
Description:	$\overline{\mathbf{O}}$	Description:	$\odot$
Preview of the prediction file (using the type	of each field)		
checking_status,duration,credit_hi	story,purpose,credit_amount		
	story,purpose,credit_amoun since,property_magnitude,ag orker,class,class	ge,other_payment_plans,housing,exi	
checking_status,duration,credit_hi al_status,other_parties,residence_ ependents,own_telephone,foreign_wc ABC,123,ABC,ABC,123,ABC,ABC,123,AA ABC,123,ABC,ABC,123,ABC,ABC,123,AA ABC,123,ABC,ABC,123,ABC,ABC,123,AB	story,purpose,credit_amouni _since,property_magnitude,a prker,class,class GC,ABC,123,ABC,123,ABC,ABC, IC,ABC,123,ABC,ABC,123,ABC,ABC,	pe,other_payment_plans,housing,exi L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC,ABC	
al_status,other_parties,residence_ ependents,own_telephone,foreign_wc ABC,123,ABC,ABC,123,ABC,ABC,123,AB ABC,123,ABC,ABC,123,ABC,ABC,123,AB	story, purpose, credit_amouni _since, property_magnitude, a prker, class, class GC, ABC, 123, ABC, 123, ABC, ABC, , GC, ABC, 123, ABC, 123, ABC, ABC, , GC, ABC, 123, ABC, 123, ABC, ABC, ABC, ,	je,other_payment_plans,housing,exi L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC	
checking_status,duration,credit_hi al_status,other_parties,residence_ ependents,own_telephone,foreign_wc ABC,123,ABC,ABC,123,ABC,ABC,123,AF ABC,123,ABC,ABC,123,ABC,ABC,123,AF ABC,123,ABC,ABC,123,ABC,ABC,123,AF	story, purpose, credit_amouni _since, property_magnitude, a prker, class, class GC, ABC, 123, ABC, 123, ABC, ABC, , GC, ABC, 123, ABC, 123, ABC, ABC, , GC, ABC, 123, ABC, 123, ABC, ABC, ABC, ,	je,other_payment_plans,housing,exi L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC L23,ABC,123,ABC,ABC,ABC,ABC	

Figure 4.101: Configuration options for logistic regression batch prediction

4. By default, BigML generates an output dataset with your batch predictions that you can later find in your datasets section in the BigML Dashboard. This option is active by default but you can deactivate it by clicking on the icon shown in Figure 4.102.

Sources Datasets	<ul> <li>Logistic Regression</li> </ul>	ns ▼ Clusters Anomalies	Associations Predictions	Tasks Scripts
1 <mark>6 1</mark> 6		New Bato	ch Prediction	<b>5</b> - (
សា ៩ស	L			
Loan risk	data dataset   Training (8	0%)'s logistic reg × ×	Loan risk data dataset	× ×
•• •	lass	Thu, 26 May 2016 10:07:52	1	Thu, 26 May 2016 10:05:25
108.6 KB size		800 instances	135.8 KB 21 size field	
Description:		⊚	Description:	$\odot$
Configure	i - ti - conta di -			•
checking_statu: al_status,othe ependents,own_ ABC,123,ABC,ABC ABC,123,ABC,ABC ABC,123,ABC,ABC	r_parties,residence_ telephone,foreign_wo C,123,ABC,ABC,123,AB C,123,ABC,ABC,123,AB C,123,ABC,ABC,123,AB	story, purpose, credit_amount since, property_magnitude, ag rker, class, class C, ABC, 123, ABC, 123, ABC, ABC, J C, ABC, 123, ABC, 123, ABC, ABC, ABC, 12 C, ABC, 123, ABC, 123, ABC, ABC, ABC, ABC, ABC, ABC, ABC, ABC	23,ABC,123,ABC,ABC,ABC,ABC	
	C,123,ABC,ABC,123,AB	C,ABC,123,ABC,123,ABC,ABC,1 C,ABC,123,ABC,123,ABC,ABC,1	.23,ABC,123,ABC,ABC,ABC,ABC .23,ABC,123,ABC,ABC,ABC,ABC	

Figure 4.102: Create a dataset from batch prediction

5. After you configure your batch prediction, click on the green button Predict to generate your batch prediction. (See Figure 4.103.)

A A. Z		New Date	ch Prediction		<b>\$</b> ~
តា fa 1					
Loan risk data da	ataset   Training (80%)':	s logistic reg × v	Loan risk data da	ataset	× *
o□I ⊕ class		Thu, 26 May 2016 10:07:52	ılî		Thu, 26 May 2016 10:05:25
108.6 KB size	21 fields	800 instances	135.8 KB size	21 fields	1,000 instances
Description:		$\odot$	Description:		$\overline{\mathbf{O}}$
ិភ្នុះ Configure					۲
	file (using the type of ea	ich field)			
		ry,purpose,credit_amount	,savings_status,employme	nt,installment_com	mitment,person
al_status,other_parti ependents,own_telepho	tion,credit_histor ies,residence_sinc one,foreign_worker	ce,property_magnitude,ag r,class,class	e,other_payment_plans,ho	using,existing_cre	
checking_status, durat al_status, other_parti ependents, own_telepho ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A	tion,credit_histor ies,residence_sind one,foreign_worker ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE	ce,property_magnitude,ag r,class,class BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,1		using,existing_cre 3C 3C 3C 3C	
checking_status, durat al_status, other_parti ependents, own_telepho ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A ABC, 123, ABC, ABC, 123, A	tion,credit_histor ies,residence_sind one,foreign_worker ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE ABC,ABC,123,ABC,AE	ce,property_magnitude,ag r,class,class BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,A BC,123,ABC,123,ABC,ABC,1	e,other_payment_plans,hou .23,ABC,123,ABC,ABC,ABC,AI .23,ABC,123,ABC,ABC,ABC,AI .23,ABC,123,ABC,ABC,ABC,ABC,AI .23,ABC,123,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC,ABC	using,existing_cre 3C 3C 3C 3C	

Figure 4.103: Create batch prediction

6. When the batch prediction is created, you will be able to **download the CSV file** containing all your dataset instances along with a prediction for each one of them. (See Figure 4.104.)

Sources Datasets Logistic Regressions ▼ Clusters Anon	nalies Associations <b>Predictions</b> Tasks	Scripts 🔻
🔒 🛸 💦 🥂 🚥 🔚 Batch Pred	diction of Loan risk da	<b>5</b> - ()
Loan Risk Data Dataset   Training (80%)' 💿	Loan Risk Data Dataset	$\overline{\mathbf{O}}$
Output preview checking_status, duration, credit_history, purpose, credit_c al_status, other_parties, residence_since, property_magnitu ependents, own_telephone, foreign_worker, class, class no checking, 24, existing paid, used car, 5433, no known savi insurance, 26, none, rent, 1, high qualif/self emp/mgmt, 1, yes 0<=X<200, 36, delayed previously, new car, 8086, 100<=X<500, 2 emp/mgmt, 1, yes, yes, bad, bad no checking, 24, existing paid, radio/tv, 1376, 500<=X<1000, 4 div/dep/mar, none, 1, car, 28, none, own, 1, skilled, 1, none, ves	ude,age,other_payment_plans,housing,existing_credi ings,unemployed,2,female div/dep/mar,none,4,life s,yes,good,good >=7,2,male single,none,4,car,42,none,own,4,high qu 4<=X<7,4,female	ts,job,num_d

Figure 4.104: Download batch prediction CSV file

7. If you didn't disable the option to create a dataset explained in step 4, you will also be able to access the **output dataset** from the batch prediction view. (See Figure 4.105.)



Figure 4.105: Batch prediction output dataset

### 4.6.2.2.1 Batch Prediction with Images

BigML logistic regression can be trained from images using extracted image features (Subsection 4.2.3). The input of a batch prediction is a dataset. So when creating a batch prediction with images, the dataset has to have the same image features used to train the logistic regression. The image features are in the dataset used to create the logistic regression.

1 T	New Batch Pred	iction	<b>5</b> - (
ରେ 🐔 🗶 🗾			
grape-strawberry resne	et18 × •	Search dataset	*
	predict-images resnet18	predict	٩
	6 instances, 514 fields (512 numeric, 1 path, image), 1 non-preferred	predict-images resnet18           6 instances, 514 fields (512 numeric, 1 path)	, 1 image), 1 n
	514 70 fields instances	predict-images 6 instances, 236 fields (234 numeric, 1 path	, 1 image), 1 n
Description:	$\odot$	iris dataset - extended - sample (100.00 60 instances, 142 fields	%) - extende
		iris dataset - extended - sample (100.00 60 instances, 139 fields	%) - extende
Prediction name:			,

Figure 4.106: Batch prediction using an image dataset

As shown in Figure 4.106, the input for the logistic regression batch prediction is selected as predict-images resnet18, which is a dataset consisting of six images and contains image features extracted from a pre-trained CNN, *ResNet-18*.

Image features are configured at the source level. For more information about the image features and how to configure them, please refer to section Image Analysis of the Sources with the BigML Dashboard¹³[11].

¹³https://static.bigml.com/pdf/BigML_Sources.pdf

For the rest of batch predictions with images, including batch prediction configuration options and output datasets, everything stated earlier in current section (Subsection 4.6.2.2) applies.

### 4.6.3 Configuring Logistic Regression Predictions

BigML provides several options to configure your predictions such as setting a probability threshold Subsection 4.6.3.1), default values for your missing numeric values (see Subsection 4.6.3.2), fields mapping (see Subsection 4.6.3.3), and output file settings (see Subsection 4.6.3.4.)

### 4.6.3.1 Probability threshold

Probability thresholds usually makes sense when you want to minimize false positives at the cost of false negatives. The positive class will be predicted if its probability is greater than the given threshold, otherwise the following class with greater probability will be predicted instead.

To configure a threshold for your predictions follow these steps:

1. Use the selector shown in Figure 4.107 to select the **positive class**, i.e., the class for which you want to apply the probability threshold.



Figure 4.107: Select the positive class for single predictions

2. Then set a threshold value between 0% and 100% using the slider (see Figure 4.108). The positive class will be predicted if its probability is greater than the given threshold, otherwise the following class with greater probability will be predicted instead.

Sources Data	sets Supervise	d 🕶 Unsupervised	d 👻 Predicti	ons 🔹 Tasks		WhizzML 👻
¥.		Predict	t using Diabe	tes diagnosis da	ataset   Trai	<b>⊘</b> ₋ (≡) ⁵ ₋ <b>⊙</b>
T	(	Di	abetes: True		78.46%	۲
				78.46%		
		21.54% çə ^{şe}	- 1	True		
True	× v Probat	ility threshold: =	0	20%		All input fields: 🕑
Pregnancies		21	ন্দু 🗹 10	Glucose	(C)	248 <b>124</b>
Blood pressure	0	142	ন্দু 🗹 71	Skinfold 0		78 <b>39</b>
<b>Insulin</b> 0		930	्राः 🗹 465	<b>BMI</b> 0.0		83.88 40.74

Figure 4.108: Set the probability threshold for single predictions

For batch predictions you can find the same options under the CONFIGURE panel:

ources Datasets Supervised -	Unsupervised - Prediction	ons 🔻 Tasks	_	WhizzML 🔻
T	New Batc	h Prediction		<b>5</b> - (i)
£1 F£ ∠ ≈				
Diabetes diagnosis dataset   Train	ing (70%)'s logist × ×	Diabetis dia	gnosis dataset - extended	×
Diabetes	Fri, 09 Jun 2017 16:41:41	ıl		Wed, 24 May 2017 11:49:56
18.9 KB 9 size fields	537 instances	5.6 KB size	11 fields	153 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
්ය Configure				۲
Select a positive class	ty threshold:	50%		
Default numeric value: ③ Select a defa	ult value			
Excluded fields:				◙
1 Make sure that the fields in the Logis	stic Regression and Dataset r	natch		
Fields mapping:			Defau	It fields

Figure 4.109: Set the probability threshold for batch predictions

#### 4.6.3.2 Default Numeric Value

If the dataset used to make the batch prediction contains instances with missing values for the numeric fields, the prediction will not be computed for them, unless you built the logistic regression enabling the **Missing numerics** parameter (see Subsection 4.4.5).

By using the **Default numeric value** before creating your batch prediction, you can easily replace all the missing numeric values by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 4.110.)

ources Datasets Supervised - Unsu	pervised - Predict	ons - Tasks	_	WhizzML 🔻
	New Bate	h Prediction		<b>5</b> - (
£1 €£ £				
Diabetes diagnosis dataset   Training (70%	)'s logist × ×	Diabetis diag	nosis dataset - extended	×
Diabetes	ri, 09 Jun 2017 16:41:41	ılî	W	ed, 24 May 2017 11:49:56
18.9 KB 9 size fields	537 instances	5.6 KB size		153 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
ද් _{තු} Configure				۲
Select a positive class    Probability thres	hold:	50%		
Default numeric value:   Select a default value	v			
Excluded fields:				$\overline{\mathbf{O}}$
1 Make sure that the fields in the Logistic Re	gression and Dataset	match		
Fields mapping:			Default	fields 💿

Figure 4.110: Configure Default numeric value for batch prediction

### 4.6.3.3 Fields Mapping

You can specify which input fields of the logistic regression match with the fields in the dataset contaning the instances you want to predict. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** by clicking on the green switcher. Additionally, you can **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the batch prediction. (See Figure 4.111.)

50%		
		$\overline{\mathbf{O}}$
set match		
	Apply this map	۲
Dataset fields	Auto-mapping by: NAME CON ID	
1. Pregnancies	123 × •	
I. Tregnancies		
2. Glucose	123 × V	
2. Glucose	123 × *	
	aset match	aset match Apply this map Dataset fields Auto-mapping by: NAME Composition 10

Figure 4.111: Configure the fields mapping for batch prediction

Note: Fields mapping from the BigML Dashboard is limited to 200 fields. For batch predictions with a higher number of fields, map your fields using the BigML API¹⁴.

#### 4.6.3.4 Output Settings

Batch predictions return a CSV file containing all your instances and the final predictions. Tune the following settings to customize your prediction file (see Figure 4.112):

- **Separator**: this option allows you to choose the best separator for your output file columns. The default separator is the comma. You can also select the semicolon, the tab, or the space.
- New line: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".
- Output fields: by clicking on the list icon next to the separator selector, you can include or exclude all your dataset fields from your output file. You can also individually select the fields you want to include or exclude using the multiple output fields selector. Note: a maximum of 100 fields can be displayed in this selector, but all your dataset fields will be included in the output file by default unless you exclude them.
- **Headers**: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability column name, etc.). By default, BigML includes the headers.
- **Prediction column name**: customize the name for your predictions column. By default, BigML takes the name of the logistic regression's objective field.
- **Probability**: this option allows you to include an additional column with the probability for the predicted class. By default it is not included in your ouput file.
- **Probability column name**: customize the name for the probability column if you include it in the output file. BigML sets "probability" as the default name.

¹⁴https://bigml.com/api/batchpredictions#bp_batch_prediction_arguments

• All class probabilities: this includes all the probabilities of the objective field classes per instance. This option will add *n* extra columns, one by class in the objective field.

Output settings					6
Separator:	New lin	e:			
, (comma)	• Unix, L	inux or OS X (LF)	<i>*</i> ≡		
Prediction column name:	Р	robability column name:			
a,b,c	%		<b>%</b>		
Output Fields:					
x commonlinkratio_1	123	x commonlinkratio_2	123	* commonlinkratio_3	123
	123	* compression_ratio	123	⊯ embed_ratio	123
× framebased	ABC	* frameTagRatio	123	* hasDomainLink	ABC
eview of the prediction file (using t	the type of each	field)			
rl,urlid,boilerplate,alchem					
nkratio_3,commonlinkratio_4 io,is_news,lengthyLinkDomai	n,linkwords	ore,news_front_page,n			
rl,parametrizedLinkRatio,sp			172 172 ADC ADC	123,ABC,123,123,123,123,12	D ADC ADC

Figure 4.112: Logistic regression output settings for batch predictions

## 4.6.4 Visualizing Logistic Regression Predictions

Logistic regression predictions visualization changes depending on if you are predicting one **single** instance (Subsection 4.6.4.1), or you are predicting multiple instances using the **batch predictions** option (Subsection 4.6.4.2).

#### 4.6.4.1 Single Predictions

For single predictions, find the predicted class given the input fields values at the top of the form along with its probability. (See Figure 4.113.)

Sources Datasets	Supervised -	Unsupervised <del>-</del>	Prediction	is 🗸 Tasks		WhizzN	иL <del>-</del>
T.		Predict us	ing US car	accidents_missir	ngs	<b>(</b> ≡) ⁵ ,-	()
₫¢		Injury Sever	ity: Fatal In	ijury (K)	74.96%		۲
	74.96%	21.91% 0.35			0.83%		
Select a positive class ¥	Probability th	reshold:	0	50%		All input fields	: 🗹
Atmospheric Condition			ا 🗹	Fatalities in crash		-	P 🗹
Blowing Sand, Soil, Dirt			v	0	-0	6	3
Roadway Rural-Local Road or Street			V	Age 0		122	P 🗹
Alcohol Results		0.81	¶ 🗹 ।	Person Type			<b></b>

Figure 4.113: Logistic regression single prediction

Below the prediction, there's a histogram representing the rest of the objective field **class probabilities**. All the class probabilities must sum 100%. Show or hide this view by clicking on the icon highlighted in Figure 4.114.

Sources Datasets	Supervised -	Unsupervised <del>-</del>	Predictions	Tasks	_	Whiz	zML 👻
₩.		Predict usin	ng US car ac	cidents_missir	igs	<ul><li></li><li></li><li></li></ul>	t (
Ŀ		Injury Severi	ty: Fatal Inju	<mark>у (К)</mark>	74.96%		(
	74.95% F8 ^{1,1} M	21.91% 0.35% what the second s		1.94%	0.83%	r c	
Select a positive class v	Probability three	eshold:	-0	50%		All input fiel	de: 🐼
Atmospheric Condition							us. 💽
			_	alities in crash			
Blowing Sand, Soil, Dirt			Fat			6	
Blowing Sand, Soil, Dirt Roadway					-0		₫₽ 🗹 3
			•				4]e € 3
Roadway							⊈ € 3

Figure 4.114: Logistic regression all class probabilities

You can see up to seven different classes at the same time; if you have more than seven classes, you will see that some arrow icons appear next to the histogram so you can see the rest of classes.

**Export** this view in PNG format, in a CSV file, or in a JSON file by clicking on the corresponding icons. (See Figure 4.115.)

Sources Datasets	Supervised -	Unsupervised •	Predicti	ons 🚽 Tasks		WhizzML 👻
₽.		Predict	using US ca	ar accidents_missi	ngs	(=) [*] , (■)
<u>1</u> 7		Injury Sev	verity: Fatal	Injury (K)	74.96%	۲
	74.96%					
	580.18			0.01% <u>1.94%</u>	0.83%	
Select a positive class	• Probability t	hreshold:	Ø	50%		All input fields: 🕑
Atmospheric Condition	ı.		✓	Fatalities in crash		ন্দু 🗹
Blowing Sand, Soil, Dirt			¥	0		
Roadway			≤	Age		শ্রু 🗹
Rural-Local Road or Stree	et		v	0	6	122 <b>61</b>
Alcohol Results			T 🗹	Person Type		<b>S</b>
0.0		0.81	0.39	Bicyclist		v

Figure 4.115: Logistic regression export all class probabilities histogram

Set a **probability threshold** for a selected positive class to minimize false negatives at the cost of false positives. If the probability for the positive class is greater than the established threshold, then the positive class will be predicted. Otherwise, the next class with higher probability will be predicted instead.

Sources Datasets	Supervised - Unsuperv	ised - Predicti	ions 🔻 Tasks		WhizzML 🗸
¥.	Pre	edict using US ca	ar accidents_missi	ngs	<b>€</b> , (≡) ⁵ , (∋)
T	Injur	y Severity: Incap	pacitating Injury (A	21.91%	۲
	74.96% 21.91% 5.8 ⁹⁰ . ^{10¹} 16 ¹⁰	0.35%	0.01% <u>1.94%</u>	0.83%	PNG CSV ISON
Incapacitating Injury × ×	Probability threshold:	-0	17%		All input fields: 🕑
Atmospheric Condition		€	Fatalities in crash		শৃ 🗹
Blowing Sand, Soil, Dirt		*	0	0	⁶ 3
Roadway		≤	Age		শৃষ্ণ 🗹
Rural-Local Road or Street		¥	0	0	122 <b>61</b>
Alcohol Results		T 🗹	Person Type		<b></b>
0.0	0.81	0.39	Bicyclist		v

Figure 4.116: Logistic regression probability threshold

### 4.6.4.1.1 Local Predictions

Local predictions are provided for single instances from the BigML Dashboard which are performed faster at no cost. Local predictions allow you to get a real-time prediction without consuming any credits or requiring any internet connection. This is possible because the logistic regression is **saved in-memory**, so when the input values change, BigML is able to compute predictions in microseconds.

#### 4.6.4.1.2 Prediction explanation

Prediction explanation helps understand why a logistic regression makes a certain prediction. This is very useful in many applications, and the reasons behind a prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

For any logistic regression, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Save (see Figure 4.117).

<b>T</b> *		Predict using Titanic Survi	val	<b>(</b> ≣) [*] - (
<u>F</u> o	Su	rvived: TRUE	65.38%	
		65.3	38%	
	34.62%			
	FALSE	TRU	¢	
Select a positive class v Pr	robability threshold:	50%		
Select a positive class	robability threshold:	50%		All input fields:
	robability threshold:	50% 50% Class/Dept		
	92.5	নু 🕑 Class/Dept		
Age		ন্টু 🗹 Class/Dept		
Age 0.0		নু 🕑 Class/Dept		۲. ۲
Age 0.0				ب
Age 0.0	92.5			· · · · · · · · · · · · · · · · · · ·
Age 0.0	92.5			te the prediction
Age 0.0 Fare today	92.5		explanation	€ ▼ €

Figure 4.117: Explain prediction

The prediction explanation represents the most important factors considered by the logistic regression in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 4.118. The importances across all input fields should sum 100%.



### Figure 4.118: Input field importances

For some input fields you will see a "+" icon next to the importance. This is because the importance may not be only directly associated with the input value, i.e., it can also be explained by other reasons. In the Figure 4.119 below, the importance of 28.81% for the field "Class/Dept" is not only explained by this field being equal to "1st Class". Rather, it is because this field value is not "3rd Class" (which accounts for the majority of importance, 23.98%) and, then because it is "1st Class" (4.83% of importance).

Sources D	atasets Super	vised 👻 Uns	supervised 👻	Predictions -	Tasks	_	_	WhizzML	- •
8 ¥-			R	Titanic Survi	val		4	. (≡) ⁵ .	(
			Survivo	ed: TRUE		65.38%			
PREDICTION EXPI	LANATION						t		
nput data		Importance							
Fare today	24,719.00							60.34%	+
Class/Dept	1st Class							28.81%	+
Age	44.93						Detailed Explana	tion 6.56%	+
Joined	Belfast						!= 3rd Class = 1st Class	23.98 4.83	

Figure 4.119: See the detailed explanation

The prediction explanation for logistic regressions is calculated using the results of over a thousand distinct predictions using random perturbations of the input data. For this reason, the calculation of the explanation may take some time to be computed.

### 4.6.4.2 Batch Prediction

After creating your batch prediction, you get a **CSV file** and, optionally, an **output dataset**. Both outputs are explained in the following subsections.

### 4.6.4.2.1 Output CSV file

The batch prediction generates a CSV file containing your **predictions** for each of your dataset instances in the last column. (See Figure 4.120.)

Sources Datasets Logistic Regressions - Clusters Anomalies Associations Predictions Tasks	Scripts ▼
Batch Prediction of Loan risk da	<b>\$</b> - 0
Loan Risk Data Dataset   Training (80%)' 😌 🕕 Loan Risk Data Dataset	ົ
Output preview checking_status, duration, credit_history, purpose, credit_amount, savings_status, employment, installment_commit al_status, other_parties, residence_since, property_magnitude, age, other_payment_plans, housing, existing_credit ependents, own_telephone, foreign_worker, class, class no checking, 24, existing paid, used car, 5433, no known savings, unemployed, 2, female div/dep/mar, none, 4, life insurance, 26, none, rent, 1, high qualif/self emp/mgmt, 1, yes, yes, good, good 0<=x<200, 36, delayed previously, new car, 8086, 100<=x<500,>=7, 2, male single, none, 4, car, 42, none, own, 4, high qua emp/mgmt, 1, yes, yes, bad, bad no checking, 24, existing paid, radio/tv, 1376, 500<=X<1000, 4<=X<7, 4, female div/dep/mar.none, 1, car, 28, none, own, 1, skilled, 1, none, ves, apod, apod	s,job,num_d
Download batch prediction     Output dataset	

Figure 4.120: Download batch prediction CSV file

You can configure several options to **customize your CSV file**. You can find a detailed explanation of those options in Subsection 4.6.3.4.

See an output CSV file example in Figure 4.121. The column **class** in this example contains the final prediction (it is named by default as your logistic regression's objective field). In this case we are predicting whether a person is a good or a bad candidate for holding a credit. This file has been configured to contain also the **probability** for each prediction.

```
duration,age,amount,purpose,class,probability
24,26,5433,used car,good,0.88785
36,42,8086,new car,bad,0.55526
24,28,1376,radio/tv,good,0.8385
48,31,6758,radio/tv,bad,0.73576
26,30,7966,used car,good,0.7201
12,42,2577,furniture/equipment,good,0.67644
36,30,4455,business,good,0.52227
18,32,1442,new car,bad,0.75488
9,22,276,new car,good,0.57819
```

Figure 4.121: An example of a logistic regression batch prediction CSV file

### 4.6.4.2.2 Output Dataset

By default BigML automatically creates a dataset out of your batch prediction. You can disable this option by configuring your batch prediction. (See Subsection 4.6.3.4.) You will find the output dataset in your batch prediction view as shown in Figure 4.122.



Figure 4.122: Batch prediction output dataset

In the output dataset, you can find an additional **field** (named by default as your logistic regression's objective field) containing the **class predicted** for each one of your instances. If you configured your batch prediction to include the prediction **probabilities** and **all class probabilites** you will be able to find them in the last fields of your output dataset. (See Figure 4.123.)

	••• 📗 Loan ris	sk data dataset	t   Training (80%	%) - batchpre	ediction 🍇 🗸 🍊 (≡)— (
					Q :
lame	Type	Count	Missing	Errors	Histogram
luration	123	800	0	0	
redit_history	ABC	800	0	0	
redit_amount	123	800	0	0	
ge	123	800	0	0	
urpose	ABC	800	0	0	and the second
rediction	ABC	800	0	0	
robability	123	800	0	0	
ood probability	123	800	0	0	
ad probability	123	800	0	0	

Figure 4.123: Logistic regression batch prediction output dataset

## 4.6.5 Consuming Logistic Regression Predictions

You can fully used single and batch predictions via the BigML API and bindings. The following subsections explain both tools.

### 4.6.5.1 Using Logistic Regression Predictions via the BigML API

Logistic regression predictions have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, and delete single and batch predictions.

In the example below, see how to create a single prediction using a logistic regression and defining the input data once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"logisticregression": "logisticregression/50650bdf3c19201b64000020",
                           "input_data": {"000001": 3, "000002":4.5, "000003":5}}}'
```

For more information on using logistic regressions through the BigML API, please refer to the documentation¹⁵.

### 4.6.5.2 Using Logistic Regression Predictions via BigML Bindings

You can also create, configure, retrieve, list, update, and delete single and batch predictions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a logistic regression with the Python bindings.

¹⁵https://bigml.com/api/logisticregressions

```
from bigml.api import BigML
api = BigML()
prediction = api.create_prediction(
    "logisticregression/50650bdf3c19201b64000020",
    {"credit_amount": 5, "duration": 2.5})
```

For more information on BigML bindings, please refer to the bindings page¹⁶.

## 4.6.6 Descriptive Information

Each logistic regression prediction has an associated **name**, **description**, **category**, and **tags**. You can find a brief description of each concept in the following subsections. The MORE INFO menu option displays a panel that provides editing options. (See Figure 4.124.)



Figure 4.124: Logistic regression prediction descriptive information

### 4.6.6.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: the name always follows the structure "Prediction for <objective field name>".
- **Batch predictions**: BigML combines your prediction dataset name and the logistic regression name: "Batch prediction of <logistic regression name> with <dataset name>".

Predictions names are displayed on the list and also on the top bar of a prediction view. Predictions names are indexed to be used in searches. Rename your predictions any time from the MORE INFO menu.

The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, but they will always have different identifiers.

### 4.6.6.2 Description

Each prediction also has a **description** that it is very useful for documenting your Machine Learning projects. Predictions take their description from the logistic regression used to create them.

¹⁶https://bigml.com/tools/bindings

Descriptions can be written using plain text and also markdown¹⁷. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 4.125.)

Edit description	>
You can add formatting and links using a simple markdown language: B $I \models 400$	
Write your **description** here	
Description:	
Write your description here	

Figure 4.125: Markdown editor for logistic regression descriptions

Descriptions cannot be longer than 8192 characters.

#### 4.6.6.3 Category

A **category** taken from the logistic regression used to create it is associated with each prediction. Categories are useful to classify predictions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 4.5.

#### 4.6.6.4 Tags

A prediction can also have a number of **tags** associated with it. These tags help to retrieve the prediction via the BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the logistic regression used to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

### 4.6.7 Logistic Regression Predictions Privacy

The link displayed in the **Privacy** panel is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared by using a secret link. (See Figure 4.126.)

¹⁷https://en.wikipedia.org/wiki/Markdown

Sources	Datasets	Logistic Regressions 🔻	Clusters	Anomalies	Associations	Predictions	Tasks	Scrip	ts 🔻
8	ılî	_≊ ••••	Bate	ch Predictio	n of Loan risk	da		∽	6
DETAILS									◙
INFO									ਂ
PRIVACY									۹
Private lin	k   Only you	can view this batchpredictior	n logged into you	ir account.					
		8	https://horiz	on.bigml.com/d	ashboard/batchpr	ediction/5746eeb1	28eb3e728a000003		



## 4.6.8 Moving Logistic Regression Predictions to Another Project

When you create a prediction, it will be assigned to the same **project** where the original logistic regression is located. You cannot move predictions between projects as you do with other resources.

### 4.6.9 Stopping Logistic Regression Predictions

Single predictions are **synchronous** resources, so you cannot cancel them during the creation since you get the result immediately.

Bycontrast, batch predictions are **asynchronous** resources, so you can stop their creation before the task is finished. Use the DELETE BATCH PREDICTION option from the **1-click action menu** (Figure 4.127) or from the **pop up menu** on the list view.



Figure 4.127: Stop logistic regression batch prediction from 1-click action menu

A modal window will be displayed asking you for confirmation. If you stop the prediction during its creation you won't be able to resume the same task again, so if you want to create the same prediction you will have to start a new task.

Lo	Are you sure you want to delete this prediction?	
e	If you delete this prediction, you will no longer have access to its data and you will need to recreate it.	♦ Instance
h Predic		
h Predic	Cancel Delete	

Figure 4.128: Logistic regression delete prediction confirmation

### 4.6.10 Deleting Logistic Regression Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the **1-click action menu** (see Figure 4.129) or using the **pop up menu** on the predictions list view (see Figure 4.130).

Sources Datasets Logistic Regressions   Clus	sters Anomalies	Associations P	redictions Task	s Script	is 👻
a 🖻 🖍 🗶 🚥 🔄	Batch Predicti	on of Loan risk da		<b>3</b> -	(
		Q	BATCH PREDICTIO	AGAIN	
	‰)' <b>⊙</b>		BATCH PREDICTIO	WITH ANOTHER DATAS	
Loan Risk Data Dataset   Training (80	l%)' ♥		BATCH PREDICTIO	USING ANOTHER LOG	í
Output preview		E	NEW BATCH PREDI	CTION	
duration, credit_history, credit_amount, age, pu 24, existing paid, 5433,26, used car, good,0.887 36, delayed previously, 8086,42, new car, bad,0. 24, existing paid, 1376,28, radio/tv, good,0.838 48, existing paid, 6758,31, radio/tv, bad,0.7357 26, existing paid, 7966,30, used car, good,0.720 12, existing paid, 2577,42, furniture/equipment 36, delayed previously, 4455, 30, business, good,	85,0.88785,0.112 55526,0.44474,0. 5,0.8385,0.1615 6,0.26424,0.7357 1,0.7201,0.2799 ,good,0.67644,0.	215 55526 76 .67644,0.32356	DELETE BATCH PRI		
18.all paid.1442.32.new car.bad.0.75488.0.24		,0.47773			

Figure 4.129: Logistic regression delete prediction from 1-click menu

Sources	s Da	asets Logistic Regressions - Clusters Anomalies Associations	Predictions Tasks	Sc	ripts 👻
ŝ		👯 📰 📰 Batch Predictions		Q	<b>5</b> -
ŝ	III	Name \$	Objective \$	Instances \$	<b>#</b> :
<u>ې</u> ر	ılî	Batch Prediction of Loan risk data dataset   Training with Loan risk	class	800	12min
<u> </u>	ılı	Batch Prediction of Loan risk data datas	class	800	19min
×	ılı	Batch Prediction of Loan risk data datas	class	800	20min
×	ılı	Batch Prediction of Loan risk data dataset   Training with Loan risk	class	800	22min
<u> </u>	ılî	Batch Prediction of Loan risk data dataset   Training with Loan risk	class	1K	2h 29min
ihow 10	) ᅌ pre	dictions 1 to 5 of 5 predictions		K < 1	>>

Figure 4.130: Logistic regression delete prediction from pop up menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

	Are you sure you want to delete this prediction?	
e	If you delete this prediction, you will no longer have access to its data and you will need to recreate it.	≎ Instan
h Predic		
h Predic	Cancel Delete	

Figure 4.131: Logistic regression delete prediction confirmation

## 4.7 Consuming Logistic Regressions

Similarly to other models in BigML, logistic regressions are white-boxed models, so you can **download** them and used them locally to make predictions. You can also create and consume your logistic regressions programmatically via the **BigML API and bindings**. The following subsections explain those three options.

## 4.7.1 Downloading Logistic Regressions

You can download your logistic regression in several programming languages including JSON PML, Python or Node.js. By downloading your logistic regression you will be able to compute **predictions locally**, free of latency and at no cost. Click on the download icon in the top menu (see Figure 4.132), and select your preferred option (see Figure 4.133.)



Figure 4.132: Click download icon

Complete actionable logistic regression:			
JSON PML × *			FA
JSON PML		0	
Python		·	3
Node.js			
"c": 1, "fields": { "000006": { "optype": "categ "name": "employn "datatype": "str "preferred": tru "term_analysis":	ment", ring", ue, : {		
"enabled": t	true		
		Close	

Figure 4.133: Select language to download logistic regression

## 4.7.2 Using Logistic Regressions Via the BigML API

Logistic regression have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them for predictions.

In the below example, see how to create a logistic regression using an existing dataset once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/logisticregression?\$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"dataset": "dataset/50650bdf3c19201b64000020"}'
```

For more information on using logistic regressions through the BigML API, please refer to the documentation¹⁸.

## 4.7.3 Using Logistic Regressions Via the BigML Bindings

You can also create and use logistic regressions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a logistic regression with the Python bindings.

```
from bigml.api import BigML
api = BigML()
logisticrregression = api.create_logistic_regression(
    'dataset/57506c472275c1666b004b10', {"objective_field":"churn"})
```

```
<sup>18</sup>https://bigml.com/api/logisticregressions
```

For more information on BigML bindings, please refer to the bindings page¹⁹.

## 4.8 Logistic Regression Limits

There are some limits that apply for the creation of any BigML resource. These are limits based on the number of classes, terms and items that can be considered to create your models. This is explained in Subsection 4.8.0.1.

Additionally, some specific limits apply for your logistic regressions **visualization**, i.e. to the logistic regression chart and the coefficient table views, depending on the number of classes in the objective field and the number of input fields in your dataset. See Subsection 4.8.2 and Subsection 4.8.1 for a detailed explanation.

Note: chart limits and coefficient table limits just affect to the visualization of the model, i.e., despite your dataset reach those limits, you can still creating the logistic regression, evaluating it and using it to make predictions.

#### 4.8.0.1 Field Limits

Logistic regression, similarly to other BigML models, has the following limitations according to the type of field:

- Classes: a maximum number of 1,000 distinct classes per field is allowed.
- **Terms**: BigML can handle up to 1,000 terms in total. If multiple text fields are defined, then the token limit per field is evenly divided by the number of text fields evenly, e.g., a dataset with two text fields would result in 500 terms per text field. BigML selects those terms with most significant frequency, discarding both those that appear either too often or too infrequently. A maximum of 256 characters per term is allowed.
- Items: a maximum number of 10,000 distinct items per field is allowed.

### 4.8.1 Chart Limits

There are some circumstances under which your chart cannot be displayed:

• As the chart only supports **numeric fields** for the x-axis, if your logistic regression **only** contains categorical, text, or items fields, the chart cannot be displayed. The view displayed by default will be the coeffcient table. When you try to click on the chart icon you will see a warning message. (See Figure 4.134.)

Sources Datasets	Logistic Regressions 👻	Clusters Anomal	ies Associations	Predictions Tasks	Scripts 🗸
	₩. •••• <u>/</u> L	oan risk data datas	et extended's logis	tic regr	<u>م</u> الم
Your logistic regression does not contain any numeric field to be included in the x-axis	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD 30 CLASSES	INTERCEPT SCALE 0	NORMALIZE FALSE
		Filter bias or predict	tors		<b>EV</b>
Bias and predictors	\$ Туре	✿ 10.80 <= duration	on < 13.07	<pre>7 &lt;= duration &lt; 15.33</pre>	✿ 15.33 <= duration < 17.60
Bias	123	0	0		0
checking_status = no checki	ABC	-0.18379	-0.03	752	-0.58011

Figure 4.134: Warning message when the logistic regression does not have any numeric field

¹⁹https://bigml.com/tools/bindings

• If your logistic regression contains more than **100 fields** the chart cannot be displayed and again the default view will be the table view. When you try to click on the chart icon, you will see the warning message shown in Figure 4.135.

Sources Datasets	Logistic Regressions 👻	Clusters Anomalies	Associations	Predictions Tasks	Scripts 👻
	🖏 •••• 🖉 An	rythmia diagnosis da	taset's logistic r	egressi	i 🗘 🕉 0
Your logistic regression exceeds the maximum number of fields (100)	DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE	OBJECTIVE FIELD	INTERCEPT SCALE 0	RORMALIZE FALSE
		Filter bias or predictors			CSV
Bias and predictors	Type	Atrial Fibrillat	on or Flutter 🗘	Ischemic changes (Coronar	ry Artery Di🃚 Left bundle branch
Bias	12	23 0		0	0
Age	1	0.00683		0.15744	-0.0036
Gender = Female	A	C 0.00158		-0.09933	0.00017
Gender = Male	AB	-0.00145		0.09504	-0.00042

Figure 4.135: Warning message when the logistic regression has more than 100 fields

• If your logistic regression contains more than **200 categories** in the objective field, neither the chart nor the table can be displayed and you will see the warning message shown in Figure 4.136. You can still see your coefficients by downloading the **CSV file**.

Sources Datasets Logistic Regressions	Clusters Anomalies Associations	Predictions Tasks	Scripts 👻
a s ^a ir 67	Loan risk datasets's logistic regress	sion	ф 🕉 🛈
DETAILS			$\overline{\mathbf{O}}$
INFO			$\overline{\mathbf{O}}$
PRIVACY			$\overline{\mathbf{O}}$
6,688 DEFAULT NUMERIC NOT DEFINED	MISSING NUMERICS TRUE ODJECTIVE FIELD STORE	0	NORMALIZE FALSE
VARIABL	Es: Filter bias or predictors		
Your model exceed	sorry but the table of coefficients for this model ca s the maximum number of classes (200) or the ma an download a file with the coefficients using the C	ximum number of fields (1,000).	

Figure 4.136: Warning message when the objective field has more than 200 classes

### 4.8.2 Coefficient Table Limits

If your logistic regression contains more than **1,000 fields** and/or more than **200 categories** in the objective field, the table cannot be displayed. You will need to **download the CSV** if you want to see your logistic regression coefficients. You will get the message shown in Figure 4.137:

	•••• <u>)</u>			n	- I
DETAILS					
INFO					
PRIVACY					
		S: Filter bias or predict	ors		
		sorry but the table of coeff the maximum number of		ot be displayed. 1um number of fields (1,000)	

Figure 4.137: Warning message when the table limits are reached

## 4.9 Descriptive Information

Each logistic regression has an associated **name**, **description**, **category**, and **tags**. The following subsections provide a brief description for each concept. In Figure 4.138, you can see the options the MORE INFO menu provides to edit them.

Sources Datasets	Logistic Regressions   Cluster	rs Anomalie	s Associations	Predictions T	asks	Scripts
l S Il	Loan risk	data datase	et extended's logi	stic regr	1	<b>5</b> (5- (
DETAILS						C
INFO						۹
Name: Loan risk data datase	et extended's logistic regression	1	Category: Miscellaneous			
Description:			Tags:			1
PRIVACY						



## 4.9.1 Logistic Regression Name

Each logistic regression has a name that is displayed in the logistic regression list view and also on the top bar of the logistic regression view. Logistic regression's names are indexed to be used in searches. When you create a logistic regression, it gets a default name. Change it using the MORE INFO menu option on the right corner of the logistic regression view. The name of a logistic regression cannot be longer than **256** characters. More than one logistic regression can have the same name even within the

same project, but they will always have different identifiers.

## 4.9.2 Description

Each logistic regression also has a **description** that it is very useful for documenting your Machine Learning projects. Logistic regressions take the description of the datasets used to create them by default.

Descriptions can be written using plain text and also markdown²⁰. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 4.139.)

Edit description	×
You can add formatting and links using a simple markdown language: B $I := \infty$	
Write your **description** here	
Description:	
Write your description here	
	Update

Figure 4.139: Markdown editor for logistic regression descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

### 4.9.3 Category

A **category**, taken from the dataset used to create it, is associated with each logistic regression. Categories are useful to classify logistic regressions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A logistic regression category must be one of the 24 categories listed on Table 4.5.

²⁰https://en.wikipedia.org/wiki/Markdown

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

### Table 4.5: Categories used to classify logistic regression by BigML

## 4.9.4 Tags

A logistic regression can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide logistic regressions with some extra information. A logistic regression inherits the tags from the dataset used to create it. Each tag is limited to a maximum of 128 characters. Each logistic regression can have up to 32 different tags.

## 4.9.5 Counters

For each logistic regression, BigML also stores a number of counters to track the number of other resources that have been created using the logistic regression as a starting point. In the logistic regression view, you can see a menu option that displays counters for evaluations, single and batch predictions, and the fusions created. It also allows you to quickly jump to all the resources of one type. (See Figure 4.140.)

INSTAN	EDICTIONS FROM THIS LOGISTIC REG		BIAS	FIELD CODINGS
	BATCH PREDICTION FROM THIS LOGI	DIABETES	YES	NO
	EVALUATION FROM THIS LOGISTIC RE			

Figure 4.140: Counters for logistic regressions

## 4.10 Logistic Regression Privacy

Privacy options for a logistic regression can be defined in the **More Info** panel, displayed in Figure 4.141. There are two levels of privacy for BigML logistic regressions:

- **Private**: only accessible by authorized users (the owner and those who have been granted access by him or her).
- Shared: accessible by any user with whom the owner shares the secret link.

ources	Datasets	Logistic Regres	sions 🔻	Clusters	Anomalies	Associations	Predictions	Tasks	s	cripts 👻
S	- Il	6m 0 - • • • •	Ch	urn in the	telecom ind	dustry's logis	tic regress	Ē	<b>\$</b>	6- 0
DETAILS										$\overline{\mathbf{O}}$
NFO										$\overline{\mathbf{O}}$
PRIVACY										۹
Private lin	k   Only you	can view this logistic r	egression lo	igged into you	ur account.					
			🔒 ht	tps://strato.	dev.bigml.com/	dashboard/logisti	cregression/57189	f06983efc543200	0033	
Secret link	k   Only peop	ole with the secret link (	can view thi	s logisticregre	ession.					
	_	3	<b>Γ</b> ≹ ht	tps://strato.	dev.bigml.com/	shared/logisticred	ression/qqXfFekfv	Cuuec2OYpYerJ	9411	

Figure 4.141: Logistic regression privacy

## 4.11 Moving Logistic Regressions to Another Project

When you create a logistic regression, it will be assigned to the same project where the original dataset is located.

Logistic regressions can only be assigned to a single project. However, you can move logistic regressions between projects. The menu option to do this can be found in two places:

1. In the logistic regression list view, click the MOVE TO... option within the **1-click action menu** and select another project or create a new one. (See Figure 4.142.)

Sources Datasets	Logistic Regressions	- Clusters Anomali	es Associati	ons	Predictions Tasks		Scripts -
a s ^a in f	8. ···· <u>)</u>	Loan risk data datas	et's logistic r	regre	ssion	<b>\$</b>	(≡) [/] - (j)
INSTANCES	DEFAULT NUMERIC	MISSING NUMERICS	OBJECTIVE	*	PREDICT		LIZE
1,000	NOT DEFINED	TRUE	CLASS	1	BATCH PREDICTION		E
					EVALUATE		
	VARIAE	Eilter bias or predict		1	DELETE LOGISTIC REGRESS	SION	CSV
Bias and predictors		NEW PROJECT			MOVE TO		
Bias					(	)	
checking_status = no checking	9	LOGISTIC REGRESSION PR		3119	1	1.13119	
checking_status = <0		ABC	0.53	3202		0.53202	

Figure 4.142: Change project from 1-click action menu

2. In the logistic regression list view, click the MOVE TO... option within the **pop up menu** and select another project or create a new one. (See Figure 4.143.)

Sources	Datasets	Logistic Regressions 🔻	Clusters An	nomalies Ass	ociations	Predictions	Tasks		Scri	ots 👻
			Log	jistic Regress	sions					Q
ill	Name		\$	Objective	\$	÷ #	<b>Å</b> \$	••• •	\$	<b>:</b> \$
ıl <mark>î</mark>	Loan risk data dat	aset extended's logistic regre	ession 💌	30 classes		45min	160.6 KB	0	0	0
ılı	Arrythmia diagnos			y otro		18h 52min	209.0 KB			
ılî	Loan risk data dat	CREATE BATCH PREC	DICTION	30 classes		18h 54min	160.6 KB			
ılî	Arrythmia diagnos	sis 📌 VIEW DETAILS		inventao		18h 56min	189.1 KB			
ılî	Loan risk datasets	BIN DELETE LOGISTIC RE	GRESSION	NEW PRO	VECT	tob Consin	°51.1 KB			
ılî	Arrythmia diagnos	SIS_TOO TIEIUS UALASEL S IOQISU	icregress	PROJECTS			90.0 KB			
ılî	Arrythmia diagnos	sis_100 fields dataset's logisti	ic regress		EGRESSION	PROJECT	90.0 KB			
ılı	Arrythmia diagnos	sis_100 fields dataset's logisti	ic regress	MY FIRST PF	ROJECT		эо.о кв			

Figure 4.143: Change project from pop up menu

## 4.12 Stopping Logistic Regressions

You can stop the creation of a logistic regression before the task is finished by clicking the DELETE LOGISTIC REGRESSION option from the **1-click action menu** (see Figure 4.144), or from the **pop up menu** in the logistic regression list view (see Figure 4.145).



Figure 4.144: Stop logistic regression creation from 1-click action menu

Sources	s Datasets	Logistic Regressions 👻	Clusters An	omalies Associatio	ons Prediction:	s Tasks		Scri	pts 👻
			Log	istic Regressions	;				Q
ill	Name		\$	Objective	≎ 🛗 ≎	<b>ê</b>	08 80 \$	\$	
ılî	Arrythmia diagnos	sis dataset's logistic regressio	on 🗵	Arrhythmia	2min	536.2 KB	0	0	0
ılî	Loan risk data data			class	1h 53min	135.8 KB			
ılî	US car accidents o	da CREATE BATCH PREC	NCTION	Injury Severity	4h 58min	741.0 KB			
ılî	Arrythmia diagnos			Arrhythmia	4h 58min	536.2 KB			
ılî	Titanic Survival da	nta Z DELETE LOGISTIC RE	GRESSION	Survived	4h 58min	75.3 KB			
ılî	Diabetis diagnosis	dataset s logistic regression	-	Diabetes	5h 3min	27.0 KB			
Show 10	logistic regress	ions	1 to 6 of	6 logistic regressions				I< < 1	> >

Figure 4.145: Stop logistic regression creation from pop up menu

A modal window will be displayed asking you for confirmation. If you stop the logistic regression during its creation you won't be able to resume the same task. If you want to create the same logistic regression, you will have to start a new task.

Sources Datasets Lo	Are you sure you want to delete this logistic regression?	
a s° n° 6.	If you delete this logistic regression, you will no longer have access to it and you will need to	
1,000	recreate it from your datasets.	NORMALIZE FALSE
	Cancel	GV
	◆ Type ◆ bad ◆	good 🗘

Figure 4.146: Confirmation message to delete a logistic regression

## 4.13 Deleting Logistic Regressions

You can delete your logistic regressions by clickin in the DELETE LOGISTIC REGRESSION option from the **1-click action menu** (see Figure 4.147) or using the **pop up menu** on the logistic regression list (see Figure 4.148).

Sources Datasets	Logistic Regressions 🔻	Clusters Anomalie	s Associations	Predictions Tasks	Scripts -
	₩. •••• <u>)</u> L	oan risk data datase	et's logistic regree	ssion 📋 🗘	∽ (≡)/- 0
INSTANCES	DEFAULT NUMERIC	MISSING NUMERICS	OBJECTIVE	PREDICT	LIZE
1,000	NOT DEFINED	TRUE		BATCH PREDICTION	E
				EVALUATE	
		Filter bias or predicto	rs 🗾 🚬	DELETE LOGISTIC REGRESSION	Csv
Bias and predictors		\$ Type	≎ bac	MOVE TO	\$
Bias		123	0	0	•
Dias		115	0	0	
checking_status = no checking	ng	ABC	-1.13119	1.13119	9

Figure 4.147: Delete logistic regression from 1-click action menu

Sources	B Datasets L	ogistic Regressions 👻	Clusters An	omalies Associa	ations	Predictions	Tasks		Scri	pts 👻
			Log	jistic Regressio	ns					0
ill	Name		\$	Objective	\$	<b>#</b> \$	<b>ê</b> \$	○ <b>8</b> ≎	\$	\$
ılî	Loan risk data datas	set's logistic regression	8	class		1h 47min	135.8 KB	0	0	0
ılî	US car accidents da			Injury Severity		4h 52min	741.0 KB			
ılî	Arrythmia diagnosis	CREATE BATCH PRED	ICTION	Arrhythmia		4h 52min	536.2 KB			
ılî	Titanic Survival data	VIEW DETAILS		Survived		4h 53min	75.3 KB			
ılî	Diabetis diagnosis (		GRESSION	Diabetes		4h 58min	27.0 KB			
show 10	ogistic regressio	MOVE TO	1 10 5 01	5 logistic regressions					K < 1	> >

Figure 4.148: Delete logistic regression from pop up menu

A modal window will be displayed asking you for confirmation. Once a logistic regression is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

Sources Datasets Lo		
a s' n' 67.	Are you sure you want to delete this logistic regression?	
INSTANCES [	If you delete this logistic regression, you will no longer have access to it and you will need to recreate it from your datasets.	NORMALIZE FALSE
	Cancel Delete	CSV
	Type 🔷 bad 🗘	good 🗘

Figure 4.149: Confirmation message to delete a logistic regression

## 4.14 Takeaways

This chapter explains logistic regressions in detail. Here is a list of key points:

- A logistic regression is a supervised Machine Learning algorithm used to solve classification problems.
- A logistic regression is built from a dataset available in BigML and used to make an evaluation, a single prediction, or a batch prediction. (See Figure 4.150.)
- You can create a logistic regression with just one click or configure it as you wish. BigML provides several **configuration** options before creating your logistic regression.
- To create a logistic regression you need a **dataset** containing at least one categorical field.
- · Categorical fields must be converted to numeric values in order to train a logistic regression model.
- If you do not specify any objective field, BigML will take the last valid field in your dataset.
- BigML allows you to include your numeric fields' missing values as valid values to train your logistic regression model.
- The chart view provides a visual way to analyze a field impact on predictions given certain values for the rest of the fields.
- You get all the objective field class probabilities along with the predicted class.
- BigML displays all your logistic regression coefficients in a **table view** which you can also download as a CSV file.
- You need to evaluate your logistic regression model's performance using data that the model has not seen before.
- The ultimate goal in building a logistic regression is being able to make predictions with it.
- BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the logistic regression, so you can easily set the input data and get an immediate response.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you
  need is the logistic regression you want to use to make predictions and a dataset containing the
  instances for which you want to obtain predictions.
- · You can configure your batch predictions output file settings.
- You can download your logistic regression to perform local predictions.
- You can add **descriptive information** to your logistic regressions (name, description, tags, and category).
- You can move your logistic regressions between projects.
- You can share your logistic regressions with other people using the secret link.
- You can stop your logistic regression creation by deleting them.
- You can permanently delete an existing logistic regression.



Figure 4.150: Logistic Regression Workflow



# **Deepnets**

## 5.1 Introduction

There are multiple Machine Learning problems that can be solved using supervised Machine Learning techniques. Some of these problems require to predict an output variable (objective field) given a number of input variables (input fields). These problems can be divided into classification and regression depending on whether you need to predict a category (label or class) or a continuous value (a real number), respectively. To learn more about concrete use cases for both problems refer to Section 1.1.

Deep neural networks (or deepnets) are a Machine Learning technique that can be used to solve classification and regression problems. These problems can also be solved with other Machine Learning methods, such as models, ensembles, or logistic regressions. These methods are explained in Chapter 1, Chapter 2, and Chapter 4 respectively. Depending on the problem you are trying to solve and the data available, some techniques may perform significantly better than others. See in Subsection 5.2.3 a detailed explanation of how deepnets perform compared to other supervised learning techniques for different use cases.

**Deepnets** are a class of machine learning models inspired by the neural circuitry of the human brain. See Section 5.2 for more details on the deepnets algorithm.

This chapter contains comprehensive description of BigML's deepnets including how they can be created with 1-click (Section 5.3), all configuration options available (Section 5.4), and the different visualizations provided by BigML (Section 5.5). See Section 5.6 for an explanation of how deepnets can be used to make predictions. You can also export your deepnets in different formats to make local predictions faster at no cost (Subsection 5.7.1). The process to evaluate your deepnets predictive performance in BigML is explained in a different chapter (Chapter 7).

In BigML, the third tab of the main menu of the Dashboard allows you to list all of your available deepnets. The deepnet list view (Figure 5.1), details the Dataset used to create it, the Name, Objective (objective field name), Age (time elapsed since it was created), Size, and number of evaluations, predictions, and batch predictions that have been created using that deepnet. The SEARCH menu option in the top right corner of the deepnet list view allows you to search your deepnets by name.
		×	Deepnets				Ô	Q
11	Name	\$ Туре	Objective	÷ #	<b>ê</b>	\$	\$	08
I <mark>P</mark>	FMEL_Dataset.csv [extended]   Training (80%) max training time=1800, missing values	o0	Result	5d 6h	2.2 MB	1		1
lî ^c	FMEL_Dataset.csv [extended]   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	o0	Result	5d 6h	2.2 MB			1
ľ	FMEL_Dataset.csv [extended]   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	o0	Result	5d 6h	2.2 MB			1
ľ	Forest Cover Type (training)   Training (80%) max training time=1800, missing values	o0	Cover_Type	5d 6h	1.5 MB			1
ľ	Forest Cover Type (training)   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	o0	Cover_Type	5d 7h	1.5 MB			1
ĥ	Churn in the telecom industry   Training (80%) v2 initial accumulator value=0.1, learning rate powe	o0	Churn	5d 7h	218.7 KB			
ĥ	Churn in the telecom industry   Training (80%) v1 initial accumulator value=0.1, learning rate powe	o0	Churn	5d 7h	218.7 KB			1
ĥ	Churn in the telecom industry   Test (20%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	o0	Churn	5d 7h	54.7 KB			
ĥ	Churn in the telecom industry   Training (80%) initial accumulator value=0.1, learning rate powe	o0	Churn	5d 7h	218.7 KB			1
ľ	Churn in the telecom industry   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	o0	Churn	5d 8h	218.7 KB			1

Figure 5.1: Deepnet list view

By default, when you first create an account at BigML, or every time that you start a new project, your list of deepnets will be empty. (See Figure 5.2.)

		X	Deepnets				Û	0
II	Name	\$ Туре	Objective	<b>***</b>	\$ <b>6</b>	\$ \$	\$	
		Ν	No deepnets					

Figure 5.2: Empty Dashboard deepnet view

Finally, in Figure 5.3 you can see the icon used to represent a deepnet.



Figure 5.3: Deepnet icon

# 5.2 Understanding Deepnets

A **deepnet** in BigML is a **supervised** learning method to solve **classification** and **regression** problems. Deepnets are an optimized version of **Deep Neural Networks**, a class of machine learning models inspired by the neural circuitry of the human brain. In these classifiers, the **input features** are fed to one or several groups of "**nodes**". Each group of nodes is called a "**layer**". Each node is essentially a function on the input that transforms the input features into another value or collection of values (see also Subsection 5.4.7.1). Then the entire layer transforms an input vector into a new "**intermediate**"

feature vector. This new vector is fed as input to another layer of nodes. This process continues layer by layer, until we reach the final "**output**", which is also a layer of nodes. The output is the network's prediction: an array of **per-class probabilities** for classification problems or a single, **real value** for regression problems.

The "deep" in deep neural networks refers to the presence of more than one "hidden layer"; that is, more than one layer of nodes between the **input** and the **output layers**. The network architectures supported by BigML can be deep or shallow. The advantage of training deep architectures is that hidden layers have the opportunity to learn "higher-level" representations of the data that can be used to make correct predictions in cases where a direct mapping between input and output is difficult. For example, when classifying images of numeric digits, the input layer is raw pixels, the output layer is the probability for each digit, and the intermediate layers may learn features that represent the presence of, say, a loop or a vertical stroke. Read Subsection 5.2.3 below for an explanation of the use cases where deepnets have a better performance.

### 5.2.1 Convolutional Neural Network

When the dataset used to create a deepnet contains images, the deepnet created will be a convolutional neural network.

Convolutional neural network, also known as CNN or ConvNet, is a type of deep neural networks. All deepnet operations described in this section, such as 1-click creation and configuration, also apply to CNNs. So does optimization, including automatic network search and structure suggestion.

The main difference between CNN and other types of neural networks is that the hidden layers of a CNN include at least one convolutional layer.

A convolutional layer performs one or more convolution operations. Each convolution operation transforms a set of neighboring inputs to an output, which is passed to the next layer. In the case of an image, the input layer consists of two-dimensional pixels. A convolution operation converts a block of pixels (say 3x3) to a single number. It can be imagined as a filter, sliding across the whole image and converting each (3x3) block of pixels to a number. The output of a convolutional layer can be thought as another image with a possibly decreased size and its pixels holding information from multiple (9) pixels of the image in the previous layer. The size of the pixel block (filter size) can be different (e.g. 3x3, 5x5), and one convolutional layer may have multiple convolution operations (number of filters).

There are other important layers in CNNs, such as ReLU and pooling layers. In the end when a CNN is fully trained, each convolutional layer, coupled with its ReLU and pooling layers, effectively captures image features. The first layers extract low-level features such as edges and colors while the deeper layers capture high-level features unique to the objects in the images. The outputs of convolutional layers are also called feature maps.

BigML can extract image features at the source level. For a composite source with images, different sets of image features can be extracted, which include features on edges, colors and texture. There are also pre-trained CNNs which capture more sophisticated features. Image features can be used to train supervised models as well as unsupervised models. If you know specific image features that will help you achieve your machine learning goals, you don't have to use CNNs. Instead, you can configure your sources to extract image features. For information about the image features, please refer to section Image Analysis of the Sources with the BigML Dashboard¹[11].

Note: When training a deepnet from a dataset containing images, that is, when training a CNN, all image feature fields extracted from the images will be ignored. In other words, when a dataset contains images, its deepnet is trained from raw image pixels, not from its extracted image features.

Because of convolution operations, which transform neighboring inputs such as 3x3 blocks, CNNs are excellent in machine learning using spatial data, especially images. However, CNNs may not work as well for non-spatial data such as tabular data. Think of this way: if many rows of a tabular data are swapped, the data is still considered the same. But doing the same to an image, it becomes a different image.

¹https://static.bigml.com/pdf/BigML_Sources.pdf

For a comprehensive introduction to CNN, please refer to its wikipedia entry².

## 5.2.2 Automatic Parameter Optimization

Deep neural networks are notoriously sensitive to the chosen topology (or network structure) and the algorithm used to learn the weights for that topology. This sensitivity means that **hand-tuning** the topology and optimization algorithm can be **difficult** and **time-consuming** as the number of choices that lead to poor networks typically vastly outnumber the choices that lead to good ones.

To combat this problem, BigML offers first-class support for **automatic parameter optimization** that allows for automated discovery of better networks via two different methods:

• Automatic network search: during the deepnet creation, BigML trains and evaluates over many possible network configurations, returning the best networks found for your problem. The final deepnet returned by the search is a "compromise" between the **top** "n" networks found in the search. The algorithm BigML uses for this optimization technique is a variant on the hyperband algorithm. Instead of selecting parameter value candidates for evaluation at random, however, BigML uses an acquisition technique based on techniques from **Bayesian parameter optimiza-tion**. The main downsides of using this optimization method is that the creation of the deepnet may be significantly slower.

Note: the search process is not totally deterministic, so although you are using the same dataset you might get slightly different results from run to run. This is because BigML trains multiple models concurrently and the order in which they finish is important. After each model finishes, the search modifies its behavior based on the performance of the one that just finished (i.e., the next trained model in the search depends on the previous ones). Although results may not be repeatable, the differences should be almost unperceivable in most cases.

• Automatic structure suggestion: BigML offers a faster technique that can also give quality results. The ability to quickly train and test your deepnets is especially useful when working on feature engineering. BigML has trained thousands of networks on dozens of datasets in order to understand the effectiveness of various network topologies. As such, BigML has learned some general rules about what makes one network structure better than another for a given dataset. BigML will automatically suggest a structure and set of parameter values that are likely to perform well for your dataset.

To learn more about these optimization techniques, read this blog post³.

You can choose either optimizing technique by selecting it in the **configuration panel** (see Subsection 5.4.2). Alternatively, you can manually set the parameters for your deepnet. By default, BigML uses the automatic structure suggestion strategy to create deepnets.

### 5.2.3 Deepnets Use Cases

A common question when solving classification and regression problems is which algorithm should be used to get the best results: models and ensembles? Logistic regressions? Deepnets? In most cases, there is not an effective way to know in advanced which method will perform better so the best strategy is training and evaluating each one of them and compare their performances. However, a general rule for deepnets is that they usually perform better with complex datasets and difficult problems. That is, **high-dimensional datasets** where either only a few features are non-noise, or where the **decision function** is spread across **many different features**.

On one hand, **decision trees and ensembles** have spectacular representational power when the dataset has a high number of variables because their hypothesis space grows with the data and the decision tree algorithm is able to efficiently search through the space to get a good solution in reasonable time. However, trees have trouble representing objectives that are smooth functions of lots of variables.

²https://en.wikipedia.org/wiki/Convolutional_neural_network

³https://blog.bigml.com/2017/10/04/deepnets-behind-the-scenes/

On the other hand, **logistic regression** optimizes a function that takes into account all of the variables at once, not just one or two at a time, and is able to optimize this function efficiently. However, their representational power is considerably lower; they can only represent linear decision boundaries.

Deepnets try to get the best of both methods. Because their structure is very flexible, they have **high representational power**, and because they're optimized via gradient descent (like logistic regression), they do fine with **smooth functions of potentially all of the input variables**.

The main **downside** of deepnets is the **efficiency** that both decision trees and logistic regression provide. The structure of deepnets is super-flexible, but there is no way to search through the possible structures and parameters as quickly as can be done for trees or logistic regression. The only way to find an optimal structure is to try a lot of them. BigML tries to make this search as clever as possible (see <u>Subsection 5.2.2</u>), but it is still significantly more time-consuming than trees and logistic regression with no guarantee that it will beat them.

## 5.2.4 Missing Values

BigML deepnets can handle missing values for any type of field. For categorical, text, and items fields, missing values are always included by default.

For numeric fields, missing values are also included by default, but you can deactivate this option by configuring your deepnet (see Subsection 5.4.4). If the missing numeric option is disabled, the instances containing missing values for numeric fields in your dataset will be ignored by the deepnet. Also when using your deepnet to make predictions, you will not be able to have missing values for the numeric fields in the input data.

# 5.3 Creating Deepnets with 1-Click

To create a deepnet in BigML you have two options: either the 1-click option which uses the default values for all available configuration options or you can tune the parameters in advanced by using the configuration options explained in Section 5.4. This section explains how to create a deepnet with 1-click.

From the dataset view, select the 1-CLICK DEEPNET option in the 1-click action menu. (See Figure 5.4.)

Sources Datasets S	Unsupervised - Unsupervise	d 🔻 Predicti	ons Tasks	ŝ	WhizzML 👻
a (4) 17, 10	-   •••• []] Dig	git recognize	er (train)   Tra	ining (80%)	2 2 (≡) ⁵ / ₂ ()
				1-CLICK MODEL	36
Name	\$ Туре	≎ Count	Missing	1-CLICK ENSEMBLE	
				1-CLICK LOGISTIC REGRESSION	
pixel39	123	33,600	0	1-CLICK DEEPNET	NEW
pixel40	123	33,600	0	1-CLICK TIME SERIES	
		00,000	,	1-CLICK CLUSTER	
pixel41	123	33,600	0	1-CLICK ANOMALY	
				1-CLICK ASSOCIATION	
pixel42	123	33,600	0	1-CLICK TOPIC MODEL	
pixel43	123	33,600	0	SPLIT DATASET	
pixel+0	120	55,000	0	REQUEST EXPORT (CSV)	
pixel44	123	33,600	0	REQUEST EXPORT (TABLEAU)	
				DELETE DATASET	
pixel45	123	33,600	0	MOVE TO	
pixel46	123	33,600	0	0	

Figure 5.4: Create 1-click deepnet from dataset 1-click action menu

Alternatively, you can use the 1-CLICK DEEPNET option in the **pop up** menu from the dataset list view. (See Figure 5.5.)

				atasets								Ô	Q
							digi					×	0
9	Name	\$	⇒	<b>Å</b> •	\$î.º	£©¢	<u>∕</u> ≎	≋≎	<b>~~</b> \$	16 ¢	° <mark>%</mark> a,≎	₽\$	<u>}</u> ;
ılî	Digit recognizer (train)   Test (20%) 8400 instances, 785 fields (131 categorical, 65		1w	14.6 MB									
ılî	Digit recognizer (train)   Training (80%) 33600 instances, 785 fields (131 categorical, 6		1w	58.6 MB	0	1	1	4	0	0	0	0	0
54	Digit reco		1w	73.2 MB									
s°.	Digit reco         1-CLICK ENSEMBLE           42000 inst         1-CLICK LOGISTIC REGRESS	SION	1w	73.2 MB									
Show	10 🔉 dat 🎤 1-CLICK TIME SERIES		of 4 datasets (filter	ed from 115 tot	al datase	ts)					1< <	1 >	×

Figure 5.5: Create 1-click deepnet from dataset popup menu

Either option builds a deepnet using the default values for all available configuration options. (See Section 5.4.)

Note: your dataset needs to contain at least one categorical or numeric field to be selected as the objective field to create the deepnet.

# 5.4 Deepnet Configuration Options

While the 1-click creation menu option (see Section 5.3) provides a convenient and easy way to create a BigML deepnet, you can also have more control over the deepnet creation and configure a number of parameters that affect the way BigML creates deepnets. Click the CONFIGURE DEEPNET menu option in the **configuration menu** of your dataset view. (See Figure 5.6.)

Sources Datasets	NEW) Supervised ✓ Unsupervise	ed - Predict	ions 🕶 Ta	asks			WhizzML	•
ם (א <mark>י</mark> ) די	ញ្ចែ, •••• []] Di	git recogniz	er (train)	Training (80%)	° <b>⊘</b> -	∽.	(≡) ^{\$} -	(
Name	≎ Туре	≎ Count	ດ ≎ № ⁶ 6					×
pixel39	123	33,600	0	CONFIGURE DEEPNET	NEW			
pixel40	123	33,600	0	CONFIGURE TIME SERIES				
pixel41	123	33,600	U	CONFIGURE ANOMALY				
pixel42	123	33,600		CONFIGURE TOPIC MODEL				
pixel43	123	33,600						
pixel44	123	33,600	。 <b>I</b>	FILTER DATASET ADD FIELDS TO DATASET				
pixel45	123	33,600	0	0				

Figure 5.6: Configure deepnet

# 5.4.1 Objective Field

The objective field, or "target field", is the field you want to predict. Deepnets support **categorical or numeric** fields as the objective field.

BigML takes the **last valid field** in your dataset as the objective field by default. If you want to change the objective field, you have two options: you can select another field from the configuration panel to build the deepnet, or you can change it permanently from your dataset view.

• Select the **Objective field** from the deepnet **configuration panel**. This option will only affect the deepnet you are building that time. (See Figure 5.7.)

Sources Datasets Supervised - Unsupervi	ised ▼ Predictions ▼ Tasks		WhizzML 🔻
읍 (\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	abetes diagnosis dataset   Training (	(80%) 🍖 🍊	(≡)∱ (i
DEEPNET CONFIGURATION			POT
Objective field: Diabetes ABC	Automatic optimization:		0
Default numeric value:     Missing numerics:       Select a default value <ul> <li>N/A</li> </ul>	Max. training time:           00:30:00         O	Max. iterations: 20000	0
Advanced configuration			◙
Deepnet name:			
Diabetes diagnosis dataset   Training (80%)		Reset Create	edeepnet

Figure 5.7: Configure the objective field to create the deepnet

• Change the **default objective field** for the dataset. This option will save your objective field preference for any model you build. Click on the edition icon next to the field name when you mouse over it, a pop up window will be displayed. Then click on the Objective field icon and Save it. (See Figure 5.8.)

Sources Datasets Super	vised  Vised Visupervised  Vised Vis	5	WhizzML 🔻
(1)% <b>↓↓</b>	••• Diabetes diagnosis dataset	Training (80%)	\$\$\$ (≣)⁵- ()
	Update field 000002 details	C	X X
Name		Serrors Serrors	
Pregnancies	Name		
Glucose	Blood pressure Label	الس	1
Blood pressure	Description	0 	dillition
Skinfold	Description	o <b>hunil</b> i	111111111111111111111111111111111111111
Insulin		• <b>•</b>	
ВМІ	This field is not the objective field	اللية. ٥	<b>II</b> III
Diabetes pedigree	Cancel Save	o dilititu	
Age	<b>123</b> 614 0	• <b>111</b> 11111	H

Figure 5.8: Change the default objective field

# 5.4.2 Automatic Parameter Optimization

The high number of configurable parameters for neural networks makes it difficult to find the optimum configuration to get good results. Hand-tuning different configurations is a time-consuming process in

which the combinations that lead to a poor result outnumber the ones that result in a satisfying performance. To combat this problem, BigML offers first-class support for **automatic parameter optimization** via two different methods:

• Automatic network search: during the deepnet creation, BigML trains and evaluates over many possible network configurations, returning the best networks found for your problem. The final deepnet returned by the search is a "compromise" between the **top** "n" networks found in the search. The main problem of using this optimization method is that the creation of the deepnet may be significantly slower.

Note: the search process is not totally deterministic, so although you are using the same dataset you might get slightly different results from run to run.

• Automatic structure suggestion: BigML offers a faster technique that can also give quality results. BigML has learned some general rules about what makes one network structure better than another for a given dataset. BigML will automatically suggest a structure and a set of parameter values that are likely to perform well for your dataset.

Read more about the automatic parameter optimization in Subsection 5.2.2. You can choose either optimizing technique by selecting it in the **configuration panel** (see Figure 5.9).

Sources Datasets Supervised - Unsupervised -	NEW Predictions		WhizzML 🔻
באין אין אין אין אין אין אין אין אין אין	diagnosis dataset   Training (8	80%) 🍖 🍊	(≡)⁵, (≣
DEEPNET CONFIGURATION			POT
Dijective field: Diabetes ABC v	Automatic optimization:	ך	0
Default numeric value:     Missing numerics:       Select a default value <ul> <li>N/A</li> </ul>	Max. training time: 00:30:00 ©	Max. iterations:	0
್ಲಿ Advanced configuration			◙
Deepnet name:			
Diabetes diagnosis dataset   Training (80%)		Reset 🥂 Create	e deepnet

Figure 5.9: Select an automatic optimization option

When you select an optimization strategy, the **Network architecture** parameters (see Subsection 5.4.7), the **Algorithm** parameters (see Subsection 5.4.8) and the **Weights** (see Subsection 5.4.9) will be automatically set. You cannot manually tune any of them except the **Weights** which you can manually configure it. If you want to configure the rest, you need to deactivate the automatic optimization options using the switcher as shown in Figure 5.10,

Sources Datasets Super	vised  Vi	NEW Predictions ▼ Tasks		WhizzML 🔻
	Diabetes	diagnosis dataset   Training (8	30%) 🍖 🍏	
DEEPNET CONFIGURATION				POT
Objective field: Diabetes	ABC 🔻	Automatic optimization:	ך	0
Default numeric value: Select a default value	Missing numerics:	Max. training time: 00:30:00 O	Max. iterations:	•
k Advanced configuration				$\overline{\mathbf{O}}$
Deepnet name:				
Diabetes diagnosis dataset   Tra	aining (80%)		Reset Creat	e deepnet

Figure 5.10: Disable automatic optimization options to manually configure the rest of the network parameters

## 5.4.3 Default Numeric Value

Deepnets can include missing values as valid values for any type of fields as explained in Subsection 5.2.4. However, there can be situations for which you do not want to include them in your model. For those cases, the **default numeric value** parameter is an easy way to replace missing numeric values by another valid value. You can select to replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 5.11.)

Sources Datasets Supervised - Unsupervised -	Predictions Tasks		WhizzML 🔻
	diagnosis dataset   Training (	(80%) 🍖 🍊	(≡)∳, (i
DEEPNET CONFIGURATION			POF
Objective field:	Automatic optimization:		0
Diabetes ABC v	**		
Default numeric value: Select a default value	Max. training time: 00:30:00 O	Max. iterations: 20000	0
Advanced configuration			$\overline{\mathbf{O}}$
Deepnet name:			
Diabetes diagnosis dataset   Training (80%)		Reset Create	deepnet

Figure 5.11: Select a default numeric value to replace missing numeric values

Note: if your dataset does not contain missing values for your numeric fields, this parameter will not have impact on your deepnet. If your dataset contains missing numeric values and you neither select a default numeric value or enable the missing numerics configuration option, instances with missing numeric values will be ignored to build the model.

### 5.4.4 Missing Numerics

By default, missing values for your numeric fields are included as valid values to build your deepnets. However, there can be cases for which you do not want them to be included in your model. The **Missing numerics** option allows you to select if you want to include or exclude the missing numeric values to build your deepnets. (See Figure 5.12.)

Sources Datasets Supervised - Unsupervised -	Predictions Tasks		٧	VhizzML 🔻
읍 (배) 토고 탑승 ···· Diabete	es diagnosis dataset   Training	(80%)	. 3- (	(≡) <b>*</b> _ (i)
DEEPNET CONFIGURATION				109
Objective field: Diabetes ABC v	Automatic optimization:	:		0
Default numeric value: Select a default value	Max. training time: 00:30:00 O	Max. iterations	20000	0
Advanced configuration				◙
Deepnet name:				
Diabetes diagnosis dataset   Training (80%)		Reset	Create de	epnet

Figure 5.12: Include missing numeric values in your deepnet

Note: missing values are always included for categorical, text, and items fields. If your dataset contains missing numeric values and you do not either select the missing numeric option or set a default numeric value (see Subsection 5.4.3), instances containing missing values will be ignored when building the deepnet.

## 5.4.5 Training Duration

The scale parameter to regulate the deepnet runtime. It's set as an integer from 1 to 10. It indicates the user preference for the amount of time they wish the optimization to take. The higher the number, the more time that users are willing to wait for possibly better deepnet performance. The lower the number, the faster that users wish the deepnet training to finish. The default value is set to 5.

The training duration is set in a scale. The actual training time depends on the dataset size, among other factors.

Sources Datasets Supervised - Unsupervised -	Predictions -	Tasks WhizzML <del>-</del>
· ····	diabetes	ıl°∼ 🍫 🏷 (≡)⁵~ ()
DEEPNET CONFIGURATION		Por
Objective field: diabetes	Automatic of the second	optimization:
Default regions value:     Default numeric value:       Not available <ul> <li>Select a default value</li> <li> </li></ul>	Missing numerics:	Training duration: 5 Max. iterations: Auto 4 4 4 4 4 4 4 4 4 4 4 4 4
Advanced configuration		$\odot$
Deepnet name: diabetes		Reset Create deepnet

Figure 5.13: Set the training duration for a deepnet

# 5.4.6 Maximum Iterations

The number of iterations in a deepnet is the **number of gradient steps** the algorithm takes during the optimization process. You can set the maximum number of iterations to train your deepnet by activating this option using the switcher (see Figure 5.14). By default, this option will be deactivated, in which case

BigML will stop training the network if a certain number of iterations goes by without substantial progress or if it reaches the limit of the training duration (see <u>Subsection 5.4.5</u>).

Sources Datasets Supervised - Unsupervised	d 🕶 Predictions 👻 Ta	isks	WhizzML 🔻
a 17. 19	diabetes		:_ (≝)⁵- (≣)
DEEPNET CONFIGURATION			POF
Objective field: diabetes ABC v	Automatic opti	mization: Number of networks	s evaluated:
Default regions value:     Default numeric value:       Not available     •         Select a default value	Missing numerics:	Training duration:	Max. iterations:
Advanced configuration			$\overline{\mathbf{O}}$
Deepnet name: diabetes		Reset 2	Create deepnet

Figure 5.14: Set the maximum number of iterations for a deepnet

# 5.4.7 Network Architecture

The basic architecture specification for a network consists of specifying the **number of hidden layers** in the network, the **number of nodes** in each layer, the **activation function**, and other parameters related to how the network connections are arranged such as **residual learning**, **batch normalization** and **tree embedding**.

### 5.4.7.1 Hidden Layers

The hidden layers in a neural network are the intermediate layers between the input layer (the one containing the input field values) and the output layer (the one containing the predictions). (See Section 5.2). In BigML you can configure **up to 32 hidden layers**. For each layer you can specify the following parameters:

• The activation function: by applying an activation function, the deepnet is able to represent a non-linear mapping between the inputs and the outputs which is necessary to solve complex problems. The activation function converts an input into an output that is used as an input again to feed the next layer in the network and so on. If no activation function is applied, the output will be simply a linear function of the inputs. In BigML you can select one of the following functions: "Tanh"⁴, "Sigmoid"⁵, "Softplus"⁶, "Softmax"⁷, "ReLU"⁸ or "None". If "None" is selected, no activation function will be set for the layers so the raw output values for each node will be used as inputs for the next layer.

To know more about each type of activation function please refer to this article⁹

- The **number of nodes**: each hidden layer in the network can have a variable number of nodes. Determining the optimal number of hidden nodes (also called hidden units or neurons) per layer is a complex task depending on:
  - The number of nodes of the **input** and **output layers**. The number of nodes in the input layer is equal to the number of fields of the dataset used to create the deepnet. The output

⁴https://en.wikipedia.org/wiki/Hyperbolic_function

⁵https://en.wikipedia.org/wiki/Sigmoid_function

⁶https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

⁷https://en.wikipedia.org/wiki/Softmax_function

⁸https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

⁹https://medium.com/towards-data-science/activation-functions-and-its-types-which-is-better-a9a5310cc8f

layer always has one node if it is a regression problem or as many nodes as classes has the objective field if it is a classification problem.

- The number of **instances** in the training dataset.
- The **complexity** of the problem that is trying to be solved.
- The gradient descent algorithm (see Subsection 5.4.8.1) and the activation function used.

The higher the number of nodes, the higher the risk of overfitting. However, too few nodes may lead to a poor solution if the function to be learned has some complexity. In most cases, you will need to try different size for the layers or use one of the BigML optimization options (see Subsection 5.4.2) to reach a number of nodes per layer that provide satisfying results. In BigML you can set **up to 8,192 nodes** per layer. A rule of thumb to determine the size of the layers is that the number of hidden nodes should be somewhere in between the sizes of the input and the output layers. Also, the size of the hidden layer should not exceed twice the size of the input layer because at this point it is very likely to overfit. Read this article¹⁰ to know more about the variables that should be taken into account when deciding the nsize of the layers.

The weights for each layer are initialized randomly according to Xavier's method¹¹.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the hidden layers will be automatically set. If you **disable the automatic options**, you can add and remove layers, select the activation function and the number of nodes for each layer (see Figure 5.15). You can add a minimum of one layer up to 32 layers.

읍 (배) 토콜- 웹 · · · · · III Diabetes diagnosis dataset   Training (80%)	🎭 🝊- (≡)	۶.
DEEPNET CONFIGURATION		POT
Objective field: Diabetes ABC		0
Default numeric value:     Missing numerics:     Max. training time:       Select a default value     N/A     00:30:00     O	terations: 20000	0
Advanced configuration		۲
Network Architecture:		۲
Hidden layers: Tanh   Number of nodes: Sigmoid  Number of nodes: ReLU  Number of nodes: Hidden layer Mumber of nodes: Hidden layer Hidden layer Mumber of nodes: Hidden layer Hidden layer	HIDDEN LAYERS	0
No learn residuals:	LEARN RESIDUALS	Ø
No batch normalization: এট্র	BATCH NORMALIZATION	0
No tree embedding:	TREE EMBEDDING	0

Figure 5.15: Configure the hidden layers of the network

¹⁰http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html

¹¹http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

#### 5.4.7.2 Learn Residuals

If **learning residuals** is enabled, it will cause alternate layers to learn a representation of the residuals for a given layer rather than the layer itself, by introducing shortcut connections. In other words, residual networks tweak the mathematical formula of the typical layer's equation to include the inputs of a lower layer in a node of a higher layer. Residual learning has proved to be very successful for image recognition as described in this paper¹².

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the residual learning will be automatically set. If you **disable the automatic options**, you can choose to include or exclude it (see Figure 5.16).

EPNET CONFIGURATION			
ective field: abetes ABC v	Automatic optimization:		
ault numeric value: Missing numerics: lect a default value V/A	Max. training time:           00:30:00         O	Max. iterations:	
Advanced configuration			
Network Architecture:			۹
Hidden layers:			
Tanh   Number of nodes:	64		0
+Add layer		MANUAL	J
No learn residuals:			Ø
		No	
<b>\$</b>			
No batch normalization:			0
		BATCH NORMALIZATION	0
No batch normalization:			0

Figure 5.16: Enable or disable the residuals learning

#### 5.4.7.3 Batch Normalization

If **batch normalization** is enabled, it will cause the outputs of a network to be normalized before being passed to the activation function, as described in this paper¹³. This will introduce extra parameters in each layer (the mean, variance, and scale of the layer), and will significantly slow down training.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the batch normalization will be automatically set. If you **disable the automatic options**, you can choose to include or exclude it (see Figure 5.17).

¹²https://arxiv.org/abs/1512.03385

¹³https://arxiv.org/abs/1502.03167

ⓐ (\\\\\} "₽- <b>\ </b> " •••• <b>    </b> Diabete	s diagnosis dataset   Training (80%)	🍫 - (≡	)*_
EEPNET CONFIGURATION			
bjective field:	Automatic optimization:		
Diabetes ABC V	N. 28		
efault numeric value: Missing numerics: Select a default value V/A	Max. training time:           00:30:00	Max. iterations:	
Advanced configuration			(
Network Architecture:			۵
Hidden layers:	64		0
Tanh   Number of nodes:		MANUAL	
+Add layer		LEARN RESIDUALS	0
No learn residuals.		No	
8		110	
No batch normalization:		BATCH NORMALIZATION	Ø
			0
No batch normalization:		BATCH NORMALIZATION	0

Figure 5.17: Enable or disable the batch normalization

### 5.4.7.4 Tree Embedding

If **tree mebedding** is enabled, the network will learn a tree-based representation of the data as engineered features along with the raw features, essentially by learning trees over slices of the input space and a small amount of the training data. The theory is that these engineered features will linearize obvious non-linear dependencies before training begins accelerating the learning process.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the tree embedding will be automatically set. If you **disable the automatic options**, you can choose to include or exclude it (see Figure 5.18).

	Diabete	es diagnosis dataset   Training (80	%) 🍖 🕉	(≡) <mark>*</mark> -
EEPNET CONFIGURATION				
bjective field: Diabetes	ABC	Automatic optimization:		
efault numeric value:	Missing numerics:	Max. training time:	Max. iterations:	
Advanced configuratio	on			
Network Architecture:				۲
Hidden lavers:				
Hidden layers:	Number of nodes:	64		Ø
Tanh	Number of nodes:	64	HIDDEN LAYERS MANUAL	0
Tanh   Add layer	Number of nodes:	64		
Tanh   Add layer  No learn residuals:	Number of nodes: 💮	64	MANUAL	
Tanh   Add layer  No learn residuals:	0	64	MANUAL LEARN RESIDUALS	0
Tanh  + Add layer No learn residuals:	0	64	MANUAL LEARN RESIDUALS No	0
Tanh       + Add layer       No learn residuals:	0	64	MANUAL LEARN RESIDUALS No BATCH NORMALIZATI	

Figure 5.18: Enable or disable the tree embedding

To learn more about the residual learning, the batch normalization and the tree embedding, read this blog post¹⁴.

# 5.4.8 Algorithm

BigML deepnets allow you to select different gradient descent algorithms to optimize the network weights in order to minimize the loss function. These algorithms have some specific parameters explained in Subsection 5.4.8.1 and also some common ones such as the **learning rate**, the **dropout rate**, and the **seed** described in Subsection 5.4.8.1, Subsection 5.4.8.3, and Subsection 5.4.8.4, respectively.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the algorithm parameters will be automatically set. If you **disable the automatic options**, you can configure them.

### 5.4.8.1 Gradient Descent Algorithm

The most widely used algorithm during the neural networks training is the gradient descent algorithm¹⁵. This optimization algorithm is used to minimize the loss function. Although the gradient descent is the most important and popular technique used to train neural networks, it presents some problems associated such as converging to a sub-optimal local minimma or setting a proper learning rate (not too small to avoid slow convergence and not too big to avoid divergence). To further optimize the gradient descent, BigML offers several optimization algorithms:

- **Momentum**: this method helps accelerate the gradient descent in the right direction dampening the oscillations until convergence. Therefore, it usually leads to faster and stable converge by reducing the unnecessary parameter updates. The configurable parameters for this algorithm include:
  - Momentum: higher values accelerate the gradient descent.

¹⁴https://blog.bigml.com/2017/10/04/deepnets-behind-the-scenes/

¹⁵https://en.wikipedia.org/wiki/Gradient_descent

betes		Automatic optimiz	zation:		
	ABC	No.			
ect a default value	Missing numerics:	Max. training time: 00:30:00 O	C Max.	iterations: 20000	
Advanced configuration					
Network Architecture:					⊽
Algorithm:					۹
Gradient Descent Algo	rithm: Seed:				0
Momentum	<b>v</b>			MOMENTUM	
	Dropout rate	9:			0
Learning rate:	0.401	50%	0.1%	50%	
Learning rate:	0.1%				
Learning rate:	0.1%	<u></u>			0

Figure 5.19: Momentum algorithm

However, this method does not solve the problem that the final performance heavily depends on the selected learning rate.

- Adagrad: is an adaptative learning method. Essentially, it adapts the learning rate to each parameter making big updates for infrequent parameters and small updates for frequent parameters. It solves the problem of selecting a unique learning rate since it can take a default rate and then adapt it for each parameter. This method works very well with sparse data. The configurable parameters for this algorithm include:
  - Initial Accumulator Value: this is the initial value for the gradient accumulator.

	•••• Diabete	es diagnosis dataset   Tra	ining (80%)	°⊗_ (∋	) <del>*</del> _
PNET CONFIGURATION					
ective field:	ABC	Automatic optimi	ization:		
ault numeric value: ect a default value	Missing numerics:	Max. training time	: Constant	iterations: 20000	
Advanced configuration	1				
Network Architecture:					َ⊘
Algorithm:					۹
Gradient Descent Algo	rithm: Seed:				0
Adagrad	·			ADAGRAD	
Learning rate:	Dropout ra	te:			0
0	0.1%	50%	0.1%	50%	
Initial Accumulator Val	ue:	0			0
Ø	0.01			0.01	
Weights:					َ⊘
Sampling:				614 instances	$\overline{\mathbf{r}}$

Figure 5.20: Adagrad algorithm

The problem of Adagrad is that the learning rate tends to decay to a very small number so the learning stops. RMSProp tries to solve this problem.

- **RMSProp**: is another adaptative learning method that can be considered an extension of Adagrad and tries to solve the problem of decaying rates. The configurable parameters for this algorithm include:
  - Momentum: higher values accelerate the gradient descent.
  - Decay: the speed to decay the moving average.
  - **Epsilon**: a parameter to avoid numeric precision problems.

	••••	Diabetes	diagnosis dataset   1	fraining (80%)	🎭 ∽ (	≡)*_ (
EPNET CONFIGURATION						
abetes		ABC	Automatic opt	imization:		(
ault numeric value:	Missing n	umerics:	Max. training tir 00:30:00 O	me: C Max.	iterations:	(
Advanced configuratio	n					e
Network Architecture:						◙
Algorithm:						۲
Gradient Descent Alg	orithm:	Seed:				0
RMS Prop	•				RMS PROP	
Learning rate:		Dropout rate:				0
<b></b>	0.1%		50%	0.1%	50%	
Momentum:		Decay:				0
	0.99	0	0	0.99	0	
						0
Epsilon:						

Figure 5.21: RMSProp algorithm

- Adam: (Adaptative Moment Estimation) is another adaptative method that computes adaptative learning rates for each parameter like Adagrad and it also solves the problem about decaying rates like RMSProp. Moreover, it keeps an exponentially decaying average of past gradients like Momentum. Adam usually works well compared to other algorithms as it converges fast and it solves the problems that other algorithms may have. The configurable parameters for this algorithm include:
  - Beta1: decay rate for the first moment estimate (the mean).
  - Beta2: decay rate for the second moment estimate (the variance).
  - Epsilon: a parameter to avoid numeric precision problems.

	•••• Diab	etes diagnosis dataset   Trai	ining (00 /0)	<b>*</b> ∽ (5- (	=)' +
EPNET CONFIGURATION					
jective field:		Automatic optimiz	zation:		
abetes	ABC	•			
ault numeric value:	Missing numerics:	Max. training time:	Max.		
elect a default value	N/A	00:30:00 ©	-0-	20000	
a Advanced configuration					(
M ravallood configuration					
Network Architecture:					ਂ
Algorithm:					۲
					0
Gradient Descent Algor				ALGORITHM	0
Gradient Descent Algor Adam	rithm: Seed:			ADAM	
	Dropout			ADAM DROPOUT RATE	0
Adam	<b></b>	t rate:	LEARNING RATE 0.1%	ADAM	
Adam	Dropout			ADAM DROPOUT RATE	
Adam Learning rate:	0.1%		0.1%	ADAM DROPOUT RATE 50%	0
Adam Learning rate:	Dropout	50%	0.1%	ADAM DROPOUT RATE 50% BETA2 0.999	0
Adam Learning rate:	0.1% Dropout	50%	0.1%	ADAM DROPOUT RATE 50% BETA2 0.999 EPSILON	0
Adam Learning rate:	Dropout	50%	0.1%	ADAM DROPOUT RATE 50% BETA2 0.999	0

Figure 5.22: Adam algorithm

- **FTRL**: it also adapts the learning rate by slowing the learning rate per prameter. The configurable parameters for this algorithm include:
  - Regularization: the regularization factor to avoid overfitting, i.e., tailoring the model to the training data at the expense of generalization. You can choose between L1 or L2 regularization.
  - Strength: is the inverse of the regularization strength, so higher values indicate less regularization. It must be a positive integer greater than 0. Too high values for strength will make the algorithm perfectly fit the training data boundaries. Too low values for strength will result in vague decision boundaries not following the data patterns.
  - Learning rate power: the learning rate power for the FTRL algorithm.
  - Initial Accumulator Value: this is the initial value for the gradient accumulator.

(
$\overline{\mathbf{O}}$
۹
0
0
0
0

Figure 5.23: FTRL algorithm

In summary, if your data is sparse, some of the **adaptative algorithms** may perform better. Adagrad, RMSProp and Adam are quite similar and perform well for similar use cases. However, **Adam** is the one that usually outperforms the rest due to its bias correction.

eRgarding the **algorithm-specific parameters** (momentum, beta1 and beta2, accumulator values, learning rate power, etc.), they all offer similar ways of controlling how much gradient descent remembers previous iterations and uses those to inform the current gradient step. Tuning those parameters have a **similar impact**: too high values for this sort of correction will send the search zooming off in the wrong direction; too low values will result in the same problems as vanilla gradient descent (overfitting and getting stuck in a local minima). If these parameters are set just right, they **improve the speed** at which the algorithm converges, and helps it to **avoid local minima**.

For all these parameters, though, the most important rule is not to hand-tune and iterate them unless you have a specific reason to do it. The best values for them depend on your data, the topology of your network, and the random conditions you start in. Hence, the best option if you are not very experienced with neural networks is to use one of the **BigML optimization options** (see <u>Subsection 5.4.2</u>) which will find the best configuration for your network automatically.

To learn more about the optimization algorithms please refer to this article¹⁶

### 5.4.8.2 Learning Rate

The **learning rate**, also known as the gradient step, controls how aggressively the gradient descent algorithm fits the training data. You can set values greater than 0% and smaller than 100%. Larger values will prevent overfitting, but smaller values generally work better (usually 1% or lower), although it usually takes longer to train the deepnet. As a general rule, you want to find a learning rate that is low enough so the network converges to a satisfying solution, but high enough to reduce as much as possible the training time.

¹⁶https://medium.com/towards-data-science/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the learning rate will be automatically set. If you **disable the automatic options**, you can select a value for the learning rate (see Figure 5.24).

EEPNET CONFIGURATION						
bjective field:			Automatic optim	nization:		(
Diabetes		ABC	***			
efault numeric value:	Missing nu	umerics:	Max. training time	e: 🕢 Max.	iterations:	(
Select a default value	× N/A		00:30:00 ©	-0-	20000	
Ö						0
Advanced configurat	ion					(
Network Architecture:						$\overline{\mathbf{O}}$
Network Architecture:						$\circ$
Algorithm:						۵
						۲
		Seed:				-
Algorithm:		Seed:			ALGORITHM ADAM	۲
Algorithm: Gradient Descent A	lgorithm:	Seed:				۲
Algorithm: Gradient Descent Al Adam	lgorithm:		50%	LEARNING RATE 0.1%	ADAM	<u>ی</u>
Algorithm: Gradient Descent A Adam Learning rate:	lgorithm:	Dropout rate:		0.1%	ADAM DROPOUT RATE 50%	© 0 0
Algorithm: Gradient Descent Al Adam	lgorithm:			0.1% BETA1	ADAM DROPOUT RATE 50% BETA2	<u>ی</u>
Algorithm: Gradient Descent A Adam Learning rate:	lgorithm:	Dropout rate:	50%	0.1%	ADAM DROPOUT RATE 50%	© 0 0
Algorithm: Gradient Descent A Adam Learning rate:	lgorithm:	Dropout rate:	50%	0.1% BETA1	ADAM DROPOUT RATE 50% BETA2	© 0 0

Figure 5.24: Configure the learning rate

#### 5.4.8.3 Dropout Rate

The **droput** mechanism consists of randomly drop nodes (and their connections) from the network at training time. This prevents nodes from co-adapting so it is an effective method to control **overfitting**. The dropout rate is the proportion of nodes dropped from the network during training.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the dropout rate will be automatically set. If you **disable the automatic options**, you can select a value for the dropout rate (see Figure 5.25).

		abetes diagnosis dataset   Tra		🎭 ⊘- (≡	ŧ) <b>%</b> _ (
EPNET CONFIGURATION					
pjective field:		Automatic optim	ization:		(
Diabetes	ABC				
efault numeric value:	Missing numerics:	: Max. training time	e: 💮 Max.	iterations:	(
Select a default value	* N/A	00:30:00 ©	-0-	20000	
advanced configuration	on				(
Network Architecture:					$\overline{\bigcirc}$
Algorithm:					۹
					0
Gradient Descent Alg	gorithm: Seed	ł:		ALGORITHM	ľ
Learning rate:		oout rate:			0
Cearning rate.	0.1%	50%	0.1%	50%	1
0		<u> </u>			0
Beta1:	0.9	2:	0.9	0.999	
	——————————————————————————————————————	6	0.9	0.999	
					0
Epsilon:					
Epsilon:	1e-8			1e-8	
Epsilon:	1e-8				

Figure 5.25: Configure the dropout rate

### 5.4.8.4 Seed

The random seed controlling the ordering of training data, the initial network weights, and the behavior of dropout during training. If the **automatic network search** option is not enabled, by setting the **same seed** you can get **repeatable** deepnets using the same dataset.

	•••• 1	Diabetes	diagnosis dataset   Train	ing (80%)	ې - 🕉 (	≡)*- (
PNET CONFIGURATION						,
ective field:			Automatic optimiza	ition:		G
abetes	A	BC 🗸	× 4			
ault numeric value:	Missing num	ierics:	Max. training time:	Max.	iterations:	•
elect a default value	N/A		00:30:00 ©	-6-	20000	
Advanced configuration						(
Network Architecture:						♥
Algorithm:						۵
		_				0
Gradient Descent Algor		Seed:			ALGORITHM	<b>U</b>
Adam	<b></b>				DROPOUT RATE	0
Learning rate:	0.1%	Dropout rate:	50%	0.1%	50%	
$\odot$			0			
Beta1:		Beta2:				0
	0.9		0.999	0.9	0.999	
Epsilon:						0
	1e-8				1e-8	
6						
Weights:						$\overline{\mathbf{\nabla}}$

Figure 5.26: Set a seed for the deepnet

# 5.4.9 Weights

It is not unusual for a dataset to have some categories that are common and others very rare. For example, in datasets used to predict fraud, usually fraudulent transactions are very scarce compared to regular ones. When this happens, models tends to predict the most frequent values simply because the overall model's performance metrics improve with that approach. However, in cases such as fraud prediction, you may be more interested in predicting rare values rather than successfully predicting frequent ones. In that case, you may want to assign more **weight** to the scarce instances so they are equivalent to the abundant ones.

BigML provides two different options to assign specific **weight** to your instances, **balance objective** and **objective weights** explained in the following sections.

If one of the automatic optimization options (see Subsection 5.4.2) is enabled, the weights will be automatically set. If you **disable the automatic options**, you can configure the weights of your dataset instances (see Figure 5.27).

EPNET CONFIGURATION					
ective field: abetes	ABC 🗸	Automatic optimization:			(
ault numeric value:	Missing numerics:	Max. training time: 00:30:00 O	Max. ite	erations: 20000	
Advanced configuration					(
Network Architecture:					$\overline{\mathbf{O}}$
Algorithm:					⊘
Weights:					۲
Select a weight option Balance objective Weight field				WEIGHTS No weights	0
Sa Objective weights				614 instances	♥
				Default settings	◙

Figure 5.27: Weight options for deepnets

### 5.4.9.1 Balance Objective

When you set the **balance objective** weight, BigML automatically balances the classes of the objective field by assigning a higher weight to the less frequent classes, with the most frequent class always having a weight of 1. For example, take the following frequencies for each class:

[False, 2000; True, 50]

By enabling the **Balance objective** option, BigML will automatically apply the following weights:

[False, 1; True, 40]

In this example, the class "True" is getting forty times more weight as it is forty times less frequent than the most abundant class.

### 5.4.9.2 Objective Weights

The **objective weights** option allows you to manually set a specific weight for each class of the objective field. BigML oversamples your weighted instances replicating them as many times as the weight stablishes. If you do not list a class, it is assumed to have a weight of 1. Weights of 0 are also valid, but if all classes have a weight of 0, the deepnet creation will produce an error.

This option can be combined with the Weight field (see Subsection 5.4.9.3).

#### 5.4.9.3 Weight field

The **Weight Field** option allows you to assign individual weights to each instance by choosing a special weight field. It can be used for both regression and classification deepnets. The selected field must be numeric and it must not contain any missing values. The weight field will be excluded from the input fields when building the ensemble. You can select an existing field in your dataset or you may create a new one in order to assign customized weights.

For deepnets, the weight field modifies the loss function to include the instance weight. The outcome is similar to the oversampling technique.

## 5.4.10 Sampling Options

Sometimes you do not need all the data contained in your dataset to build your deepnet. If you have a very large dataset, sampling may be a good way of getting faster results. BigML allows you to sample your dataset before creating the deepnet, so you do not need to create a separate dataset first. You can find a detailed explanation of the sampling parameters available in the following subsections. (See Figure 5.28.)

#### 5.4.10.1 Rate

The **rate** is the proportion of instances to include in your sample. Set any value between 0% and 100%. Defaults to 100%.

#### 5.4.10.2 Range

Specifies a subset of instances from which to sample, e.g., choose from instance 1 until 200. The **rate** you set will be computed over the **range** configured. This option may be useful when you have temporal data, and you want to train your deepnet with historical data, and test it with the most recent one to check if it can predict based on time.

#### 5.4.10.3 Sampling

By default, BigML selects your instances for the sample by using a **random** number generator, which means two samples from the same dataset will likely be different even when using the same rates and row ranges. If you choose **deterministic** sampling, the random-number generator will always use the same seed, thus producing repeatable results. This lets you work with identical samples from the same dataset.

#### 5.4.10.4 Replacement

Sampling with **replacement** allows a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once. By default, BigML generates samples without replacement.

#### 5.4.10.5 Out of bag

This argument will create a sample containing only **out-of-bag** instances for the currently defined rate, so the final total number of instances for your sample will be one minus the rate configured for your sample (when replacement is false). This can be useful for splitting a dataset into training and testing subsets. It is only selectable when a sample rate is less than 100%.

bjective field:	ABC	Automatic optimization:			(
plabetes efault numeric value: Select a default value	Missing numerics:	Max. training time: 00:30:00 O	C Max. itera	20000	
Advanced configuration					(
Network Architecture:		The Network Archi	tecture parameters will be	automatically optimize	d 🕤
Algorithm:					
Algoritaini.		The Alg	orithm parameters will be	automatically optimize	
Weights:			orithm parameters will be Veights parameters will be		
-			veights parameters will be		
Weights:	100%		veights parameters will be	automatically optimize	d 🕤
Weights: Sampling:	-0		Veights parameters will be 61	automatically optimize	<ul> <li>✓</li> <li>✓</li> <li>✓</li> </ul>

Figure 5.28: Sampling parameters for deepnet

# 5.4.11 Creating Deepnets with Configured Options

After finishing the configuration of your options, you can change the default deepnet name in the editable text box. Then you can click on the Create deepnet button to create the new deepnet, or reset the configuration by clicking on the Reset button.

betes ABC	Autom	natic optimization:			
ault numeric value: Missing numeric lect a default value V/A	es: Max. tr 01:00:00	aining time:	Max.i	iterations: 20000	
Advanced configuration					(
Network Architecture:					$\overline{\mathbf{S}}$
Algorithm:					$\overline{\bigcirc}$
Weights:					$\overline{\mathbf{O}}$
Sampling:				768 instances	$\overline{\mathbf{O}}$
Dataset advanced sampling:				Default settings	$\overline{\mathbf{O}}$
pnet name: abetes diagnosis dataset			Reset	💐 Create de	epnet

Figure 5.29: Create deepnet after configuration

### 5.4.12 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to

the Reset button (See (Figure 5.29)). This is to show how to create the deepnet programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 5.30) below:

		101
Objective field:	Automatic antimization	0
diabetes	API request preview	
Default numeric v	1 * {	0
Select a default value	<pre>'''' ''objective_field": "000008", ''number_of_hidden_layers": 1, ''name": "Diabetes diagnosis dataset",</pre>	00
Advanced o	<pre>5 "suggest_structure": false, 6 "missing_numerics": true, 7 "max_training_time": 3600,</pre>	۲
Network Arc	8 - "hidden_layers": [ 9 - {	۲
Algorithm:	<pre>10 "activation_function": "tanh", 11 "number_of_nodes": 64</pre>	•
Weights:	Check the full list of arguments in the API documentation	$\overline{\mathbf{v}}$
Deepnet name:		

Figure 5.30: Deepnet API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for deepnets, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the chosen configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

# 5.5 Visualizing Deepnets

If the dataset for creating a deepnet does not contain images, you can analyze your results with BigML **Partial Dependence Plot (PDP)** after the deepnet is created. If the dataset for creating a deepnet contains images, you can analyze your results with BigML **Image Deepnet Page**. In either case, you can also inspect **field importances**.

# 5.5.1 Partial Dependence Plot

Partial Dependence Plot (PDP) is a heatmap chart for examining the impact of the input fileds on the objective field.

The PDP view is composed of three main parts: the CHART itself, the PREDICTION legend and the INPUT FIELDS form. (See Figure 5.31.) You can find a detailed explanation of each one below.



Figure 5.31: Deepnet chart parts

• The CHART allows you to view the impact of two **input fields** on the objective class **predictions** remaining the rest of input field values constant. You can select any categorical or numeric input field for each axis and the class probabilities are represented in different colors in the heatmap. You can switch the axis by clicking on the option on top of the chart area. (See Figure 5.32).

You can find the values for the fields in the axis in the grey area next to the selector. Freeze the view by pressing Shift and release it again by pressing Escape from your keyboard. When the view is frozen, an edition icon will appear in this grey area so you can edit the axis values and obtain the prediction for another preferred value. (See Figure 5.32).

AUTOMATIC OPTIMIZATION	NETWORKS EVALUATED	MISSING NUMERICS TRUE	OBJECTIVE FIELD DIABETES	614
				C
Age 123	▼ 52.8750	• 13		esc Releas
80		True	PREDICTION	Probability
75 -		50.3%	6	72.6%
70 -		False		
65 -				28.7
60 ·		True		71.2
55 -				
50 -		All cl	asses	
45		17	INPUT FIELDS	All fields:
40 -		Pre	gnancies	6.38%
35 -		0		21
30				
25 -		Glu	cose	50.69%
0 ²⁹ 0 ¹⁰ 0 ¹⁰ 0 ¹⁰	10. 19. 18. 10. 19.	v v		248
80 80 40 20		2° 22		
X Diabetes pedigree	e <b>123</b> 🔹 1	.2040 🖋 Bio	od pressure 🧰	38.42%

Figure 5.32: Deepnet chart

The PREDICTION legend allows you to visualize the objective field classes (classification deepnets) or the predicted value (regression deepnets). For classification, each class is represented by a color, the main probability color bar at the top is the probability for the predicted class. By default, colors are shaded according to the prediction range shown in the chart area. That way, smaller differences in predictions are easier to perceive. However, you can choose to see the color shading according to the total range of possible values for the objective field by clicking on the icon next to the prediction bar Total. (See Figure 5.33.) You can also select to see only one of the classes using the class selector at the bottom of the legend.

AUTOMATIC OPTIMIZATION	NETWORKS EVALUATED	MISSING NUMERICS	OBJECTIVE FIELD DIABETES	614
				C
<b>Y</b> Age 123	52.8750	<ul> <li>\$\$</li> </ul>		esc Releases
- 123	02.0100	C X	PREDICTI	ION
- 08		True		Probability
75		50.39	Ж	72.6%
70 -		False	2	28.74
65 -		True		71.26
60				11.20
55 -				
50 -		All c	lasses	
45 -		17	INPUT FIE	LDS All fields:
40 -		Pre	egnancies	6.38%
35 -			0	21
30 -				5
25		Glu	icose	50.69%
0 ²⁹ 0 ¹⁰ 0 ¹⁸ 0 ¹⁸	2. 2. 1. 1. 1. 1.	2 22	0	248

Figure 5.33: Prediction legend

Again, freeze this view by pressing Shift, and release it again by pressing Escape from your keyboard.

• Below the chart legend, you can find the INPUT FIELDS form. (See Figure 5.34.) You can configure the values for any **numeric**, **categorical**, **text** or **items field**. By changing their values, you can see the predictions changing in real-time. You can also enable or disable the input fields, so they will be treated as missing values. You can sort the fields by their importance to predict the objective field. (See Subsection 5.4.4.)

AUTOMATIC OPTIMIZATION	NETWORKS EVALUATED	MISSING NUMERICS TRUE	OBJECTIVE FIELD DIABETES	614
				Q
Y Age 123	52.8750	<ul> <li>Ck</li> </ul>	PREDICTI	ese Releases
80 -		True		Probability
75		50.39	6	72.6%
70		False	)	28.74
65		True		71.269
60 ·				
50 -		All c	asses	
45				
40 -		17		
35		Pre	gnancies	6.38%
30			-0	3
25 -		Glu	cose	<b></b> 50.69% 🔲
0. ²⁰ 0. ⁴⁰ 0. ⁴⁰ 0. ⁴⁰	20 2° 2° 2° 20 20	2° 22		248 <b>120</b>

Figure 5.34: Configure the values for other input fields

Moreover, the chart includes a reset option for your input fields values, and an export option to download your chart in PNG format explained below:

• After selecting the fields for the axis or configuring the input fields values, you can set them again to the default view by clicking the **reset** icon highlighted in Figure 5.35.



Figure 5.35: Reset the values for the input fields

• You can also **export** your chart in PNG format with or without legends. Freeze the view by pressing Shift from your keyboard and export the chart to get the class percentages in the legend. Release the view by pressing Escape.



Figure 5.36: Export chart as image with or without legends

Note: there are some limitations in the number of classes of the objective field and the number of input fields to visualize your deepnet in the chart (explained in Subsection 5.8.1).

# 5.5.2 Image Deepnet Page

As stated in the previous section, when the dataset used to create a deepnet contains images, the deepnet created will be a convolutional neural network (CNN). See the explanation in Subsection 5.2.1.

A CNN deepnet uses images (i.e. raw image pixels) as input fields. If there are image feature fields that were extracted from those same images in the dataset, they are ignored during CNN training. After a CNN deepnet is created, you can analyze the results with BigML Image Deepnet Page.

On the top row of the Image Deepnet Page are applicable parameters of the deepnet, which may include its hidden layer number, algorithm, optimization option. It also lists the objective field of the deepnet and its number of instances.

Below the top row, the view of the Image Deepnet Page shows the performance of the deepnet on a set of sampled instances. This set is called holdout set, which is used for validation during deepnet training.

Depending on the objective field, the Image Deepnet Page has two variations.

### 5.5.2.1 Image Deepnet Page - Classification

If the objective field is categorical, the Image Deepnet Page is the classification variation.

Below the top row, the view is composed of three main sections: the Image Results, the Performance Panel and the Class List. See Figure 5.37.

		ifar10-train	Ê	
AUTOMATIC OPTIML	NETWORKS EVALUATED	MISSING NUMERICS	OBJECTIVE FIELD	INSTANCES 50,000
CORRECTLY PREDICTED:			893/1020	DEEPNET PERFORMANCE
and .	and the second second	5	Corre	Performance overall ectly predicted: 87.55% 893/1020
✓ dog 93.34%	automobile 99.999	√ 6 ⊘ frog	99.99%	PROBABILITY
Sec.	4-	-	26.5	10 CLASSES 97.06% 99/102
✓ deer 99.96%	irplane	√ truck	99.92%	mobile 94.12%
	< >		airpla	ane 91.18% 93/102
NCORRECTLY PREDICTED:		and a	12.45% 127/1020 deer	89.22% 91/102
A DECK	21	100	horse	e 87.25% 89/102
(≫ ship √ frog 63.84%	Cat 72.64	% ⊗ horse √ frog	64.73%	87.25% 89/102
Star V.			frog	86.27% 88/102
1	1	<u>_</u> )	bird	81.37% 83/102
(×) dog	Cat V truck 44.95	% Ship ✓ horse	59.58%	81.37%
	< >			

Figure 5.37: Image Deepnet Page Classification layout

As stated above, the Image Deepnet Page mainly shows the performance of the deepnet on a set of sampled instances, called the holdout set. The number of the instances in the holdout set is at most 20% of the total instances in the dataset, or 1024, whichever is smaller.

• The Performance Panel shows the overall performance of the deepnet on the holdout set. See Figure 5.38. This is by default. However when a class is selected in the Class List, the Performance Panel shows the performance for that specific class.



Figure 5.38: Image Deepnet Page Classification Performance Panel

The performance metrics, either correctly predicted or incorrectly predicted, are provided as both percentage and count. For instance, when the count is *893/1020* as shown in Figure 5.38, that means 893 images were predicted correctly in the holdout set of 1020.

The probability slider allows users to filter the results by selecting the range of probabilities for each image classification. This controls how many images are shown in the Image Results section.

 The Image Results section shows two subsections, with each a paginable list of images from the holdout set, Correctly predicted and Incorrectly predicted. In each list, every image has a caption which shows its predicted class, true class and probability. In the subsection of Correctly predicted, because a predicted class is the same as its true class, only one is shown in the caption. In the Incorrectly predicted subsection, both predicted class and true class are shown for each image. The length of the solid color in a probability bar is proportional to its value. See Figure 5.39.



Figure 5.39: Image Deepnet Page Classification Image Results

On the top right of each subsection are the performance metrics by percentage and count. The list of images are paginable by using the pagination arrows at the bottom of each subsection. Each page shows up to 6 images, scaled to fit the area.

When users mouse-over an image, a popup box would show prediction result of the image: predicted class, true class and probability. In the example shown in Figure 5.39, the popup box is red in background, signaling an incorrect prediction, which its predicted class is *frog* while its true class is *bird* and the probability is *70.08%*.

The Class List shows all the classes in the holdout set, sorted by their numbers of occurrences. In
other words, the list is ranked by class popularity. See Figure 5.40. The recall rate of each class is
displayed by percentage and count at the right side of class bar. Recall is defined as the number
of correctly predicted images for the class divided by the total number of images for the class. The
length of the green color bar in proportion to the full class bar corresponds to the value of the recall
of the class.



Figure 5.40: Image Deepnet Page Classification Class List

The Class List is scrollable when there are not enough space to show all of them together. There is also a download icon in the section heading that users can use to download the class list as a CSV.

There are two controls that can change the number of images to show in Image Results section. One is the probability slider in the Performance Panel. Another is a class bar in the Class List.

By default, the Image Results section shows all the images in the holdout set, both correctly
predicted and incorrectly predicted. The probability slider displays the lower and upper ends of the
probabilities associated with all the classification results in the set. Either end of the slider can be
changed by dragging the respective knob, then the Image Results section will only show images
with the probabilities within the range of the slider. The images having the probabilities outside of
the range will be filtered out.
AUTOMATIC OPTIML	NETWORKS EVALU	JATED MISS		OBJECTIVE FI		ISTANCES
NETWORK SEARCH	200		TRUE	LABEL		50,000
RRECTLY PREDICTED:				92.24% 856/928	DEEPNET PERF	
					From 1020 sample	
2275	1 -	7	at		Performance Correctly predicted:	<u>e overall</u> 92.24%
- A		-			Incorrectly predicted	856/928
deer 99.6	5% automobile	99.94%	⊘ horse	99.68%	64.0%	LITY
				2	10 CLAS	SES
truck 100.0	0% 🕑 deer	97.48%	automobile	98.24%	automobile	94.12%
	<	>			airplane	91.18%
ORRECTLY PREDICTED:				7.76% 72/928		93/102
- 1			Refer Cont	12/928	deer	89.22% 91/102
02	An	6	and a state	100	horse	87.25%
0.3-4				2		89/102
airplane	6% × horse	98.83%	× airplane	66.79%	truck	87.25% 89/102
✓automobile	✓ bird		✓automobile		frog	86.27%
and so it			-	3		88/102
	¥	*	ASC		bird	81.37% 83/102
deer 67.0	8% Cat	73.11%	(×) dog √ cat	99.50%	cat	81.37%
			,			

Figure 5.41: Image Deepnet Page Classification filtering by probability

As in Figure 5.41, the lower end of the probability slider was changed to 64.0%. Now any image with the classification probability lower than 64.0% won't show up in the Image Results section, in both correctly predicted and incorrectly predicted. The performance metrics were changed as well, from 893/1020 to 856/928, reflecting the fewer number of images shown.

The probability slider can be reset to the default by dragging its knobs to include all possible probabilities, or by pressing the reset icon in the heading of the Performance Panel.

Both ends of the probability slider are editable text input boxes, so users can enter precise numbers if so desired.

 In the Class List, one class can be selected by clicking on its class bar. When this happens, the Image Results section will only show images of that class. The Correctly Predicted subsection shows all images whose true class is that class, and whose predicted class is that class too. The Incorrectly Predicted subsection shows all images whose true class is that class selected, but whose predicted class is a different class. The performance metrics are changed accordingly.



Figure 5.42: Image Deepnet Page Classification showing one class

In Figure 5.42 above, the class *truck* is selected, as shown by its darkened background. Now the Images Results section shows 89 images in the correctly predicted subsection, all *truck*, and 13 images in the incorrectly predicted subsection, all *truck* but all predicted as something else.

Also in Figure 5.42, note that in the Performance Panel, not only the performance metrics are changed accordingly, but also the title becomes "Performance for truck" instead of "Performance overall".

To reset the Image Results to include all classes, use the reset icon in the Performance Panel heading.

• The class selection and the probability slider can be combined to show only images of one class which has a selected range of probabilities.

#### 5.5.2.2 Image Deepnet Page - Regression

If the objective field is numeric, the Image Deepnet Page is the regression variation.

Below the top row, the view is composed of two main sections: the Image Results and the Performance Panel. See Figure 5.43. The example deepnet in the figures of this section was created from a dataset for estimating the number of penguins in images, with its numeric objective field "count".



Figure 5.43: Image Deepnet Page Regression layout

As stated above, the Image Deepnet Page mainly shows the performance of the deepnet on a set of sampled instances, called the holdout set. The number of the instances in the holdout set is at most 20% of the total instances in the dataset, or 1024, whichever is smaller.

• The performance panel shows the performance of the deepnet on the holdout set, in terms of error percentages. All numeric predictions have errors. A value in percentage, called **split**, is used to divide all predictions in the holdout set into two groups. The default split is 10%, which can be changed by users.



Figure 5.44: Image Deepnet Page Regression Performance Panel

The size of the holdout set is shown right below the panel header. It also shows how many instances are displayed, as this can be changed by a filter.

As seen in Figure 5.44, the split is at 10%, by default, so the images are divided into two groups, one "Predicted within 10.00% error", another "Predicted outside of 10.00% error". Both percentages and fractions are employed to show the relative sizes of the two groups in regard to the split. In the example, for instance, there are 2 images predicted within 10% error, so for this group, it's *6.67%* or *2/30*.

"Average percent error" is a metric showing the average error a prediction has in each group. It is the sum of the absolute values of the percentage errors of all instances, divided by the number of instances.

There are two controls that can change how the Image Results section appears.

1. The DISPLAY ERROR slider allows users to filter the results by selecting the range of errors for each prediction. This controls how many images are shown in the Image Results section.

By default, the Image Results section shows all the images in the holdout set, which are divided into two groups by the split. The DISPLAY ERROR slider shows the greatest negative error at its left end, and the greatest positive error at its right end. Note: When a prediction is more than the true value, it produces a positive error. Conversely, a negative error. Either end of the slider can be changed by dragging the respective knob, then the Image Results section will only show images with the predictions within the range of the slider. The images having the prediction errors outside of the range will be filtered out.



Figure 5.45: Image Deepnet Page Regression Display Error Slider

As in Figure 5.45, the lower end of the DISPLAY ERROR slider was changed to -15.0 and the upper end to 0.5. Now any image with the prediction error outside of the range, i.e. either less than -15.0 or greater than 0.5, doesn't show up in the Image Results section. There are 21 images displayed, instead of 30, as indicated inside the parentheses below the panel heading. The relative sizes of the split groups are changed as well, from 2/30 and 28/30 to 2/21 and 19/21, respectively, reflecting the fewer number of images displayed.

The DISPLAY ERROR slider can be reset to the default by dragging its knobs to include all possible errors, or by pressing the reset icon in the heading of the Performance Panel.

Both ends of the DISPLAY ERROR slider are editable text input boxes, so users can enter precise numbers if so desired.

2. The SPLIT ERROR slider allows users to select the **split**, which is the error percentage used to divide the Image Results section into two groups: one having prediction errors smaller than the split, and another having prediction errors bigger than the split.



Figure 5.46: Image Deepnet Page Regression Split Error Slider

As seen in Figure 5.46, moving the knob on the SPLIT ERROR slider changes the split value, which is shown above the slider, anywhere between 0% and the biggest possible error percentage produced by the holdout set. In addition, below the slider shows the total number of instances displayed, on the left of the slider is the number of instances in the group with prediction errors smaller than the split, and on the right the number of instances in the group with prediction errors bigger than the split. In the parentheses are the percentages of the group size in respect to the total instances, respectively. As the split value changes, both group sizes (and percentages) on the left and right sides change accordingly. This gives a direct visualization of how the split affects the partition of the Image Results section.

The SPLIT ERROR slider can be reset to the default value, 10%, by pressing the reset icon in the heading of the Performance Panel.

The split value above the SPLIT ERROR slider is an editable text input box, so users can enter a precise split value if so desired.

The total number of instances below the SPLIT ERROR slider may be affected by the DIS-PLAY ERROR slider.  The Image Results section shows two subsections, which represent two groups divided by the split value. The default split value is 10%. One group is called "Predicted within 10% error", which has the images with prediction errors less than 10%. Another group is called "Predicted outside of 10% error", which has the images with prediction errors greater than 10%. When the split value is changed by the SPLIT ERROR slider in the Performance Panel, the subsection titles and images change accordingly.



Figure 5.47: Image Deepnet Page Regression Image Results

As seen in Figure 5.47, each subsection in the Image Results section is a paginable list of images from the holdout set. In each list, every image has a caption which shows its predicted value, true value, error and error percentage.

On the top right of each subsection is the relative size of the subsection by percentage and count, with respect to the total number of instances displayed. The list of images are paginable by using the pagination arrows at the bottom of each subsection. Each page shows up to 6 images, scaled to fit the area.

When users mouse-over an image, a popup box would show prediction result of the image: pre-

dicted value, true value and the error which is defined as (predicted - true). In the example shown in Figure 5.47, the popup box is red in background, signaling a prediction which error is greater than the split value, with its predicted value as *2.90374*, its true value *6.00000* and the error *-3.09626*.

The total number of images displayed in the Image Results section may be affected by the DIS-PLAY ERROR slider in the Performance Panel.

#### 5.5.3 Summary Report

#### 5.5.3.1 Field Importances

The **field importances** for deepnets provide a measure of how important an input field is relative to the others to predict the objective field. Each field importance is normalized to take values between 0% and 100%. All field importances should sum 100%. You can access them by clicking in the Summary Report option shown in Figure 5.48. You can also export the field importances in PNG, CSV and JSON format.



Figure 5.48: Field importance for deepnets

Deepnet field importances are based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

#### 5.5.3.2 Summary

If you created your deepnet using the **Automatic Network Search** option (see Subsection 5.4.2), you will be able to see a tab called "Summary" next to the field importances tab (see Figure 5.49). Find here the configuration of each of the networks composing the deepnet in JSON format. You can find all the parameters explained in Section 5.4 per network.

	Deepnet Summary Report	× WhizzML •
onet u	Field importance Summary	uration.
	Top performing networks configuration:	(=)5, (
1000	1- {	AMOES
	<pre>2 - "network1": { 3 "learn_residuals": false,</pre>	
NETV	4 "dropout_rate": 0,	614
	5 - "activation_functions": [ 6 "relu"	
	7 ],	0 9
	8 - "class_weights": [	
	9 1,	
	10 1 11 ],	
	12 "activation_function": "relu",	
	13 "descent_algorithm": "adam",	pility
	14 "learning_rate": 0.01, 15 - "layer_sizes": [	
	16 64	max.
	17 ], 18 "hatch size", 128	THAT.
		Close

Figure 5.49: View networks' configuration

## 5.6 Deepnet Predictions

## 5.6.1 Introduction

The ultimate goal in building a deepnet is being able to make predictions with it. In BigML, you can make predictions for **single instances** or for **many instances in batch**. Each prediction comes with a measure indicating the prediction confidence. For regression problems, the **expected error** is provided along with the predicted value, while for classification problems the vector of **probabilities** per class is returned (a percentage ranging from 0% up to 100%).

The predictions tab in the main menu of the BigML Dashboard is where all your saved predictions are listed. (See Figure 5.50.) You can **search** your predictions by name clicking on the search option on the top menu. In the predictions list view, you can see, for each prediction, the **deepnet** icon used for the prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created).

Source	s Datasets Supervised    Unsupervise	d   Predictions	Tasks			WhizzML 🔻
4		Pre	edictions			â <
វា	Name	Ob	jective 🗘	Prediction	\$	<b>***</b>
R	Diabetes diagnosis   Training (80%) 8 inputs	Dia	betes	True		1mi
R	Fictional Wine Sales   Training (80%) 5 inputs	Tota	al Sales	62.539		1h 43mi
R	Diabetes diagnosis   Training (80%) 8 inputs	Dia	betes	True		1h 44mi
×°	Diabetes diagnosis   Training (80%) 8 inputs	Dia	betes	True		1h 47mi
×	Diabetes diagnosis   Training (80%) 8 inputs	Dia	betes	True		1h 47m
×°	fashion-mnist_train.csv   Training (80%) 785 inputs	lab	el	5		2d 14
R	fashion-mnist_train.csv   Training (80%) 785 inputs	lab	el	5		3d 12
R	fashion-mnist_train.csv   Training (80%) 785 inputs	lab	el	5		3d 13
Ś	Fictional Wine Sales   Training (80%) 5 inputs	Tota	al Sales	62.29335		4d 1
Ś	fashion-mnist_train.csv   Training (80%) 785 inputs	lab	el	5		4d 13
10w 10	0 ᅌ predictions	1 to 10 of 38	3 predictions		K < 1 2	34>>

Figure 5.50: Predictions list view

When you first create an account at BigML, or every time that you start a new project, your list of predictions will be empty. (See Figure 5.51.)

Sources	Datasets	Supervised - U	Jnsupervised - Pred	ictions Tasks	5		WhizzM	иL <del>-</del>
Ŧ	Ē			Predictions			ŵ	0
វា	Name			Objective	\$	Prediction	\$	<b>a</b> 🗘
				No predictions				
Show 10	predictions			No predictions			I< <	> >1

Figure 5.51: Empty predictions list view

Deepnet predictions are saved under the CLASSIFICATION & REGRESSION option in the menu (see Figure 5.52.)

Sources	NEW Datasets Supervised ▼ Un	supervised  Predictions	Tasks		Whizz	:ML 🕶
ŧ	<b>I</b>	CLASSIFICATION & REGRESSION	ions		Ô	Q,
ŵ	Name	ANOMALY DETECTION	\$	Prediction	≎ É	<b>*</b>
×°	Diabetes diagnosis   Training (80%) 8 inputs	ASSOCIATION DISCOVERY		True		2min
×°	Fictional Wine Sales   Training (80%) 5 inputs	Total Sale	S	62.539		1h 44min
R	Diabetes diagnosis   Training (80%) 8 inputs	Diabetes		True		1h 45min

Figure 5.52: Menu options of the predictions list view

Select the list for your single instances predictions or your batch predictions by clicking on the corresponding icons. (See Figure 5.53 and Figure 5.54.)



Figure 5.53: Single predictions icon



Figure 5.54: Batch predictions icon

## 5.6.2 Creating Deepnet Predictions

BigML provides two options to predict with your deepnets explained in the following subsections:

- PREDICT: to predict one single instance
- BATCH PREDICTION to predict multiple instances in batch.

#### 5.6.2.1 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the input fields used by the deepnet, so you can easily set the values and get an immediate response.

Follow these steps to create a single prediction:

1. Click PREDICT in the deepnet 1-click action menu. (See Figure 5.55.)



Figure 5.55: Predict using the 1-click action menu

Alternatively, click PREDICT in the pop up menu in the list view. (See Figure 5.56.)

			×	Deepnets				ŵ	Q
11	Name	\$	Туре	Objective	÷ 🛗	<b>ê</b> \$	\$	\$	08
ılî	Digit recognizer (train)   Training (80%) 1 hidden layers, adam, learning rate=0.001, 2689	8	00	label	5h 53min	58.6 MB	0	0	0
ılî	Digit recogni		00	label	5h 57min	58.6 MB			
ılî ^Q	US car accid 1 hidden laye EVALUATE			Injury Severity	6h	578.3 KB			1
ılî	US car accid 1 hidden laye		00	Injury Severity	6h 22min	578.3 KB			1
ılî	US car accid Z DELETE DEEPNET			Injury Severity	6h 23min	578.3 KB			
ılî	US car accid 1 hidden layers, rms_prop, learning rate=0.001, 2	3	*	Crash Date.day-of-we	6h 27min	592.8 KB			
ılî	US car accidents   Training (80%) 1 hidden layers, momentum, learning rate=0.04,		00	Injury Severity	6h 27min	592.8 KB	1		
ılî	US car accidents   Training (80%) 1 hidden layers, ftrl, learning rate=0.001, 177 iter		00	Injury Severity	6h 29min	592.8 KB			
ılî	US car accidents   Training (80%) 1 hidden layers, adagrad, learning rate=0.001, 24		00	Injury Severity	6h 30min	592.8 KB			
ıl <mark>e</mark>	US car accidents   Training (80%) 1 hidden layers, adam, learning rate=0.02, 1201 i		00	Injury Severity	6h 31min	592.8 KB			

Figure 5.56: Predict using the pop up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the deepnet as input fields. (See Figure 5.57.)

Sources Data	asets Supervised	<ul> <li>Unsupervised -</li> </ul>	Predictions	Tasks			WhizzML 👻
¥.		Predict	using Diabe	tes diagnosis	Training (80	<b>3</b> -	(≡)⁵-⊂ 〔〕
			Diabetes:	?	$\langle$	?	
						All inp	ut fields: 🕑
Pregnancies 0		21	<b></b>	Glucose 0		248	✓ 124
Blood pressure	0	152	<b></b>	Skinfold 0	0	123	€ 61
Insulin 0	0	1057	<b>5</b> 28	<b>BMI</b> 0.0	0	83.88	<b>€</b> 40.74
0.0	ree	2.89	<b>¥</b> 1.4	<b>Age</b> 6		96	<b>∑</b> 51
New prediction Diabetes diagr	name nosis   Training (80%	6)				Pr	edict

Figure 5.57: Deepnet prediction form

3. Select the fields to be used for your prediction, set the input values for your selected fields and click Predict (Figure 5.58.) Non-selected fields will be considered as missing values during the

#### prediction.

Sources Datasets	Supervised -	Unsupervised -	Predictions				WhizzML 🔻
1		Predict	using Diabe	tes diagnosi	is   Training (80	3-	(≡)⁵, (i)
			Diabetes:	:?		?	
						All inp	out fields: 😑
Pregnancies				Glucose			≤
0		21	10	0		248	168
			Disable this inp	out field	0		100
Blood pressure			Č	Skinfold			≤
0		152	112	0		123	36
	0		112		0		00
Insulin			≤	BMI			
0		1057	750				40.74
	0		750				40.74
Diabetes pedigree				Age			
			1.4				51
			1.4				51
New prediction name							
Diabetes diagnosis   1	Training (80%)					Pr	redict

Figure 5.58: Deepnet predictions form

4. Get the prediction at the top of the view along with the rest of class probabilities. (See Figure 5.59.) BigML predictions are synchronous, i.e., when you send the input data, you get an immediate response.

Sources Datasets S	upervised  Unsupervised	Predictions Tasks		WhizzML 👻
₽ <b>₽</b> .	Dia	betes diagnosis   Trainir	ng (80%)	<, (≡), (≡), (∋)
		Diabetes: True		73.42%
			73.42%	
	26.58%			
	False		THE	
				All input fields: 🗖
Pregnancies		Glucose		
°G	21	10	©	248 <b>168</b>
Blood pressure		Skinfold		<b>≤</b>
0	152	112 0	6	123 <b>36</b>
Insulin		🗹 вмі		
0	1057	750	6	^{83.88} 40.74

Figure 5.59: Get the deepnet prediction

5. Hide or display the histogram view containing the rest of your **class probabilities** by clicking on the icon highlighted in Figure 5.60. You can download all the probabilities in PNG format, in CSV or JSON file by clicking on the corresponding icons. (See Subsection 5.6.4.1.)



Figure 5.60: Display all class probabilities

6. Your prediction is automatically saved and you can find it in the predictions list view.

For **regression** deepnets, the process is the same, but instead of the predicted classes and the probabilities you get a numeric value for the objective field along with an expected error as the certainty measure.

Note: this option is only available from the BigML Dashboard for deepnets with less than 100 fields. If you want to perform single instance predictions for a higher number of fields, use the BigML API¹⁷.

#### 5.6.2.2 Deepnet Prediction with Images

Deepnet created from a dataset containing images is a convolutionary neural network (CNN) (See Subsection 5.2.1). Because BigML treats images as an input type just like any other data types such as numeric and text, creating predictions using CNN is the same as other deepnets. Everything stated in the previous section still applies. The only thing different is input fields of images.

Follow these steps to create a single prediction:

1. Click PREDICT in the deepnet 1-click action menu. See Figure 5.61.

Sources	Datasets	Supervised -	Unsupervised -	Predictions -	Tasks		WhizzML 👻
A 7	• 60-	••••	×.	grape-strawbe	rry	Ê G	¢ 🧆, (≡) [¢] , ()
HIDDEN	LAYERS	ALGORITHM	LEARNING RATE	MISSING NUME			NCES
		ADAM	0.001	TRUE	BATCH PRED	DICTION	D
					EVALUATE		
	I				CREATE FUSI	ION	
					Relete deef	PNET	
CORRECTLY	PREDICTED:				MOVE TO		MANCE O
	L -		- the second sec			14/14 Fron	n 14 sampled instances

Figure 5.61: Predict with images using the 1-click action menu

Sourc	es Datasets	Supervised - Unsupervi	ised 🔻 Pre	dictions 👻	Tasks				Whizzł	ИL <del>-</del>
×			×	Deepnets				Ŵ	Q	Ø
						Search by	name		:	ר
ill	Name		$\dagger_{\downarrow}$ Type $\dagger_{\downarrow}$	Objective	ţ.	t,	i ti	<b>\$</b> 1,	<b>(</b>	08
ılı	bluebird-gps-tagge auto structure sugge	d stion, 1 hidden layers, adam	00	label	1	2h 14min	36.6 MB			
ılı	bluebird-gps-tagge auto structure sugge	d stion, 1 hidden layers, adam	木	gpsaltitude	1	2h 24min	36.6 MB			
ılı	grape-strawberry auto structure sugge	stion, 1 hidden layers, adam	8 ol	label	1	9h 19min	13.4 MB	9		
ılî	auto structure	PREDICT	20 <b>1</b>	label		3d 13h	177.2 MB			
ılî	lily-daylily   Tr auto structure		20	label		5d 14h	77.5 MB			
ılî	hot_dog-not_		20	label		1w 6d	229.3 MB			1
ılî	hot_dog-not_ 1 hidden layer:	VIEW DETAILS	20	label		1w 6d	229.3 MB			1
ılî	gorilla-horse- auto structure	MOVE TO	>.o	label		2w 2d	115.4 MB			

Alternatively, click PREDICT in the pop-up menu in the deepnet list view. See Figure 5.62.

Figure 5.62: Predict using the pop-up menu

17https://bigml.com/api/predictions

³⁹⁷ 

2. You will be directed to the **prediction form** where you will find all the input fields used by the deepnet. One of them is the image field. As seen in Figure 5.63, click on the input field box to select an image. Because this is a single prediction, an image is input by using a single image source. Clicking on the input box, single image sources available will be in the dropdown list. You can also use the search box to locate specific ones.

Sources Datasets	Supervised - Unsup	ervised - Predictions - Tas	sks	WhizzML 🔻
<b>7</b> .		Predict using grape-straw	vberry	(≡) ⁵ , (Ξ)
		label: ?	?	e
Select a positive class	Probability threshold:	50%		
				All input fields: 🕑
image_id		100.00%		
Select image		A		
New pred	<b>).jpg</b> ed, image, 2 fields (1 path, 1 image	e)		
	9 <mark>69.jpeg</mark> , image, 2 fields (1 path, 1 image)			Predict
	ict-images.zip , image, 6 sources, 236 fields (234			

Figure 5.63: Select a single image source in the image input field

3. Oftentimes single image sources were used for creating a composite source, they become component sources of the composite source. Or an image was uploaded as a part of an archive file (zip/tar) which created a composite source. In those cases, the composite source will be shown in the dropdown list, along with an icon "List components". In the example in Figure 5.64, predict-images.zip is a composite source, click on the icon to show its component sources.

Sources	Datasets Supervised -	Unsupervised -	Predictions -	Tasks			WhizzM	L 🕶
<b>T</b> .		Prec	lict using grape	-strawberry				()
			label: ?		?			6
Select a posit	tive class   Probability	threshold:	-6	50%				
							All input fields:	≤
image_id			00.00%					
Se	lect image		^ <b>D</b>					
			~					
New prec	1659.jpg closed, image, 2 fields (1	oath, 1 image)						
grape-s 🍒	696969.jpeg closed, image, 2 fields (1	oath, 1 image)				۵.	Predict	
	predict-images.zip open, image, 6 sources, 2	3 List components						
	grape-strawberry.zip closed, image, 70 sources	, 237 fields (1 c	=					

Figure 5.64: List the components of a composite source

After the component sources of the composite are listed, scroll the dropdown list to find the desired one, then click to select it, as shown in Figure 5.65.

Sources	Datasets	Supervised -	Unsupervised	Predictions	Tasks			WhizzM	L
<b>**</b> *			n Pr	edict using grap	e-strawberry				()
				label: ?		?			۲
Select a po	ositive class	<ul> <li>Probability th</li> </ul>	reshold:	0	50%				
								All input fields:	≤
image_id				100.00%					
	Select image			^ <u>D</u>					
	🔶 🔛 pi	redict-images.zip							
New pree		ba		Q					
grape-s	open,	image, 2 fields (1 path	, 1 image)				e.	Predict	
		69458.jpg image, 2 fields (1 path	, 1 image)						
1		6 <b>9.jpeg</b> image, 2 fields (1 path	, 1 image)						
		20.jpg image, 2 fields (1 path	, 1 image)						

Figure 5.65: Select a component of a composite source

1	Once an image is calested	aliak on the groon button	Prodict	to create a prediction for the image.
4.	Once an image is selected,	click on the green button	Fledict	to create a prediction for the image.

	label: straw	perry 91.12	<mark>⁄₀</mark>
	91.12%	8.85%	
	starter 1	st ^{ore}	
Select a positive class	Probability threshold:	50%	All input fields: 🕑
mage_id 696969.jpeg	× ×		



As shown in Figure 5.66, after the new prediction is created, the predicted class is at the top of the form along with its probability.

Below the prediction is a histogram representing all the classes of the objective field with their respective predicted probabilities. Show or hide this histogram by clicking on the icon on the right of the predicted class (Figure 5.66). Up to seven different classes can be shown in the histogram together. When there are more than seven classes, the view can be scrolled by clicking on the **arrow** icons on the right. See Figure 5.67. If desired, click on the **sort** icon on the left to sort the classes in the histogram by probability.

Sources	Datasets	Supervised 🕶	Unsupervise	ed 👻 🏼 Pi	redictions	<ul> <li>Tasks</li> </ul>				WhizzM	L 🕶
A 77-				2	cifar10-t	rain				(=) ⁵ .	()
Sort classes	by probability			label:	horse			95.93%	)		۲
<u>hi</u>						95.93%					
		0.31%	0.48%	0.25%	0.49%		0.1%	1.26%	← →		
		c ^{ab}	deet	80 ⁶	140 ⁹	HOISE	ship	wuck			
Select a posi	itive class 🔻	Probability the	reshold: =	6	)	50%				All input fields:	<b>F</b> .
image_id				100.00%	<b>S</b>					An input neius.	
10	)5_0174.jpg			× ×							
New predict									€.	Predict	

Figure 5.67: Deepnet image prediction with more than seven classes

The histogram can be exported in PNG format, in a CSV file, or in a JSON file by clicking on the corresponding icons on the top right.

5. In addition to images, deepnets may use other fields, which will be in the prediction form too. As shown in Figure 5.68, all the fields can be selected, and their input values be set by dragging the knobs on the sliders or by entering precise values in their input boxes.

Predict using bluebird-gps-tagged     Iabel:?     ?     Belect a positive class        Probability threshold:     Box     All input fields:     Image_id     Imag	Sources	Datasets	Supervised -	Unsupervi	sed - Predic	tions - Tasks	_	_	WhizzML 👻
Select a positive class       Probability threshold:       50%         Image_id       98.94%       99.10325         image_id       98.94%       158.32       45.18       97.10325         gpsalititude       1.07%       99.51       58.08       37.57352				E F	Predict using b	luebird-gps-tag	jged	G	· (≡) [€] . (i)
Select a positive class Probability threshold:					label: ?		?	)	۲
Select a positive class     Probability threshold:     Select a positive class     Probability threshold:     Sold     Select a positive class     Probability threshold:     Select a positive class     Probability threshold:     Select a positive class     Probability threshold:     Select a positive class     Select a positive class <tr< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr<>									
Construction       All input fields:       Image_id         Image_id       96.94%       Image_id       Image_id         Image_id       96.94%       Image_id       Image_id         Image_id       96.94%       Image_id       Image_id         Image_id       96.94%       Image_id       Image_id         Image_id       14419421.jpg       Image_id       Image_id         Image_id       107%       Image_id       Image_id         Image_id       107%       Image_id       Image_id         Image_id       2217       Tog       Image_id       Image_id         Image_id       2217       Tog       Image_id       Image_id       Image_id									
Control of a linear of									
All input fields:         image_id       96.94%       gpslongitude       1.35%       1         344419421.jpg       x       image_id       -158.32       45.18       -97.10325         gpsaltitude       1.07%       gpslatitude       0.44%       23.57       58.08       37.57352									
image_id     96.94%     gpslongitude     1.55%     Image_id       344419421.jpg     Image_id     158.32     45.18     -97.10325       gpsaltitude     1.07%     Image_id     1.07%     Image_id     0.04%       0     2217     709     23.57     58.08     37.57352	Select a po	ositive class 🔻	Probability thre	eshold:		50%			
344419421.jpg     -97.10325       gpsaltitude     1.07%       0     2217       709	image_id			(	96.94%	gpslongitude		All in	
0 2217 709 23.57 58.08 37.57352	<u>a</u>	344419421.jpg			× •	-158.32	0	45.18	-97.10325
709 37.57352		e			1.07%				0.64%
New prediction name	0	0		2217	709	23.57	0	58.08	37.57352
New prediction name									
bluebird-gps-tagged  Predict									Des Part

Figure 5.68: Deepnet image prediction form, more fields

Non-selected fields will be considered as having missing values during the prediction.

#### 5.6.2.3 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the deepnet you want to use to make predictions and a dataset containing the instances you want to predict. BigML will create a prediction for each instance in the dataset.

Follow these steps to create a batch prediction:

1. Click on BATCH PREDICTION option under the deepnet 1-click action menu (Figure 5.69)

	••••	Diabetes of	diagnosis   Train	ning (80%) 📋 🗘 🤇	• (≡) [*] • (
HIDDEN LAYERS	ALGORITHM	LEARNING RATE	MISSING NUMER		VCES
5	ADAM	0.01	TRUE	BATCH PREDICTION	4
				EVALUATE	
				K DELETE DEEPNET	C
				MOVE TO	
Y Age	123 🔻		£3		

Figure 5.69: Create batch prediction using 1-click action menu

Alternatively, click on CREATE BATCH PREDICTION in the **pop up menu** of the list view (Figure 5.70).

			K	Deepnets				Ô	Q
ill	Name	≎ Ty	pe	Objective	₩ ≎	<b>Å</b> \$	\$	\$	
ıl <mark>î</mark>	Digit recognizer (train)   Training (80%) 1 hidden layers, adam, learning rate=0.001, 3873		10	label	5h 55min	58.6 MB	0	0	0
ılî	1 hidden laye		10	Injury Severity	5h 59min	578.3 KB	1		
ılî	Digit recogni CREATE BATCH PREDICTION 1 hidden laye BB EVALUATE		0	label	6h 7min	58.6 MB			
ılî	Digit recogni 1 hidden layer		0	label	6h 10min	58.6 MB			
ılî	US car accid		0	Injury Severity	6h 13min	578.3 KB			1
ılî	1 hidden layer US car accid 1 hidden layers, adam, learning rate=0.001, 993 i	2	0	Injury Severity	6h 36min	578.3 KB			1
ılî	US car accidents_missings   Training (80%) 1 hidden layers, adam, learning rate=0.001, 1057	0	0	Injury Severity	6h 37min	578.3 KB			
ılî	US car accidents   Training (80%) 1 hidden layers, rms_prop, learning rate=0.001, 2		٨	Crash Date.day-of-we	6h 40min	592.8 KB			
ılî	US car accidents   Training (80%) 1 hidden layers, momentum, learning rate=0.04,	0	0	Injury Severity	6h 41 min	592.8 KB	1		
ılî	US car accidents   Training (80%) 1 hidden layers, ftri, learning rate=0.001, 177 iter		0	Injury Severity	6h 43min	592.8 KB			

Figure 5.70: Create batch prediction using pop up menu

2. Select the dataset containing all the instances you want to predict. The instances should contain the input values for the fields used by the deepnet as input fields. From this view you can also select another deepnet from the selector. (See Figure 5.71.)

8 17		New Batc	h Prediction		<b>5</b> -
£1 £61 ≤					
Digit recognizer	(train)   Training (80%	) × ×		earch dataset	*
🛯 🕹 Iabel		Thu, 28 Sep 2017 10:04:00	[	liab Diabetes diagnosis   Test (20%) wit 154 instances, 11 fields (2 categorical	
58.6 MB size	155 fields	33,600 instances		Diabetes diagnosis   Test (20%) with 54 instances, 12 fields (3 categorical)	°
Description:		Diabetes diagnosis   Test (20		Diabetes diagnosis   Test (20%) wit 154 instances, 11 fields (3 categorical	•
	154 i	nstances, 10 fields, 1 non-prefen rate: 0.2, out of bag	red, sample 🔀		ad completetter 0

Figure 5.71: Select dataset for batch prediction

3. After the deepnet and the dataset are selected, the batch prediction **configuration options** will appear along with a **preview** of the prediction output (a CSV file). (See Figure 5.72.) The default output format includes all your prediction dataset fields and adds an extra column with the class predicted. See Subsection 5.6.3 for a detailed explanation of all configuration options.

4	New B	atch Prediction	ۍ. (
£1 6€1 ≤	X ×		
Diabetes diagno	osis   Training (80%) x v	Diabetes diagnosis   Test (20%)	×
💐 🛛 🖨 Diabete	S Wed, 27 Sep 2017 10:54:11	I	Mon, 03 Jul 2017 15:06:45
21.6 KB size	9 614 fields instances	5.4 KB 10 size fields	154 instances
Description:	$\odot$	Description:	$\odot$
Configure			$\overline{\bigcirc}$
	file (using the type of each field)		Ŭ
Preview of the prediction Pregnancies, Glucose 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123, 123,	file (using the type of each field) Blood pressure, Skinfold, Insulin, BMI, D 123, 123, 123, ABC, ABC, ABC 123, 123, 123, ABC, ABC, ABC 123, 123, 123, 123, ABC, ABC, ABC 123, 123, 123, ABC, ABC, ABC 123, 123, 123, ABC, ABC, ABC	iabetes pedigree,Age,Diabetes,weight,	

Figure 5.72: Configuration options for deepnet batch prediction

4. By default, BigML generates an output dataset with your batch predictions that you can later find in your datasets section in the BigML Dashboard. This option is active by default but you can deactivate it by clicking on the icon shown in Figure 5.73.

1 12	New Bate	ch Prediction		<b>5</b> - (i
£1 Fa ≤ 1 ≤				
Diabetes diagnosis   Training (80%)	× v	Diabetes diag	nosis   Test (20%)	×
Second Diabetes	/ed, 27 Sep 2017 10:54:11	ıl	Ν	lon, 03 Jul 2017 15:06:45
21.6 KB 9 size fields	614 instances	5.4 KB size	10 fields	154 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
Configure Preview of the prediction file (using the type of eac	sh field)			⊚
Pregnancies, Glucose, Blood pressure, Ski 123, 123, 123, 123, 123, 123, 123, 123,	C,ABC C,ABC C,ABC C,ABC C,ABC	petes pedigree,Age,Diab	etes,weight,	
	_			

Figure 5.73: Create a dataset from batch prediction

5. After you configure your batch prediction, click on the green button Predict to generate your batch prediction. (See Figure 5.74.)

A 77	New	Batch Prediction	<b>5</b> - (i)
<b>ଛା</b> ଜିଲ <b>≋</b>	XX		
Diabetes diagnosis	Training (80%) × v	Diabetes diagnosis   Test	t (20%) × *
Diabetes	Wed, 27 Sep 2017 10:54:	1	Mon, 03 Jul 2017 15:06:45
21.6 KB size	9 614 fields instances		10 154 elds instances
Description:	⊚	Description:	$\overline{\mathbf{O}}$
Configure			$\overline{\mathbf{O}}$
Preview of the prediction file Pregnancies, Glucose, Blo 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123	Dod pressure,Skinfold,Insulin,BM 3,123,123,ABC,ABC,ABC 3,123,123,ABC,ABC,ABC 3,123,123,ABC,ABC,ABC 3,123,123,ABC,ABC,ABC	.,Diabetes pedigree,Age,Diabetes,weig	iht,
Prediction name: Diabetes diagnosis   Test (2)	0%) with Diabetes diagnosis   Trainii	di	Reset

Figure 5.74: Create batch prediction

6. When the batch prediction is created, you will be able to **download the CSV file** containing all your dataset instances along with a prediction for each one of them. (See Figure 5.75.)

Sources Datasets Supervised  Unsupervised	Predictions	NEW Tasks	WhizzML 🔻
A Diabetes dia	ignosis   Tes	t (20%) with Diabetes diagno	<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li></ul>
Diabetes Diagnosis   Training (80%)	◙	Diabetes Diagnosis   Test (20%)	$\odot$
Configuration			$\overline{\mathbf{O}}$
Output preview			
Pregnancies, Glucose, Blood pressure, Skinfold, Insu 8,183,64,0,0,23.3,0.672,32,True,1,True 10,168,74,0,0,38.0,0.537,34,True,1,True 7,107,74,0,0,29.6,0.254,31,True,1,False 1,103,30,38,83,43.3,0.183,33,False,1,False 3,126,88,41,235,39.3,0.704,27,False,1,False 9,102,76,37,0,32.9,0.665,46,True,1,False 2,71,70,27,0,28.0,0.586,22,False,1,False 7.103,66.32.0.39.1.0.344.31.True.1.False	lin,BMI,Diabe	tes pedigree,Age,Diabetes,weight,Diabetes	
Download batch prediction		UP Output dataset	

Figure 5.75: Download batch prediction CSV file

7. If you did not disable the option to create a dataset explained in step 4, you will also be able to access the **output dataset** from the batch prediction view. (See Figure 5.76.)

ources	Datasets	Supervised 🗸 L	Jnsupervised 🔻	Predictions	Tasks				WhizzML 🔻
¥.	••••	Ē	Diabetes dia	ignosis   Tes	t (20%) with	Diabetes di	agno	∽.	(≡)⁵- (ĵ
×,	Diabetes I	Diagnosis   Train	ing (80%)	◙		abetes Dia	gnosis   Test (20%)	)	◙
Configura	ation								◙
8,183,64 10,168,7 7,107,74 1,103,30 3,126,88 9,102,70 2,71,70	cies,Gluco 4,0,0,23.3 74,0,0,38.4 4,0,0,29.6 0,38,83,43 8,41,235,3 6,37,0,32.	se,Blood pressure ,0.672,32,True,1, ,0.0537,34,True,1, ,0.254,31,True,1, .3,0.183,33,False .3,0.704,27,False ,0.665,46,True,1 0.6586,22,False,1 1.0.344,31,True,1	True .,True False e,1,False .,False .,False	lin,BMI,Diabe	etes pedigree	,Age,Diabete	s,weight,Diabetes		
	Ŧ	Download batch p	rediction			۵î	Output dataset		

Figure 5.76: Batch prediction output dataset

#### 5.6.2.3.1 Batch Predictions with Images

Deepnet created from a dataset containing images is a convolutionary neural network (CNN) (See Subsection 5.2.1). Because BigML treats images as an input type just like any other data types such as numeric and text, creating batch predictions using CNN is the same as other deepnets.

The input of a batch prediction is a dataset. The dataset contains multiple images which was created by a composite source.

		New Batch Pr	rediction	<b>5</b> - (
කි 🕰				
grape-strawbe	rry	x v	Search dataset	*
			predict	٩
💐 💵 🕀 label	6	predict-images instances, 236 fields (234 numeric, 1 pa	ath, 1 predict-images	
13.4 MB	2	image), 1 non-preferred	6 instances, 236 fields (234 num	eric, 1 path, 1 image), 1 n
size	Z fields	instances	iris dataset - extended - samp 60 instances, 142 fields	le (100.00%) - extende
Description:		$\odot$	iris dataset - extended - samp 60 instances, 139 fields	le (100.00%) - extende
			iris dataset - extended - samp 60 instances, 136 fields	le (100.00%) - extende
Prediction name:			iris dataset - extended - samp	le (100.00%) - extende

Figure 5.77: Batch prediction using an image dataset

As shown in Figure 5.77, the input for the deepnet batch prediction is selected as predict-images, which is a dataset consisting of six images.

Everything stated earlier in current section (Subsection 5.6.2.3) applies.

## 5.6.3 Configuring Deepnet Predictions

BigML provides several options to configure your batch predictions such as setting a probability threshold Subsection 5.6.3.1), default values for your missing numeric values (see Subsection 5.6.3.2), fields mapping (see Subsection 5.6.3.3), and output file settings (see Subsection 5.6.3.4.)

#### 5.6.3.1 Probability threshold

Probability thresholds usually makes sense when you want to minimize false positives at the cost of false negatives. The positive class will be predicted if its probability is greater than the given threshold, otherwise the following class with greater probability will be predicted instead. This option is only available for classification deepnets.

To configure a threshold for your batch prediction follow these steps:

1. Select the positive class, i.e., the class for which you want to apply the threshold (Figure 5.78).

Sources Datasets	Supervised - Unsupervis	ed - Predict	ons - Tasks	WhizzML 👻
¥.	Pred	ict using US ca	ır accidents dataset - saı	m (≞)⁵- ③
	Injur	y Severity: ?		?
Select a positive class  A Fatal Injury (K)	Probability threshold:		50%	All input fields: 🕑
Injured, Severity U No Injury (O) Non-incapacitating Possible Injury (C) Unknown	capacitating Injury (A)	•	Fatalities in crash 0 Age 0	6 3 122
Alcohol Results	0.59	0.29	Person Type Bicyclist	61 ©
Drug Involvement		v.	Race All other races	<b>S</b>
Gender Female		<b>~</b>	Crash Date.month	14 <b>7</b>

Figure 5.78: Select the positive class

Sources Datasets Supervised - Unsupervise	d - Predicti	ons 👻 Tasks	WhizzML 👻
Predic	t using US ca	r accidents dataset - sam	. (=) ^{\$} , ()
Injury	Severity: ?	?	
Incapacitating Injury ×   Probability threshold:	9	10%	
			All input fields:
Atmospheric Condition	<b></b>	Fatalities in crash	6
Blowing Sand, Soil, Dirt	v	°	3
Roadway	<b>S</b>	Age	<b>S</b>
Rural-Local Road or Street	Ŧ	0	122 <b>61</b>
Alcohol Results		Person Type	
0.0 0.59	0.29	Bicyclist	٣
Drug Involvement		Race	
No	v	All other races	Ψ.

2. Set a probability threshold using the slider shown in Figure 5.79.

Figure 5.79: Set a probability threshold

3. Click Predict . If the positive class probability is greater than the given threshold, it will be predicted, otherwise the following class with greater probability will be predicted instead.

Sources Datasets	Supervised -	Unsupervised - Prec	dictions - Tasks	_	WhizzML 👻
8 77-		US car accidents	dataset - sample (8	30.00%)	<b>८५</b> → (≡) [€] → (i)
		Injury Severity: In	capacitating Injury	(A) <b>41.33%</b>	e
	8.48%	41.33% 0.51% 14. 14. 14. 14. 14. 14. 14. 14.		94% 0.53%	
Incapacitating Injury × 1	Probability th	hreshold:	10%		All input fields: 🖃
Incapacitating Injury × Atmospheric Condition	Probability th	0	Fatalities in crash		· _
	Probability th	0	Fatalities in crash		
Atmospheric Condition	Probability th	6	Fatalities in crash		
Atmospheric Condition Blowing Sand, Soil, Dirt	Probability th	6	<ul> <li>Fatalities in crash</li> <li>0</li> <li>Age</li> <li>0</li> </ul>		6 3
Atmospheric Condition Blowing Sand, Soil, Dirt Roadway	Probability th		<ul> <li>Fatalities in crash</li> <li>0</li> <li>Age</li> <li>0</li> </ul>		6 3 122

Figure 5.80: Save the prediction

You can also find the same options to set a threshold for batch predictions under the CONFIGURE panel (see Figure 5.81).

Sources Datasets Supervised - Unsupervised - Predictions - Tasks	WhizzML 👻
New Batch Prediction	۵- (آ
n fn <u>/</u> ≤	
US car accidents dataset - sample (80.00%) × ×	(20%) × *
∞         □         □         Injury Severity         Wed, 27 Dec 2017 16:24:07	Tue, 26 Dec 2017 14:23:09
592.5 KB         16         3,999         148.1 KB         20           size         fields         instances         size         fields	1,000 instances
Description: 🕤 Description:	$\odot$
್ಲೈ Configure	۲
Select a positive class    Probability threshold: 50%	
Default numeric value: ③ Select a default value *	
Excluded fields:	$\odot$
1 Make sure that the fields in the Deepnet and Dataset match	
Fields mapping:	Default fields 🕞

Figure 5.81: Configure a probability threhsold for batch predictions

#### 5.6.3.2 Default Numeric Value

If the dataset used to make the batch prediction contains instances with missing values for the numeric fields, the prediction will not be computed for them, unless you built the deepnet enabling the **Missing numerics** parameter (see Subsection 5.4.4).

By using the **Default numeric value** before creating your batch prediction, you can easily replace all the missing numeric values by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See Figure 5.82.)



Figure 5.82: Configure Default numeric value for batch prediction

#### 5.6.3.3 Fields Mapping

You can specify which input fields of the deepnet match with the fields in the dataset contaning the instances you want to predict. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** by clicking on the green switcher. Additionally, you can **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the batch prediction. (See Figure 5.83.)

Configure		(
Select a positive class   Probability threshol	old: 50%	
Default numeric value: 😢 Select a default value		
Excluded fields:		⊽
1 Make sure that the fields in the Deepnet and D		
1 Make sure that the fields in the Deepnet and D Fields mapping: Deepnet fields	Dataset match Apply this map Dataset fields Auto-mapping by: NAME Composition	۲
Fields mapping:	Apply this map	۲
Fields mapping: Deepnet fields	Apply this map Dataset fields Auto-mapping by: NAME	٢
Fields mapping: Deepnet fields 1. Pregnancies	Apply this map Dataset fields Auto-mapping by: NAME CONTINUE 123 1. Pregnancies 123 × V	۲
Fields mapping: Deepnet fields 1. Pregnancies 2. Glucose	Apply this map         Dataset fields       Auto-mapping by:       NAME       ID         123       1.       Pregnancies       123       ×       ×         123       2.       Glucose       123       ×       ×	٢

Figure 5.83: Configure the fields mapping for batch prediction

# Note: Fields mapping from the BigML Dashboard is limited to 200 fields. For batch predictions with a higher number of fields, map your fields using the **BigML API**¹⁸.

#### 5.6.3.4 Output Settings

Batch predictions return a CSV file containing all your instances and the final predictions. Tune the following settings to customize your prediction file (see Figure 5.84):

- **Separator**: this option allows you to choose the best separator for your output file columns. The default separator is the comma. You can also select the semicolon, the tab, or the space.
- New line: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".
- Output fields: by clicking on the list icon next to the separator selector, you can include or exclude all your dataset fields from your output file. You can also individually select the fields you want to include or exclude using the multiple output fields selector. Note: a maximum of 100 fields can be displayed in this selector, but all your dataset fields will be included in the output file by default unless you exclude them.
- **Headers**: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability column name, etc.). By default, BigML includes the headers.
- **Prediction column name**: customize the name for your predictions column. By default, BigML takes the name of the deepnet's objective field.
- **Probability**: this option allows you to include an additional column with the probability for the predicted class. By default it is not included in your ouput file. For **regression** deepnets, you will find the expected error instead of the probability.
- **Probability column name**: customize the name for the probability column if you include it in the output file. BigML sets "probability" as the default name. For **regression** deepnets, you will find the expected error column name.
- All class probabilities: this includes all the probabilities of the objective field classes per instance. This option will add *n* extra columns, one by class in the objective field. This option does not exist for **regression** deepnets.

¹⁸https://bigml.com/api/batchpredictions#bp_batch_prediction_arguments

1 Make sure that the fields i	in the <b>Deepnet</b> and	Dataset match				
Fields mapping:					Default fields	6
2 [OPTIONAL] Customize p	prediction output se	ottings				
Output settings						e
Separator:	New line					
, (comma)	• Unix, Lin	ux or OS X (LF)	\$≡			
Prediction column na	me: Pro	bability column name:				
a, b, c -, -, -	-					
Output Fields:						
* Pregnancies	123	⊮ Glucose	123		12	3
* Skinfold	123	× Insulin	123	⊮ BMI	12	3
× Diabetes pedigree	123	* Age	123	* Diabetes	AB	

Figure 5.84: Deepnet output settings for batch predictions

## 5.6.4 Visualizing Deepnet Predictions

Deepnet predictions visualization changes depending if you are predicting one **single** instance (Subsection 5.6.4.1), or you are predicting multiple instances using the **batch predictions** option (Subsection 5.6.4.2).

#### 5.6.4.1 Single Predictions

For single predictions, find the predicted class given the input fields values at the top of the form along with its probability. (See Figure 5.85.)

Sources Datasets Supervised Vinsupervis	ed - Predictions	Tasks	WhizzML 🔻
≗ ¥.	car accidents_m	nissings   Training (80%)	
	Injury Severity:	Incapacitating Injury (A)	58.55%         Probability         Image: Comparison of the second
Select a positive class		50%	All input fields: 🗖
Atmospheric Condition	<b>S</b>	Fatalities in crash	<b></b>
Blowing Sand, Soil, Dirt	v	0	6 <b>3</b>
Roadway	۲	Age	€
Rural-Local Road or Street	v	0	122 <b>61</b>
Alcohol Results		Person Type	

Figure 5.85: Deepnet single prediction

Below the prediction, there's a histogram representing the rest of the objective field **class probabilities**. All the class probabilities must sum 100%. Show or hide this view by clicking on the icon highlighted in Figure 5.86. You can see up to seven different classes at the same time; if you have more than seven classes, you can see the others by clicking on the **arrows** icons. **Export** this view in PNG format, in a CSV file, or in a JSON file by clicking on the corresponding icons. (See Figure 5.86.)

ources Datasets	Supervised 🗸 U	Insupervised -	Predictions	Tasks			WhizzML
- ¥-		📰 US car ac	cidents_m	issings   Training (8	30%)		(≣)*, (
		Injury	y Severity:	Incapacitating Inju	ury (A)	58.55%	
	18.14%	58.55% 0.48%	1.05%	15.93% 4.93%	0.0170		
Select a positive class	<ul> <li>Probability the</li> </ul>	reshold:	0	50%		A	ll input fields: 🗧
Atmospheric Condition				Fatalities in crash			٩
Blowing Sand, Soil, Dirt			*	0	-6-	6	3
Roadway			<b>S</b>	Age			٦
	t		*	0	0	122	61
Rural-Local Road or Street							
Rural-Local Road or Street				Person Type			C

Figure 5.86: Deepnet all class probabilities

For **regression** deepnets, instead of the class probabilities you will get the predicted numeric value for the objective field.

#### 5.6.4.1.1 Prediction explanation

Prediction explanation helps understand why a deepnet makes a certain prediction. This is very useful in many applications, and the reasons behind a deepnet's prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

For any classification or regression deepnet, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Predict (see Figure 5.87).

Sources	Datasets	Supervised <del>-</del>	Unsupervised -	Prediction	ns 👻 Tasks	WhizzM	
-			Predict	using Ficti	onal Wine Sales v1	<b>€</b> 5- (≡) ⁵ -	(
			Total Sa	ales: ?	?		
						All input fields:	<b></b>
Country				2	Grape		V
Argentina				¥	Cabernet Sauvignon		Ŧ
Rating				€	From Oregon?		۲
88		0	92	90	No		Ŧ
Price				<b></b>			
0.0		0	42.0	20.4			
					Click to co	mpute the explanation	
New predic	tion name					with the prediction	
Fictional )	Wine Sales v	1				Predict	

Figure 5.87: Explain prediction

The prediction explanation represents the most important factors considered by the deepnet in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 5.88. The importances across all input fields should sum 100%.

Sources Dat	tasets Super	vised - Unsupervised -	Predictions -	Tasks	WhizzML 🗸
8 4.		<b>F</b>	ictional Wine Sa	lles v1	(≡) [*] , (≡)
		Total Sal	es: 73.01		
PREDICTION EXPLA	ANATION				
Input data		Importance			
Price	20.40				59.27% +
Rating	90.00				36.58% +
Grape	Cabernet S				3.29%
From Oregon?	No				0.00%
Country	Argentina				0.00%



For some input fields you will see a "+" icon next to the importance. This is because the importance may not be directly associated with the input value, i.e., it can be explained by other reasons. In the Figure 5.89 below, the importance of 36.58% for the field "Rating" is not explained by this field being equal to 90.00. Rather, it is because this field value is not missing.

Sources	Datasets	Supervised -	Unsupervised <del>-</del>	Predictions -	Tasks	v	VhizzML 👻
8	₽.		F	ictional Wine Sa	iles v1	<b>5</b> -	(≡)∕~
			Total Sal	es: 73.01			
PREDICTIC	ON EXPLANATION					PNG	CSV JSO
nput data		Impor	rtance			•	
Price	20	0.40					59.27% +
Rating	90	0.00					36.58% +
Grape	Caber	net S				Detailed Explanation	
From Oreg	gon?	No				!= missing	36.58%
Country	Arge	entina					0.00%
Country	Arge	entina					0

Figure 5.89: See the detailed explanation

The prediction explanation for deepnets is calculated using the results of over a thousand distinct predictions using random perturbations of the input data. For this reason, the calculation of the explanation may take some time to be computed.

Note: the input field importances in the prediction explanation are different from the overall field importances of the deepnet. A field can be very important for the deepnet but insignificant for a given prediction.

#### 5.6.4.2 Batch Prediction

After creating your batch prediction, you get a **CSV file** and, optionally, an **output dataset**. Both outputs are explained in the following subsections.

#### 5.6.4.2.1 Output CSV file

The batch prediction generates a CSV file containing your **predictions** for each of your dataset instances in the last column. (See Figure 5.90.)

Durces Datasets Supervised Vursupervised Predictions Tasks Whi	zzML 🔻
Diabetes diagnosis   Test (20%) with Diabetes diagno 🦽 (=)	۰.
Diabetes Diagnosis   Training (80%)	◙
Configuration	ਂ
Output preview	
Pregnancies, Glucose, Blood pressure, Skinfold, Insulin, BMI, Diabetes pedigree, Age, Diabetes, weight, Diabetes 8, 183, 64, 0, 0, 23. 3, 0. 672, 32, True, 1, True 10, 168, 74, 0, 0, 38. 0, 0. 537, 34, True, 1, True 7, 107, 74, 0, 0, 29. 66, 0. 254, 31, True, 1, False 1, 103, 30, 38, 83, 43. 3, 0. 183, 33, False, 1, False 3, 126, 88, 41, 235, 39. 3, 0. 704, 27, False, 1, False 9, 102, 76, 37, 0, 32. 9, 0. 665, 46, True, 1, False 2, 71, 70, 27, 70, 28. 0, 0. 586, 22, False, 1, False 7, 103, 26, 32. 0, 39. 1, 0. 344, 31, True, 1, False	
Download batch prediction     Output dataset	

Figure 5.90: Download batch prediction CSV file

You can configure several options to **customize your CSV file**. You can find a detailed explanation of those options in <u>Subsection 5.6.3.4</u>.

See an output CSV file example in Figure 5.91. The column **class** in this example contains the final prediction (it is named by default as your deepnet's objective field). In this case we are predicting whether a person is a good or a bad candidate for holding a credit. This file has been configured to contain also the **probability** for each prediction.

```
duration,age,amount,purpose,class,probability
24,26,5433,used car,good,0.88785
36,42,8086,new car,bad,0.55526
24,28,1376,radio/tv,good,0.8385
48,31,6758,radio/tv,bad,0.73576
26,30,7966,used car,good,0.7201
12,42,2577,furniture/equipment,good,0.67644
36,30,4455,business,good,0.52227
18,32,1442,new car,bad,0.75488
9,22,276,new car,good,0.57819
```

Figure 5.91: An example of a deepnet batch prediction CSV file

#### 5.6.4.2.2 Output Dataset

By default BigML automatically creates a dataset out of your batch prediction. You can disable this option by configuring your batch prediction. (See Subsection 5.6.3.4.) You will find the output dataset in your batch prediction view as shown in Figure 5.92.

ources Datasets Su	pervised - Unsupervised -	Predictions -	Tasks		WhizzML 🔻
¥	Diabetes diag	gnosis   Test	t (20%) with Diabetes	diagno	ૐ- (≡)⁵- 0
Diabetes Diag	gnosis   Training (80%)	◙	Diabetes Di	agnosis   Test (20%)	$\odot$
Configuration					$\overline{\mathbf{O}}$
8,183,64,0,0,23.3,0.6 10,168,74,0,0,38.0,0. 7,107,74,0,0,29.6,0.2 1,103,30,38,83,43.3,0	537,34,True,1,True 54,31,True,1,False 0.183,33,False,1,False 0.704,27,False,1,False 665,46,True,1,False 86,22,False,1,False	in,BMI,Diabe	rtes pedigree,Age,Diabe	tes,weight,Diabetes	

Figure 5.92: Batch prediction output dataset

In the output dataset, you can find an additional **field** (named by default as your deepnet's objective field) containing the **class predicted** for each one of your instances. If you configured your batch prediction to include the prediction **probabilities** and **all class probabilites** you will be able to find them in the last fields of your output dataset. (See Figure 5.93.)

Sources Datasets Si	upervised - Unsupervised		ctions - Tasks				WhizzML 🔻
	Diabet	tes diagn	osis   Test (20%	%) with Dial	oetes	\$. <b>6</b> .	(≡)⁵-⊂ 🚺
					c	٦	×
Name	≎ Type :	Count	Missing	Errors	Histogram		
Pregnancies	123	154	0	0			-
Glucose	123	154	0	0			d
Blood pressure	123	154	0	0	dl	lilin	a
Skinfold	123	154	0	0		ullillu	
Insulin	123	154	0	0	6.01600		
BMI	123	154	0	0	dl		
Diabetes pedigree	123	154	0	0	Ulhau	ulluu	
Age	123	154	0	0	llu		
Diabetes	ABC	154	0	0			
Diabetes	ABC	154	0	0			
probability	123	154	0	0	nationally	munullill	

Figure 5.93: Deepnet batch prediction output dataset

## 5.6.5 Consuming Deepnet Predictions

You can fully used single and batch predictions via the BigML API and bindings. The following subsections explain both tools.

#### 5.6.5.1 Using Deepnet Predictions via the BigML API

Deepnet predictions have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, and delete single and batch predictions.

In the example below, see how to create a single prediction using a deepnet and defining the input data once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"deepnet": "deepnet/50650bdf3c19201b64000020",
                          "input_data": {"000001": 3, "000002":4.5, "000003":5}}}'
```

For more information on using predictions through the BigML API, please refer to the documentation¹⁹.

#### 5.6.5.2 Using Deepnet Predictions via BigML Bindings

You can also create, configure, retrieve, list, update, and delete single and batch predictions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice.

¹⁹https://bigml.com/api/predictions

BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a deepnet with the Python bindings.

```
from bigml.api import BigML
api = BigML()
prediction = api.create_prediction(
    "deepnet/50650bdf3c19201b64000020",
    {"credit_amount": 5, "duration": 2.5})
```

For more information on BigML bindings, please refer to the bindings page²⁰.

### 5.6.6 Descriptive Information

Each deepnet prediction has an associated **name**, **description**, **category**, and **tags**. You can find a brief description of each concept in the following subsections. The MORE INFO menu option displays a panel that provides editing options. (See Figure 5.94.)

urces Datasets Supervised  Unsupervised  Predicto	ons Tasks		Whizz№	1L <b>-</b>
Tr ••••• Diabetes diagnosis	Test (20%) with Diabetes diagno	∽.	(≡)⁵ָ	6
ETAILS				◙
IFO				۲
Vame: Diabetes diagnosis   Test (20%) with Diabetes diagnosis   Training (8	Category: Healthcare	T		
Description: Patient data to predict diabetes	Tags: health predictive diagnosis diabetes			,

Figure 5.94: Deepnet prediction descriptive information

#### 5.6.6.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: the name always follows the structure "<deepnet name>".
- **Batch predictions**: BigML combines your prediction dataset name and the deepnet name: "<deepnet name> with <dataset name>".

Prediction names are displayed on the list and also on the top bar of a prediction view. Prediction names are indexed to be used in searches. Rename your predictions any time from the MORE INFO menu.

The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, but they will always have different identifiers.

²⁰https://bigml.com/tools/bindings
#### 5.6.6.2 Description

Each prediction also has a **description** that it is very useful for documenting your Machine Learning projects. Predictions take their description from the deepnet used to create them.

Descriptions can be written using plain text and also markdown²¹. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 5.95.)

You can add formatting and links us	ing a simple markdown language:	
$B I \equiv O \leftrightarrow$		
Patient data to predict d	iabetes	
Description:		
Patient data to predict diabetes		

Figure 5.95: Markdown editor for deepnet descriptions

Descriptions cannot be longer than **8192** characters.

#### 5.6.6.3 Category

A **category** taken from the deepnet used to create it is associated with each prediction. Categories are useful to classify predictions according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 5.1.

#### 5.6.6.4 Tags

A prediction can also have a number of **tags** associated with it. These tags help to retrieve the prediction via the BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the deepnet used to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

#### 5.6.7 Deepnet Predictions Privacy

The link displayed in the **Privacy** panel is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared by using a secret link. (See Figure 5.96.)

²¹https://en.wikipedia.org/wiki/Markdown

ources Dataset	Supervised  Unsupervised  Predictions  Tasks Wi	nizzML 🔻
	Churn in the teltry   Test (20%) with Churn in the tele (5-	=) <b>%</b>
DETAILS		$\overline{\mathbf{O}}$
INFO		$\overline{\mathbf{O}}$
PRIVACY		۲
Private link   Only y	ou can view this batchprediction logged into your account.	
	https://tropo.dev.bigml.com/dashboard/batchprediction/59c8e8f2b95b39796c00000a	

Figure 5.96: Deepnet predictions privacy

# 5.6.8 Moving Deepnet Predictions to Another Project

When you create a prediction, it will be assigned to the same **project** where the original deepnet is located. You cannot move predictions between projects as you do with other resources.

# 5.6.9 Stopping Deepnet Predictions

Single predictions are **synchronous** resources, so you cannot cancel them during the creation since you get the result immediately.

By contrast, batch predictions are **asynchronous** resources, so you can stop their creation before the task is finished. Use the DELETE BATCH PREDICTION option from the **1-click action menu** (Figure 5.97) or from the **pop up menu** on the list view.



Figure 5.97: Stop deepnet batch prediction from 1-click action menu

A modal window will be displayed asking you for confirmation. If you stop the prediction during its creation you won't be able to resume the same task again, so if you want to create the same prediction you will have to start a new task.

5	Are you sure you want to delete this prediction?	
	If you delete this prediction, you will no longer have access to its data, and you will need to recreate it.	
Th	Cancel Delete	tr

Figure 5.98: Deepnet delete prediction confirmation

# 5.6.10 Deleting Deepnet Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the **1-click action menu** (see Figure 5.99) or using the **pop up menu** on the predictions list view (see Figure 5.100).



Figure 5.99: Deepnet delete prediction from 1-click menu

Source	es Di	atasets Supervised  Vinsupervised  V			WhizzML 👻			
ŧ	F		Batch Predictions				Q	∽-
ជា	ill	Name		\$	Objective	\$	Instances 💠	÷
×	ılî	Churn in the teltry   Test (20%) with Churr use all fields	in the telecom industry   Training (80%)	۲	Churn		667	4d 19h
R	ıl <mark>o</mark>	Iris   Test (20%) with Iris   Training (80%) use all fields	VIEW DETAILS		Species		30	1w 6d
Ś	ılî	<b>Iris - missing</b> default numeric value=zero, use all fields	DELETE PREDICTION		Species		30	2w 5d
2	ılı	<b>Iris - missing</b> default numeric value=zero, use all fields			Species		30	2w 5d
~	ılı	Iris dataset show probabilities, use all fields			Species		30	2w 5d

Figure 5.100: Deepnet delete prediction from pop up menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

s	Are you sure you want to delete this prediction?	L
	If you delete this prediction, you will no longer have access to its data, and you will need to recreate it.	
Th	Cancel Delete	tr

Figure 5.101: Deepnet delete prediction confirmation

# 5.7 Consuming Deepnet

Similarly to other models in BigML, deepnets are white-boxed models, so you can **download** them and used them locally to make predictions. You can also create and consume your deepnet programmatically via the **BigML API and bindings**. The following subsections explain those three options.

# 5.7.1 Downloading Deepnet

You can download your deepnet in several programming languages including Python or Node.js. By downloading your deepnet you will be able to compute **predictions locally**, free of latency and at no cost. Click on the download icon in the top menu (see Figure 5.102), and select your preferred option (see Figure 5.103.)

8 7. 8.	•••• \$	Diabetes diag	nosis dataset   Training	(80%)	ົ ໕. (≡) ⁵ (
HIDDEN LAYERS 3	ALGORITHM ADAM	LEARNING RATE	MISSING NUMERICS	OBJECTIVE FIELD DIABETES	INSTANCES 614
					S
<b>Ү</b> ВМІ	123 *	44.0340	£3	PREDICTIO	esc Releases vi
			True		Probability
60 -			50.7%		100%

Figure 5.102: Click download icon

0	PROJECT: All	o	
Sources Datas	Actionable Deepnet Download	×	
	Complete actionable deepnet:		
	Python x *		
HIDDEN LAYERS	Python		
3	Node.js Ings		614
	# # Install via: pip install bigml #		0
<b>Y</b> ВМІ	<pre># or clone it: # git clone https://github.com/bigmlcom/python.git</pre>		
Divit	<pre>from bigml.deepnet import Deepnet</pre>		
	<pre>from bigml.api import BigML # Downloads and generates a local version of the DEEPNET,</pre>		
60 -	<pre># if it hasn't been downloaded previously. deepnet = Deepnet('deepnet/59d5444db95b396d7100015d',</pre>		100% TOTAL
50 -	api=BigML("teresa",		
40			
	Clos	se	
30 -			

Figure 5.103: Select language to download deepnet

# 5.7.2 Using Deepnets Via the BigML API

Deepnets have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them for predictions.

In the below example, see how to create a deepnet using an existing dataset once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/deepnet?\$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"dataset": "dataset/50650bdf3c19201b64000020"}'
```

For more information on using deepnet through the BigML API, please refer to the documentation²².

## 5.7.3 Using Deepnets Via the BigML Bindings

You can also create and use deepnets via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a deepnet with the Python bindings.

```
from bigml.api import BigML
api = BigML()
deepnets = api.create_deepnet(
    'dataset/57506c472275c1666b004b10', {"objective_field":"churn"})
```

For more information on BigML bindings, please refer to the bindings page²³.

²² https://bigml.com/api/deepnets

²³https://bigml.com/tools/bindings

# 5.8 Deepnet Limits

Some specific limits apply for your deepnets regarding your **dataset field characteristics** (see Subsection 5.8.0.1) and the **visualization**, i.e. to the deepnet Partial Dependence Plot, depending on the number of classes in the objective field and the number of input fields in your dataset (see Subsection 5.8.1).

Note: the visualization limits just affect to the visualization of the model, i.e., despite your dataset reach those limits, you can still creating the deepnet, evaluating it and using it to make predictions.

#### 5.8.0.1 Field Limits

Deepnets, similarly to other BigML models, has the following limitations according to the type of field:

- Classes: a maximum number of 1,000 distinct classes per field is allowed.
- **Terms**: BigML can handle up to 1,000 terms in total. If multiple text fields are defined, then the token limit per field is evenly divided by the number of text fields evenly, e.g., a dataset with two text fields would result in 500 terms per text field. BigML selects those terms with most significant frequency, discarding both those that appear either too often or too infrequently. A maximum of 256 characters per term is allowed.
- Items: a maximum number of 10,000 distinct items per field is allowed.

#### 5.8.1 PDP Limits

There are some circumstances under which your chart cannot be displayed:

- As the PDP only supports **numeric and categorical fields** for the axes, if your deepnet **only** contains text, or items fields, the PDP cannot be displayed.
- If your deepnet contains more than **100 fields** the top 100 fields will be included as input fields ranked by importance. The rest of fields will be excluded from the view.
- If your deepnet contains more than **200 categories** in the objective field, the PDP cannot be displayed.

# 5.9 Descriptive Information

Each deepnet has an associated **name**, **description**, **category**, and **tags**. The following subsections provide a brief description for each concept. In Figure 5.104, you can see the options the MORE INFO menu provides to edit them.

Sources Datasets Supervised	Unsupervised  Predictions  Tasks		WhizzML 🔻
≙ \ <b>₹</b> . <mark>8</mark> 7. ••••	Diabetes diagnosis   Training (80%)	â 🗘 👶	(≡)⁵₊ (〕_
DETAILS			$\odot$
INFO			۲
Name: Diabetes diagnosis   Training (80%) Description: Patient data to predict diabetes	Category: Healthcare Tags: health predictive diagno	v osis diabetes	•
PRIVACY			$\overline{\mathbf{v}}$

Figure 5.104: Edit deepnet descriptive information

# 5.9.1 Deepnet Name

Each deepnet has a name that is displayed in the deepnet list view and also on the top bar of the deepnet view. Deepnet's names are indexed to be used in searches. When you create a deepnet, it gets a default name which is your dataset name. Change it using the MORE INFO menu option on the right corner of the deepnet view. The name of a deepnet cannot be longer than **256** characters. More than one deepnet can have the same name even within the same project, but they will always have different identifiers.

## 5.9.2 Deepnet Description

Each deepnet also has a **description** that it is very useful for documenting your Machine Learning projects. Deepnets take the description of the datasets used to create them by default.

Descriptions can be written using plain text and also markdown²⁴. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 5.105.)

²⁴https://en.wikipedia.org/wiki/Markdown

Edit description	2
You can add formatting and links using a simple markdown language: B $~I~\equiv~ \mathscr{O}~ \diamond )$	
Patient data to predict diabetes	
Description:	
Patient data to predict diabetes	

Figure 5.105: Markdown editor for deepnet descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

## 5.9.3 Deepnet Category

A **category**, taken from the dataset used to create it, is associated with each deepnet. Categories are useful to classify deepnet according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

A deepnet category must be one of the 24 categories listed on Table 5.1.

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 5.1: Categories used to classify deepnet by BigML

## 5.9.4 Deepnet Tags

A deepnet can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide deepnet with some extra information. A deepnet inherits the tags from the dataset used to create it. Each tag is limited to a maximum of 128 characters. Each deepnet can have up to 32 different tags.

## 5.9.5 Deepnet Counters

For each deepnet, BigML also stores a number of counters to track the number of other resources that have been created using the deepnet as a starting point. In the deepnet view, you can see a menu option that displays counters for evaluations, single and batch predictions. It also allows you to quickly jump to all the resources of one type. (See Figure 5.106.)

a 🖅-	80 y	•••• 🖉 US car accid	dents dataset - sample (8	30.00%)	
HIDDEN L/	9	VIEW 9 PREDICTIONS FROM THIS DEEPNET	MISSING NUMERICS	OBJECTIVE FIELD	INSTANCES
1	<b>E</b> 1	VIEW 1 BATCH PREDICTION FROM THIS DEEP	TRUE	INJURY SEVERITY	3999
		NO EVALUATIONS FROM THIS DEEPNET YET			
	82.0	NO FUSIONS THAT INCLUDE THIS DEEPNET Y			0

Figure 5.106: Counters for deepnet

# 5.10 Deepnet Privacy

Privacy options for a deepnet can be defined in the **More Info** panel, displayed in Figure 5.107. There are two levels of privacy for BigML deepnets:

- **Private**: only accessible by authorized users (the owner and those who have been granted access by him or her).
- Shared: accessible by any user with whom the owner shares the secret link.

ources	Datasets Supervised	Unsuper	vised 👻 Predictions 👻 Tasks		Whizzk	ΛL ▼
¥.	ເທັ - ••••	X	Diabetes diagnosis   Training (80%)	🏮 🗘 🤹	, (≡) <mark>%</mark> ,	0
DETAILS						⊽
NFO						•
RIVACY						۲
Private link	Only you can view this deepne	et logged into	your account.			
		ht ht	tps://tropo.dev.bigml.com/dashboard/deepnet/59c4fffbb	95b39431b000013		
		k can view th	s deepnet.			
Secret link	Only people with the secret line	K call view ui				
Secret link	Only people with the secret line		tps://tropo.dev.bigml.com/shared/deepnet/gELOUn3rO6	6KHflf1pDmOlnynKS6		

Figure 5.107: Deepnet privacy

# 5.11 Moving Deepnets to Another Project

When you create a deepnet, it will be assigned to the same project where the original dataset is located.

Deepnets can only be assigned to a single project. However, you can move deepnets between projects. The menu option to do this can be found in two places:

1. In the deepnet list view, click the MOVE TO... option within the **1-click action menu** and select another project or create a new one. (See Figure 5.108.)

HIDDEN LAYERS ALGOR	· · · · · · · · · · · · · · · · · · ·		
1 ADA		SING NUMER TRUE BATCH PREDICTION EVALUATE EVALUATE DELETE DEEPNET	
Y Age 123	PROJECT	MOVE TO	

Figure 5.108: Change project from 1-click action menu

2. In the deepnet list view, click the MOVE TO... option within the **pop up menu** and select another project or create a new one. (See Figure 5.109.)

				×	Deepnets				Ô	Q,
ıll	Name		\$	Туре	Objective	÷ #	\$ ♦	\$	\$	00
ıl <mark>e</mark>	Diabetes diagnosi beta1=0.9, beta2=0	<b>s   Training (80%)</b> ).999, epsilon=1e-08, learnin	8	o0	Diabetes	2d 22h	21.6 KB	3	0	0
ıl <mark>o</mark>	Diabetes dia beta1=0.9, be				Diabetes	2d 22h	21.6 KB			
ılı	Happiness 20 beta1=0.9, be			杰	Dystopia.Residual	2d 23h	28.8 KB			
ıl <mark>e</mark>	Diabetes dia beta1=0.9, be			o0	Diabetes	3d 21h	21.6 KB			
ılî	Churn in the max training t				Churp	4⊲ 3h	218.7 KB			
ılî	Happiness su	MOVE TO 0.999, epsilon=1e-08, learnin	_	-	NEW PROJECT	8h	26.0 KB	1		1
ılî	Fictional Wine Sal	<b>es   Training (80%)</b> ).999, epsilon=1e-08, learnin		PROJE	PROJECT	8h	40.3 KB	2		1
ılî		om industry   Training (80%) 0.999, epsilon=1e-08, learnin			Churn	4d 19h	218.7 KB	1	1	1
ılî		in.csv   Training (80%) 1800, missing values		00	label	4d 20h	101.5 MB			
ıß		om industry   Test (20%) 0.999, epsilon=1e-08, learnin		.0	Churn	4d 20h	54.7 KB			1

Figure 5.109: Change project from pop up menu

# 5.12 Stopping Deepnets

You can stop the creation of a deepnet before the task is finished by clicking the DELETE DEEPNET option from the **1-click action menu** (see Figure 5.110), or from the **pop up menu** in the deepnet list view (see Figure 5.111).



Figure 5.110: Stop deepnet creation from 1-click action menu

Sourc	es Datasets Supervised  Unsupervised	•	Predicti	ions 👻 Tasks				WhizzN	∕IL ▼
			×	Deepnets				Ô	Q
ill	Name	\$	Туре	Objective	₩ ≎	<b>ê</b>	\$	\$	
ılî	Churn in the telecom industry   Training (80%) No extra information	8	00	Churn	Omin	218.7 KB	0	0	0
ılî	Diabetes dia predict beta1=0.9, be		00	Diabetes	2d 22h	21.6 KB	3		
ılî	Diabetes dia         CREATE BATCH PREDICTION           beta1=0.9, be         Image: Evaluate		00	Diabetes	2d 22h	21.6 KB			
ılî ^Q	Happiness 21 beta1=0.9, be		٨	Dystopia.Residual	2d 23h	28.8 KB			
ılî	Diabetes dia Z DELETE DEEPNET		.0	Diabetes	3d 21h	21.6 KB			
ılî	Churn in the max training time=1800, missing values	>		Churn	4d 2h	218.7 KB			

Figure 5.111: Stop deepnet creation from pop up menu

A modal window will be displayed asking you for confirmation. If you stop the deepnet during its creation you won't be able to resume the same task. If you want to create the same deepnet, you will have to start a new task.

· · · · · · · · · · · · · · · · · · ·	to longer have access to it, and you will need to recreate it from
your datasets.	

Figure 5.112: Confirmation message to delete a deepnet

# 5.13 Deleting Deepnets

You can delete your deepnet by clicking in the DELETE DEEPNET option from the **1-click action menu** (see Figure 5.113) or using the **pop up menu** on the deepnet list (see Figure 5.114).

¥. 80.	••••	Diabetes	diagnosis   Train	ning (80%)	Ê	$\mathbf{\Phi}$	₲.	(≡)⁵-	G
HIDDEN LAYERS	ALGORITHM	LEARNING RATE	MISSING NUMER		I			ICES	
1	ADAM	0.01	TRUE	SATCH PREE	DICTION			4	
			]	EVALUATE					
				Delete dee				3	PN
				MOVE TO					-

Figure 5.113: Delete deepnet from 1-click action menu

Source	es Datasets Supervised TUnsuperv	ised 🔻	Predict	ions ▼ Tasks				WhizzN	иL 🕶
			×	Deepnets				Ô	Q,
II	Name	\$	Туре	Objective	÷	<b>Å</b> \$	\$	\$	08
ılî ^s	Diabetes diagnosis   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	۲	00	Diabetes	2d 22h	21.6 KB	3	0	0
۱ <mark>۴</mark>	Diabetes dia predict		00	Diabetes	2d 22h	21.6 KB			
ılî	Happiness 22 CREATE BATCH PREDICTION beta1=0.9, be EVALUATE		木	Dystopia.Residual	2d 23h	28.8 KB			
ı <mark>R</mark>	Diabetes dia beta1=0.9, be		00	Diabetes	3d 21h	21.6 KB			
ıl ^Q	Churn in the Churn in the		.0	Churn	4d 2h	218.7 KB			
ıP	max training t Happiness s beta 1=0.9, beta 2=0.999, epsilon=1e-08, learnin	3		Happiness Score	4d 18h	26.0 KB	1		1

Figure 5.114: Delete deepnet from pop up menu

A modal window will be displayed asking you for confirmation. Once a deepnet is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

6	PROJECT: All	L
5	Are you sure you want to delete this deepnet?	
	If you delete this deepnet, you will no longer have access to it, and you will need to recreate it from your datasets.	4
<b>osis  </b> =0.9	Cancel Delete	
osis   2=0.999	9. epsilon=1e-08. learnin	

Figure 5.115: Confirmation message to delete a deepnet

# 5.14 Takeaways

This chapter explains deepnets in detail. Here is a list of key points:

- A deepnet is a supervised Machine Learning algorithm used to solve classification and regression problems.
- A deepnet is built from a dataset available in BigML and used to make an evaluation, a single prediction, or a batch prediction. (See Figure 5.116.)
- You can create a deepnet with just one click or configure it as you wish. BigML provides several **configuration** options before creating your deepnet.
- To create a deepnet you need a dataset containing at least one categorical or numeric field.
- If you do not specify any objective field, BigML will take the last valid field in your dataset.
- BigML allows you to include your numeric fields' missing values as valid values to train your deepnet.
- The PDP view provides a visual way to analyze a field impact on predictions given certain values for the rest of the fields.
- · You get all the objective field class probabilities along with the predicted class.
- You need to evaluate your deepnet performance using data that the model has not seen before.
- · The ultimate goal in building a deepnet is being able to make predictions with it.
- BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the deepnet, so you can easily set the input data and get an immediate response.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the deepnet you want to use to make predictions and a dataset containing the instances for which you want to obtain predictions.
- · You can configure your batch predictions output file settings.
- · You can download your deepnet to perform local predictions.
- You can add descriptive information to your deepnets (name, description, tags, and category).
- You can move your deepnets between projects.
- You can share your deepnets with other people using the secret link.
- · You can stop your deepnet creation by deleting it.
- You can permanently **delete** an existing deepnet.



Figure 5.116: Deepnet Workflow



# Fusions

# 6.1 Introduction

There are many Machine Learning problems that can be solved using supervised Machine Learning techniques. These techniques solve problems that require the prediction of an output variable (objective field) given a number of input variables (input fields). These problems can be classified into two groups: classification problems if you need to predict a category (label or class) or regression problems if the output is a continuous value (a real number).

Classification and regression problems can be solved using multiple Machine Learning methods in BigML, such as models, ensembles, logistic regressions, and deepnets. These methods are explained in Chapter 1, Chapter 2, Chapter 4, and Chapter 5 respectively. Depending on the problem you are trying to solve and the data available, some techniques may perform significantly better than others. **Fusions combine these Machine Learning models and average their predictions** to balance out the individual weaknesses of the single models and yield a **better performance**. Fusions are based on the same "wisdom of the crowds" principle as ensembles under which the combination of multiple models is often more performant than any of its individual models. The component models have to be as accurate and diverse as possible. See Section 6.2 for more details.

This chapter contains a comprehensive description of BigML's fusions including how they can be created (Section 6.3), all configuration options available (Section 6.4), and the different visualizations provided by BigML (Section 6.5). See Section 6.6 for an explanation of how fusions can be used to make predictions. You can also export your fusions in different formats to make local predictions faster at no cost (Subsection 6.7.1). The process to evaluate your fusions' predictive performance in BigML is explained in a different chapter (Chapter 7).

On BigML, the third tab of the main menu of the **Dashboard** allows you to list all your available fusions. The fusion list view (Figure 6.1), details the **Name**, the objective field **Type** (classification or regression), the **Objective** (objective field name), **Age** (time elapsed since it was created), and number of **evaluations**, **predictions**, and **batch predictions** that have been created using that fusion. The SEARCH menu option in the top right corner of the fusion list view allows you to **search** your fusions by name or ID¹ (using the syntax "id:" followed by the fusion ID). You can also search a fusion by the parameters used to create it by typing in the search box the syntax "config:" followed by the parameters you are looking for.

¹https://support.bigml.com/hc/en-us/articles/360000029074-How-do-I-refer-to-my-resources-in-BigML-

Sources Datasets Supervised Unsupervise	d 🔻 Prec	lictions 🕶 Tasks			Whizz	ML 🕶
		Fusions		Ø	Q	∽.
Name	†ј Туре	Objective	t,	<b>\$</b>	<b>(</b>	
Arrythmia diagnosis [filtered] 2 total models (logisticregression: 1, model: 1)	00	Arrhythmia	42min			
Arrythmia diagnosis [filtered] 2 total models (logisticregression: 1, model: 1)	00	Arrhythmia	44min			
Arrythmia diagnosis [filtered] 1 total model (logisticregression: 1)	00	Arrhythmia	44min			
Iris dataset - extended 1 total model (logisticregression: 1)	00	Species	46min			
Arrythmia diagnosis dataset's logistic regression 1 total model (logisticregression: 1)	00	Arrhythmia	48min			
Diabetes iosjd 38 total models (ensemble: 38)	00	Diabetes	1h 44min			
Iris_missing 4 total models (deepnet: 1, ensemble: 1, logisticregression: 1	00	Species	3h 22min			
Titanic 1 total model (logisticregression: 1)	00	Survived	6h 16min			2
Iris_missing 1 total model (logisticregression: 1)	00	Species	3d 4h		1	
Wine Sales 2 total models (model: 2)	٨	Total Sales	3d 4h		2	1
Show 10 C fusions	1 to 1	D of 57 fusions		< < 1	2 3 4 5	> >

Figure 6.1: Fusion list view

By default, when you first create an account at BigML, or every time that you start a new project, your list of fusions will be empty. (See Figure 6.2.)

Sources Datasets	Supervised - Unsupervised - Predictions - Tasks	WhizzML 👻
	C Fusions	S 🛍 🔍 🥵-
Name	↑ Type Objective	
	No fusions	
Show 10 ᅌ fusions	No fusions	к < > )

Figure 6.2: Empty Dashboard fusion view

Finally, in Figure 6.3 you can see the icon used to represent a fusion.



# 6.2 Understanding Fusions

BigML fusions combine multiple Machine Learning models, ensembles, logistic regressions, and deepnets and average their predictions to balance out the individual weaknesses of the single models and yield a better performance. Fusions are based on the assumption that the combination of multiple models often outperforms the component single models. We can see fusions as an heterogeneous ensemble composed by different types of models instead of just decision trees.

For fusions to improve upon the individual model performance, the component models have to be **as accurate and diverse as possible**. If you use several identical models or models with sub-par performance, the fusion will not be able to improve the results of the models. Although for many cases the gains in performance may not be huge, fusions have other advantages such as the fact that they are usually more stable than single models and the model errors tend to be smoothed out across the entire input space. For problems in which every small gain in model performance measures counts, fusions can be a quick solution because they are so easy to execute on BigML.

Fusions can solve classification and regression problems on BigML. For classification problems, fusions average the per-class probabilities across all the component models. The class with the highest probability is predicted. For regression models, the final prediction is the result of averaging the per-model predicted values. These methods to combine single model predictions and return an output is equivalent to the ensembles "probability" method (see Subsection 2.6.3.2).

You need at least one existing supervised model to create a fusion. The component models of a fusion must have a compatible categorical or numeric **objective field** (see Subsection 6.2.1). The models can be built using different datasets and different **input fields**. If two fields from different models have the same name, they will be considered the same field to create the fusion. The per-field importances from the components models will be average to create the **fusion field importances** (see Subsection 6.2.2).

# 6.2.1 Fusion Objective Field

All the models composing the fusion must have the same objective field. BigML checks that two models have the same objective field by ensuring the objectives have the same **field optype** (numeric or categorical) and the same **name** (if you create a fusion from the BigML API² the field ID instead of the name will be used to validate the compatibility of two objective fields). You can select models with different objective field names (see Subsection 6.4.2) if they are compatible.

## 6.2.2 Fusion Field Importances

The fusion field importances are calculated by averaging the per-field importances of the following component models: **decision trees, ensembles and deepnets**. These averages are normalized so the sum of all field importances is 100%. Logistic regressions are excluded from this calculation since the field importances cannot be calculated for them. If the fusion is only composed by logistic regression models, it will not have importances.

# 6.3 Creating Fusions

You need **at least one model** to create a fusion. You can use several options depending if you want to select multiple models of one type or few models of different types to create a fusion:

• If you want to use a single decision tree, ensemble, logistic regression or deepnet, you can click on the CREATE FUSION option from the model **1-click action menu** (see Figure 6.4) or from the **pop up menu** on the list view (see Figure 6.5):

²https://bigml.com/api/fusions



Figure 6.4: Create fusion from 1-click menu

ŝ			Models				Ø	Ô	Q,
ill	Name	t¦ Type t	Objective	1	1	₿ ţ	<b>\$</b>	<b>1</b>	00
ılî	Iris_missing 512-node, pruned, deterministic order	00	Species		1d 2h	4.4 KB			
ılı	rossmann_20%sample_500_stores 512-node, pruned, deterministic order	00	Store		2d 3h	645.2 KB			
ılî	rossmann_20%sample_100_stores 512-node, pruned, deterministic order	00	Store		2d 3h	126.1 KB			
ılî	Movie_awards_2016 512-node, pruned, deterministic order	00	awards		2d 3h	128.4 KB			
ılî	Titanic Survival_two_fields auto-optimized, model candidates=8, max. tr	00	Survived		2d 3h	23.8 KB			
ılî	Titanic Survival_two_fields 512-node, pruned, deterministic order	00	Survived		2d 3h	23.8 KB			
ılî	Stumble_upon.csv 512-node, pruned, deterministic order	00	label		4d 4h	9.8 MB			
ılî	Stumble PREDICT BY QUESTION	.0	label		4d 5h	9.8 MB			
ılî	Titanic S 512-node CREATE BATCH PREDICTION		Survived		1w	16.4 KB			
ılî	Titanic S EVALUATE	.0	Survived		1w 4d	73.1 KB			
Show 10	CREATE FUSION		1 to 10 of 164 models			Κ	< 1 2	3 4 5 3	×
	DELETE MODEL								

Figure 6.5: Create fusion from pop up menu

• If you want to select multiple decision trees, ensembles, logistic regressions or deepnets at the same time, you can click on the CREATE FUSION option from the list view (see Figure 6.6).

Sourc	es Datasets Supervised → Unsu	pervised 👻	Predictions -	Tasks		6	Close fusio	Whiz	ZZML ew
ា			Models					x*   1	
	Create fusion     0 models selected				Search	by name			× Ø
ជា	Name	t̂↓ Type t	Objective	tj.	<b>#</b> 1	t,	<b>*</b>	<b>1</b>	
	Iris_missing 512-node, pruned, deterministic order	00	Species		1d 2h	4.4 KB			
	rossmann_20%sample_500_stores 512-node, pruned, deterministic order	00	Store		2d 3h	645.2 KB			
	rossmann_20%sample_100_stores 512-node, pruned, deterministic order	00	Store		2d 3h	126.1 KB			
	Movie_awards_2016 512-node, pruned, deterministic order	00	awards		2d 3h	128.4 KB			
	Titanic Survival_two_fields auto-optimized, model candidates=8, max. tr	00	Survived		2d 3h	23.8 KB			
	Titanic Survival_two_fields 512-node, pruned, deterministic order	00	Survived		2d 3h	23.8 KB			
	Stumble_upon.csv 512-node, pruned, deterministic order	00	label		4d 4h	9.8 MB			
	Stumble_upon.csv 512-node, pruned, deterministic order	00	label		4d 5h	9.8 MB			
	Titanic Survival 512-node, pruned, deterministic order	00	Survived		1w	16.4 KB			
	Titanic Survival_disorder_fields 512-node, pruned, deterministic order	00	Survived		1w 4d	73.1 KB			
Show 1	10 📀 models		1 to 10 of 164 models				IK < 1	2 3 4 5	5 > >

Figure 6.6: Create fusion from model list views

When you select one model from the list, you will only be able to select other compatible models, i.e., those models with the same objective field type and name. The rest of models will not be selectable (see Figure 6.7).

	Source	es Datasets Supervise	ed - Unsupervise	ed 🕶	Predictions -	Tasks				Whiza	ML 🗸
	នា				Models				8	C 🕯	Q,
	•	Create fusion 1 mo	dels selected				Search	by name			x Ø
	ŵ	Name	t _i	Type $\dagger_{\downarrow}$	Objective	t,	<b>***</b> 1,		<b>\$</b>	<b>F</b>	
		Iris_missing 512-node, pruned, deterministic o	rder	00	Species		1d2h	4.4 KB			
		rossmann_20%sample_500_sto 512-node, pruned, deterministic o		00	Store		2d 3h	645.2 KB			
You cannot s different	select a m	non pruned deterministic o		00	Store		2d 3h	126.1 KB			
	Ď	Movie_awards_2016 512-node, pruned, deterministic o	rder 🖉 📀	00	awards		2d 3h	128.4 KB			
		Titanic Survival_two_fields auto-optimized, model candidates	=8, max. tr	00	Survived		2d 3h	23.8 KB			
	€	Titanic Survival_two_fields 512-node, pruned, deterministic of	rder	o0	Survived		2d 3h	23.8 KB			
		Stumble_upon.csv 512-node, pruned, deterministic o	rder	o0	label		4d 4h	9.8 MB			
		Stumble_upon.csv 512-node, pruned, deterministic o	rder	o0	label		4d 5h	9.8 MB			
		Titanic Survival 512-node, pruned, deterministic o	rder	00	Survived		1w	16.4 KB			
		Titanic Survival_disorder_fields 512-node, pruned, deterministic of	rder	00	Survived		1w 4d	73.1 KB			
	Show 1	0 🔉 models		1	to 10 of 164 models				< < 1	2 3 4 5	> >I

Figure 6.7: Select models with the same objective field

• You can also select multiple models from the OptiML view and click on the Create fusion button (see Figure 6.8).

All models 🔹	Max. phi coefficient 🔹 🖣	ROC AUC	▼ lį	
L1 regularized (c=1.71198295057), auto-scaled,	0.39377	0.69740		$\sim$
1 regularized (c=2.79845577478), auto-scaled,	0.39377	0.69717		$\sim$
L1 regularized (c=5.14958842099), auto-scaled,	0.39377	0.70056		$\sim$
12 regularized (c=0.679651863873), auto-scaled,	0.39364	0.69736		$\sim$
11 regularized (c=2.92560433596), auto-scaled,	0.39240	0.69613		$\sim$
L1 regularized (c=0.732339121401), auto-scaled,	0.39082	0.69610		$\sim$
auto network search, 128 networks evaluated, mi	0.39071	0.73452		~
L2 regularized (c=2.37360942364), bias, auto-sca	0.39000	0.70988		$\sim$
L2 regularized (c=0.966485567391), bias, auto-sc	0.38973	0.69660		$\sim$
L1 regularized (c=1.2953491807), bias, auto-scal	0.38867	0.69465		$\sim$
ihow 10 🗘 models	1 to 10 of 73 models		K < 1 2 3 4 5	> >I

Figure 6.8: Select models from the OptiML table

• If you want to create a new fusion using the same models of another existing fusion, you can click on the CREATE FUSION USING THIS ONE option from the fusion **1-click action menu**. This option is very useful to iterate your existing fusions.

Sources Datasets Supervised	Unsupervised -	Predictions -	Tasks	WhizzML 👻
		Titanic Surviv	al 📋 🗘	ૐ- (≡)⁵- ③
TOTAL MODELS	OBJECTIVE FIELD		BATCH PREDICTION	
Y Joined ABC •		£3	CREATE FUSION USING THIS ONE	
			- min.	Probability max.
Southampton			FALSE TRUE	
Queenstown			All classes	*

Figure 6.9: Create fusion using another fusion

All of the above options will redirect you to the fusion creation view (see Figure 6.10). From this view, you can select more models and configure the fusion parameters. See a detailed explanation of these options in Section 6.4.

Sources Datasets Supervised Unsup	pervised - Predictions - Tasks		WhizzML 👻
a 17	New fusion		<b>5</b> - i
Select different models with the same objective field to make or	ombined predictions and evaluations		
Show OptiML models	Add models	Survived	
Models selected 👔			
Z Titanic Survival L2 regularized (c=1), bias, auto-scaled, missing values	s Survived	Weight:	1 ×
Fusion name:		Reset 🔀 Create	e Fusion

Figure 6.10: Fusion creation view

# 6.4 Fusion Configuration Options

From the fusion creation view, you can select more models or remove the existing ones (Subsection 6.4.1), remove the objective field filter (Subsection 6.4.2) and configure the model weights (Subsection 6.4.3).

# 6.4.1 Models

Select up to 1,000 models, ensembles, logistic regressions, and/or deepnets to create your fusion. You can also use existing fusions to include them in a new fusion.

Click on the model icon you want and type the model name in the selector (see Figure 6.11). You can also select models created using OptiML by clicking the option "Show OptiML models". By enabling this option, only OptiML models will be listed in the selector.

Sources Datasets Supervised	Unsupervised - Predictions - Tasks	WhizzML 👻
	New fusion	<b>5</b> - (i)
Select different models with the same objective field to	create a fusion	
	Ciabet	Φ <b>,</b>
Show OptiML models	Diabetes 512-node, pruned, deterministic order	
Models selected: 0 👔	Diabetes diagnosiscxc 512-node, pruned, deterministic order	
	Diabetes diagnosis dataset   Training (70%) v1 512-node, pruned, deterministic order	
	Diabetes diagnosis dataset   Training (70%) v1 512-node, pruned, deterministic order	
	Diabetes diagnosis dataset   Training (70%)	
Fusion name:		
Name based on the seleted resources		Create fusion

Figure 6.11: Search for models

The selected models will be shown below (see Figure 6.12) and you can remove them or assign them different weights (see Subsection 6.4.3). All the selected models must have compatible objective fields;

otherwise, the fusion creation will fail. When you select the first model, the rest of models will be filtered by the same objective type (numeric or categorical) and the same objective name. You can remove the filter in case you want to select models with different objective field names (see Subsection 6.4.2).

Sources Datasets Super	vised - Unsupervised - Predict	ions 🕶 Tasks	٧	WhizzML 🔻
A 17	🚫 Nev	v fusion		<b>%</b> - i
elect different models with the same ob	ective field to create a fusion			
Show OptiML models	Add models		Diabetes	כ
Models selected: 4 🝞				
1 Diabetis diagnosis - sampl 512-node, pruned, determin	e (80.00%) istic order	Diabetes	Weight: 1	×
2 Z Diabetis diagnosis - sample L2 regularized (c=10000.0),	e (80.00%) oias, auto-scaled, missing values, stats	Diabetes	Weight: 1	×
3 Piabetes diagnosis   Traini bootstrap decision forest, 5	ng (80%) 2-node, 31-model, deterministic order	Diabetes	Weight: 1	×
4 Solution 2 4 Sol	ng (80%) hidden layers, adam, learning rate=0.002, 17-it	te Diabetes	Weight: 1	×
्र्कू Configure				$\overline{\mathbf{O}}$
Fusion name:				
Diabetes diagnosis   Training (	30%)		Reset 🛛 🖉 Create	fusion

Figure 6.12: Selected models

## 6.4.2 Objective Field

The objective field, or "target field", is the field you want to predict. Fusions support **categorical** and **numeric** fields as the objective field. All the models selected to create a fusion must have compatible obejctive fields.

BigML takes the first model **objective field type and name to filter** the rest of models available in the selector (see Figure 6.13).

Sources Datasets Supervised - Unsuperv	vised - Predictions - Tasks	WhizzML 🔻
A 17	New fusion	ن الح
Select different models with the same objective field to create a fus	sion	
Show OptiML models		Diabetes
Models selected: 4 👔		
1 A Diabetis diagnosis - sample (80.00%) 512-node, pruned, deterministic order	Diabetes	Weight: 1
2 Z Diabetis diagnosis - sample (80.00%) L2 regularized (c=10000.0), bias, auto-scaled, missing	g values, stats Diabetes	Weight: 3
3 An Diabetes diagnosis   Training (80%) bootstrap decision forest, 512-node, 31-model, deter	ministic order Diabetes	Weight: 2
4 Signature Structure Suggestion, 3 hidden layers, adam, lear	ming rate=0.002, 17-ite	Weight: 5
Configure		$\overline{\mathbf{O}}$
Fusion name:		
Diabetes diagnosis   Training (80%)		Reset Create fusion

Figure 6.13: The model search will be filtered by the objective field name

You can remove this filter and select models with different objective field names (see Figure 6.14.

Sources Datasets Supervise	d - Unsupervised - Predictio	ns 🕶 Tasks		WhizzML 👻
A 17	New 1	fusion		<b>%</b> - (i)
Select different models with the same objecti	ve field to create a fusion			
Show OptiML models	Add models		Diabetes Click to filter the models by the selected objective	
Models selected: 4 🕜		C		<b>)</b>
1 Siabetis diagnosis - sample (8) 512-node, pruned, deterministic		Diabetes	Weight:	1 ×
2 Z Diabetis diagnosis - sample (8) L2 regularized (c=10000.0), bias,	0.00%) auto-scaled, missing values, stats	Diabetes	Weight:	3 ×
3 A Diabetes diagnosis   Training ( bootstrap decision forest, 512-n	30%) ode, 31-model, deterministic order	Diabetes	Weight:	2 ×
4 K Diabetes diagnosis   Training ( auto structure suggestion, 3 hido	<mark>30%)</mark> Ien layers, adam, learning rate=0.002, 17-ite.	Diabetes	Weight:	5 ×
Configure				$\overline{\mathbf{O}}$
Fusion name:				
Diabetes diagnosis   Training (80%	6)		Reset Creat	e fusion

Figure 6.14: Remove the objective field filter to select models with different objective field names

When you remove the filter and select a new model with a different objective field name, you can choose which objective field from the selected models you want to use for the filter, if any.

Sources Datasets Supervised - Predictions - Tasks	WhizzML 👻
New fusion	<b>5</b> - (i)
Select different models with the same objective field to create a fusion	
Add models  Add models  Models selected: 5 0	Diabetes diagnosis   Diabetes Diabetes diagnosis
1     1     1     Diabetis diagnosis - sample (80.00%) 512-node, pruned, deterministic order	Weight: 1
2 Z Diabetis diagnosis - sample (80.00%) L2 regularized (c=10000.0), bias, auto-scaled, missing values, stats Diabetes	Weight: 3
3 A Diabetes diagnosis   Training (80%) bootstrap decision forest, 512-node, 31-model, deterministic order	Weight: 2
4 X       Diabetes diagnosis   Training (80%) auto structure suggestion, 3 hidden layers, adam, learning rate=0.002, 17-ite       Diabetes	Weight: 5
5 A Diabetes 512-node, pruned, deterministic order	Weight: 1
්සු Configure	•
Fusion name:	~*
Diabetis diagnosis - sample (80.00%)	Reset Create fusion

Figure 6.15: Select which objective field you want to use to select more models

# 6.4.3 Weights

You can assign different weights to the selected models before creating the fusion. At the prediction time, BigML will perform a **weighted average** of all model predictions taking into account each model weight. Therefore, if a model has a weight of 2 (while the rest of models have a weight of 1) this model's predictions will count double. You can assign weights of 0 if you do not want a model to have any impact on the final predictions.

Sources Datasets Supervised	Unsupervised - Prediction	is ▼ Tasks		WhizzML 👻
a 🔽	New fr	usion		<b>%</b> - (i)
Select different models with the same objective	field to create a fusion			
Show OptiML models			Diabetes	
Models selected: 4 👔				
1 Diabetis diagnosis - sample (80.) 512-node, pruned, deterministic o		Diabetes	Weight:	1 ×
2 Z Diabetis diagnosis - sample (80. L2 regularized (c=10000.0), bias, a		Diabetes	Weight:	3 ×
3 A Diabetes diagnosis   Training (80 bootstrap decision forest, 512-noc	%) e, 31-model, deterministic order	Diabetes	Weight:	2 ×
4 Solution 2 A Biabetes diagnosis   Training (80 auto structure suggestion, 3 hidde	%) Iayers, adam, learning rate=0.002, 17-ite…	Diabetes	Weight:	5 ×
್ಕ್ಷಿ Configure				$\overline{\mathbf{O}}$
Fusion name:				
Diabetes diagnosis   Training (80%)			Reset 🔀 Creat	e fusion

Figure 6.16: Assign weights to the models

## 6.4.4 Creating Fusions with Configured Options

After finishing the configuration of your options, you can change the default fusion name in the editable text box. Then you can click on the Create fusion button to create the new fusion, or reset the configuration by clicking on the Reset button.

A 17	New fu	sion	<b>5</b> - (i)
Show OptiML deepnets	Add deepnets diabetes		$\Phi_{\mathbf{x}}$
Models selected: 4			
1 A diabetes training set 512-node, pruned, deterministic	order	diabetes	Weight: 2
2 A diabetes training set bootstrap decision forest, 512-n	ode, 10-model, pruned, deterministic order	diabetes	Weight: 1
3 Z diabetes training set L2 regularized (c=1), bias, auto-s	caled, missing values	diabetes	Weight: 1
4 Kiabetes training set auto structure suggestion, 3 hide	len layers, adam, learning rate=0.002, 33	diabetes	Weight: 1
Fusion name: diabetes training set			Reset Create fusion
diabetes training set			
Fusion Name		API Preview	Reset Create Fusion

Figure 6.17: Create fusion after configuration

# 6.4.5 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to the Reset button (See (Figure 6.17)). This is to show how to create the fusion programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 6.18) below:

S3 68		🔏 🕺 Add deepnets	$\Phi^{\mathbf{y}}$	
Models selecter	API reques		×	
2 Pan diabe	1 - { 2 - 3 - 4	<pre>"models": [     {         id": "model/5e8f5bb528e1f40aa1001201",</pre>		
3 diabe	4 5 7 - 8 9	<pre>}, {     "id": "ensemble/5e8f5bf028e1f40aab000747",</pre>		
4 diabe auto s	9 10 11 -	"weight": 1 }, { Check the full list of arguments in the /		

Figure 6.18: Fusion API request preview

There are options on the upper right to either export the JSON or copy it to clipboard. On the bottom there is a link to the API documentation for fusions, in case you need to check any of the possible values or want to extend your knowledge in the use of the API to automate your workflows.

Please note: when a default value for an argument is used in the configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments are missing, there is no need to send them in the creation request.

# 6.5 Visualizing Fusions

After creating your fusion, you will be able to analyze the results with BigML's unique visualization: a **Partial Dependence Plot (PDP)** to examine how the input fields impact the objective field (see Subsection 6.5.1). You will also be able to find the models composing the fusion under the **model list** tab (see Subsection 6.5.2).

Switch from the PDP view to the model list view by clicking on the icons in the top bar menu of the fusion view (see Figure 6.19).

Sources Datasets Supervis	ed - Unsupervised - Predictions -	Tasks	WhizzML •
₽ <b>₽</b> ₽↓ •••••	Diabetes diagnosis   Tr	aining (70%)'s	〕 ♠ ∽ (≡)⁵~ (
Model list	DIABETES	9	FALSE
			S
Y au	35.6469		esc Releases v
Y BMI 123	<ul> <li>35.6469 </li> <li>✓ </li> </ul>	PF	REDICTION
		True	Probability
60 -		50.1%	92.6%
		False	38.63%
50		True	61.38%
40 -			
		All classes	¥
30			
		INF	PUT FIELDS All fields:
20 -		Pregnancies	3.44%
10 -		0	21 3
0.0		Blood pressure	● 6.61% <b>□</b> 152
0, 10 10 60 6	87, 89, 94, 95, 95, 8		69
Glucose	123 152.8809	Skinfold	5.78%
∧ Giucose	102.0009	0	123

Figure 6.19: Switch from PDP to model list views

# 6.5.1 Fusion Partial Dependence Plot

The chart view is composed of three main parts: the CHART itself, the PREDICTION legend and the INPUT FIELDS form. (See Figure 6.20.) You can find a detailed explanation of each one below.



Figure 6.20: Fusions chart view parts

• The CHART allows you to view the impact of the input fields on the objective field predictions. You can select a different input field for each axis and the predictions are represented in different colors on the heat map. You can select numeric or categorical fields for the axis and inspect the axis values in the grey area next to the selector. You can switch the axis by clicking on the option on top of the chart area (see Figure 6.21).

ources Datasets Super	vised - Unsupervised - Predictions		WhizzML 🔻
60	Diabetes diagnosis	Training (70%)'s	⊕
TOTAL MODELS 6	OBJECTIVE FIELD DIABETES	FIELDS 9	FIELDS MAP False
			C Pro
<b>Y</b> BMI 123	35.6469		esc Releases vie
		True	Probability
60 -		50.1%	92.6%
50 -		False	38.63%
		True	61.38%
40 -			
30 -		All classes	¥
		1F	INPUT FIELDS All fields:
20 -		Pregnancies	3.44%
10 -		0	21
0.0 -		Blood pressure	●                     6.61% □ 152
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	8°, 8°, 9°, 9°, 8°, 8°, 8°	6	69
X Glucose	<b>123 •</b> 152.8809		5.78%
		0	123

Figure 6.21: Select fields for your fusion PDP axis

You can freeze the view by pressing Shift and release it again by pressing Escape from your keyboard. When the view is frozen, an edition icon will appear so you can edit the axis values and obtain a prediction for another desired value. You can see the predictions in the legend to the right.

• The PREDICTION legend allows you to visualize the **objective field classes** (classification fusions) or the predicted **value** (regression fusions). For classification, each class is represented by a color, the main probability color bar at the top is the probability for the predicted class. By default, colors are **shaded** according to the prediction range shown in the chart area. That way, smaller differences in predictions are easier to perceive. However, you can choose to see the color shading according to the total range of possible values for the objective field by clicking on the icon next to the prediction bar Total (see Figure 6.22). You can also select to see only one of the classes using the class selector at the bottom of the legend.



Figure 6.22: Prediction legend

Freeze this view by pressing Shift, and release it again by pressing Escape from your keyboard.

• Below the chart legend, you can find the INPUT FIELDS form (see Figure 6.23). You can configure the values for any numeric, categorical, text or items field. By changing their values, you can see the predictions changing in real-time. You can also select or disable your input fields, so they will be treated as missing values. If your fusion contains models, ensembles or deepnets, you will be able to see each field importance and order the fields from the highest to lowest importance. If your fusion only contains logistic regressions, the fields will not have importances.

urces Datasets Super	vised  Unsupervised  Predictions	· Tasks	WhizzML 🔻
60	Diabetes diagnosis   1	raining (70%)'s	<b>□ □ □ □ □ □ □ □ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓</b>
TOTAL MODELS	OBJECTIVE FIELD	FIELDS	FIELDS MAP
6	DIABETES	9	FALSE
			S
BMI 123	35.6469 🖌 🕻		esc Releases vi
Divil	00.0403 P Cx		PREDICTION
		True	Probability
60 -		50.1%	92.6%
50		False	38.63%
		True	61.38%
40 -			
		All classes	¥
30			
20 -		Pregnancies	INPUT FIELDS All fields:
		0	
10		-0	3
		Blood pressure	6.61%
0.0 8 & 8 & 8	\$ \$ \$ \$ \$ \$ \$ \$	0	152 69
· · ·	- · · · · · · · ·		03
X Glucose	152.8809	Skinfold	5.78%

Figure 6.23: Configure the values for other input fields

#### Note: if any of the models composing the fusion fail to calculate the prediction (e.g., if a logistic regression is trained without missing numerics it will fail for predictions that have missing numeric values). BigML will use the other models to return a result instead of returning an error.

Moreover, the chart includes a reset option for your input fields values, and an export option to download your chart in PNG format explained below:

• After selecting the fields for the axis or configuring the input fields values, you can set them again to the default view by clicking the **reset** icon highlighted in Figure 6.24.

Sources Datasets Supervis	ed - Unsupervised - Predictions	s ▼ Tasks	WhizzML 🗸
ີ ເ <mark>ພີ້ ••••</mark>	Diabetes diagnosis	Training (70%)'s	<b>〕</b>
6	DIABETES	9	TRU: Reset values
			Reset values
Y BMI 123	• 44.0344 🖋 🕃		esc Releases vi
			PREDICTION
		True	Probability
60		50.3%	93.8%
		False	22.88%
50 -			
		True	77.12%

Figure 6.24: Reset the values for the input fields

• You can also **export** your chart in PNG format with or without legends. Freeze the view by pressing Shift from your keyboard and export the chart to get the prediction values in the legend. Release the view by pressing Escape.

ources Datasets Supervi	sed	Tasks	WhizzML 🔻
រ ព្រឹ ^ង ••••	Diabetes diagnosis   Ti	raining (70%)'s	🚉 💠 🦽₋ (≡)⁵₋ ઉ
TOTAL MODELS	OBJECTIVE FIELD	FIELDS	FIELDS MAP
6	DIABETES	9	TRUE
			EXPORT CHART AS IMAGE
Y BMI 123	▼ 44.0344 € Č¥		EXPORT CHART AS IMAGE WITH LEGENDS
		True	Probability
60 -		50.3%	93.8%
50		False	22.88%
		True	77.12%
40			

Figure 6.25: Export chart as image with or without legends

Note: there are some limitations for the number of classes of the objective field and the number of input fields to visualize your fusion in the chart (explained in Section 6.8).

## 6.5.2 Fusion Model List

The model list contains all the models composing the fusion. You can see the model number (this is the order in which you selected the models to create the fusion), the model type icon, the model name, the model configuration options, and the model weights (see Subsection 6.4.3 for an explanation about the weights). If you click on the model name, you will be taken to that model view. You can filter this list by model type by clicking on the icons shown in Figure 6.26.

Sources	Datase	ets Supervised - Unsupervised - Predictions - Tasks White	zML 👻
<b>A B</b>	P	• Diabetis diagnosis - sample (80.00%)	<b>(</b>
	TOTAL MO	ODELS     OBJECTIVE FIELD     FIELDS     FIELDS MAP       DIABETES     9     FALSE	
			2   0
Number $\boldsymbol{\dagger}_{\downarrow}$	Model ${\uparrow}_{\downarrow}$	Name	Weight 🖞
1	សា	Diabetis diagnosis - sample (80.00%) 512-node, pruned, deterministic order	1
2	Σ	Diabetis diagnosis - sample (80.00%) L2 regularized (c=10000.0), bias, auto-scaled, missing values, stats	3
3	ŕa	Diabetes diagnosis   Training (80%) bootstrap decision forest, 512-node, 31-model, deterministic order	2
4	×	Diabetes diagnosis   Training (80%) auto structure suggestion, 3 hidden layers, adam, learning rate=0.002, 17-iteration, beta1=0.9, beta2=0.999, epsilon=1e-08, ml	5
Show 10	c models	1 to 4 of 4 models   K   K	

Figure 6.26: Fusion model list view

# 6.6 **Fusion Predictions**

### 6.6.1 Introduction

The ultimate goal in building a fusion is being able to make predictions with it. On BigML, you can make predictions for **single instances** or for **many instances in batch**. Each prediction comes with a measure indicating the prediction confidence. For regression problems, the **expected error** is provided along with the predicted value, for classification problems the vector of **probabilities** per class is returned (a percentage ranging from 0% to 100%).

The predictions tab in the main menu of the BigML Dashboard is where all your saved predictions are listed. (see Figure 6.27). You can **search** your predictions by name clicking on the search option on the top menu. In the predictions list view, you can see the **fusion** icon used for each prediction, the **Name** of the prediction, the **Objective** (objective field name), the **Prediction** (the prediction result), and the **Age** (time since the prediction was created).

*	Ē	Predictions		Q
ជា	Name	Objective	1 Prediction	t, 🛗
83	Wine Sales operating kind=probability, 5 inputs	Total Sales	78.15923	23h 39mir
C2*	Titanic Survival v1 operating kind=probability, 3 inputs	Survived	FALSE	1w
C2*	Titanic Survival operating kind=probability, 3 inputs	Survived	FALSE	1v
C2*	Iris dataset - extended operating kind=probability, 4 inputs	Species	Iris-versicolor	1w 1c
C2*	Titanic Survival_two_fields operating kind=probability, 1 inputs	Survived	FALSE	1w 1
C2	Stumble_upon.csv operating kind=probability, 24 inputs	label	0	1w 30
C2*	Stumble_upon.csv operating kind=probability, 24 inputs	label	0	1w 3
C2*	Stumble_upon.csv operating kind=probability, 24 inputs	label	0	1 w 30
C2*	Stumble_upon.csv operating kind=probability, 24 inputs	label	0	1 w 30
83	Titanic Survival operating kind=probability, 2 inputs	Survived	FALSE	1w 6

Figure 6.27: Predictions list view

When you first create an account on BigML, or every time that you start a new project, your list of predictions will be empty (see Figure 6.28).

Source	s Datasets	Supervised - Unsup	ervised - Pred	lictions - Tasks				WhizzMl	
ŧ			2	Predictions				Ô	Q,
ŝ	Name		t _i	Objective	ţ.	Prediction	ţ.		1
			No	predictions					
Show 10	predictions		No	predictions				< >	>1

Figure 6.28: Empty predictions list view

Fusion predictions are saved under the CLASSIFICATION & REGRESSION option in the menu (see Figure 6.29).

		Unsupervised      Predictions      Ta					
=		CLASSIFICATION & REGRESSION				ŵ	Q,
-	42	CLUSTER ANALYSIS				ш	~
ŝ	Name	ANOMALY DETECTION	1	Prediction	1		
		ASSOCIATION DISCOVERY					
		TOPIC MODELING					
Show 1	0 c predictions	No predictions			1	< < >	5

Figure 6.29: Menu options of the predictions list view

Select the list for your single instances predictions or your batch predictions by clicking on the corresponding icons. (See Figure 6.30 and Figure 6.31.)



Figure 6.30: Single predictions icon



Figure 6.31: Batch predictions icon

### 6.6.2 Creating Fusion Predictions

BigML provides two options to predict with your fusions explained in the following subsections:

- PREDICT: to predict a single instance
- BATCH PREDICTION: to predict multiple instances in batch.

#### 6.6.2.1 Predict

BigML allows you to quickly make predictions for single instances by providing a form containing the input fields used by the fusion, so you can easily set the values and get an immediate response.

Follow these steps to create a single prediction:

1. Click PREDICT in the fusion 1-click action menu. (See Figure 6.32.)



Figure 6.32: Predict using the 1-click action menu

Alternatively, click PREDICT in the pop up menu in the list view. (See Figure 6.33.)
	🚫 Fusi	ions		⊠ <b>*</b>	Î Q	4
Name	tj Type	Objective	<b>#</b> 1	<b>\$</b>		
Diabetes diagnosis   Training (70%)'s 6 total models (deepnet: 1, ensemble: 1, logisticregression: 2	8 al	Diabetes	22min			
Titanic Survival PREDICT		Survived	1h 16min			
Titanic Survival       CREATE BATCH PREDICTION         3 total models (logisticregression:       EVALUATE		Survived	1h 28min			
Titanic Survival 35 total models (deepnet: 1, enser	.0	Survived	1d 1h			
Diabetes diagnosis VIEW DETAILS 66 total models (ensemble: 38, log X* DELETE FUSION		Diabetes	1d 22h			
Arrythmia diagnosis [filtered] 2 total models (logisticregression:	>	Arrhythmia	2d 23h			
Arrythmia diagnosis [filtered] 2 total models (logisticregression: 1, model; 1)	00	Arrhythmia	2d 23h		1	
Arrythmia diagnosis [filtered] 1 total model (logisticregression: 1)	00	Arrhythmia	2d 23h			
Iris dataset - extended 1 total model (logisticregression: 1)	00	Species	2d 23h			1
Arrythmia diagnosis dataset's logistic regression 1 total model (logisticregression: 1)	00	Arrhythmia	2d 23h			

Figure 6.33: Predict using the pop up menu

2. You will be redirected to the **prediction form** where you will find all the fields used by the fusion as input fields. (See Figure 6.34.)

÷		dict using Diabe	0			(≡)⁵- (
		Diabetes: ?		?		
Select a positive class	Probability threshold:		50%		All inp	out fields: 💽
regnancies		۲	Glucose			C
0C	21	10	0		244	122
lood pressure		۲	Skinfold			C
0(	152	76	0	6	123	61
sulin		<b></b>	вмі			C
0(	1057	528	0.0	6	83.88	40.74
iabetes pedigree		<b>S</b>	Age			C
0.0	3.0	1.46	6	0	95	51
ew prediction name						

Figure 6.34: Fusion prediction form

3. Select the fields to be used for your prediction, set the input values for your selected fields and

click Predict. Non-selected fields will be considered as missing values during the prediction. See the prediction at the top of the view along with the rest of the class probabilities (Figure 6.35). Hide or display the histogram view containing the rest of your **class probabilities**. You can download all the probabilities in PNG, CSV or JSON format by clicking on the corresponding icons. (See Subsection 6.6.4.1.)

Sources Datasets Supervised - U	Insuperv	ised - Predicti	ons - Tasi			WhizzML 👻
9 ¥.		iabetes diagnos	sis   Training	(70%)'s	4	)- (≡)5.
		Diabetes: True		86.8	1%	
			8	5.81%		
	13.19%					
4	alse		~	.u ^e		
Select a positive class Probability three	shold:		50%		AU :	nput fields: 🕑
Pregnancies		<b>S</b>	Glucose		AII I	input neius. 💽
0	21		0		244	
Ö		10		0		122
Blood pressure		2	Skinfold			۲
0	152	76	0	6	123	61
Insulin		<b></b>	вмі			<b>e</b>
0	1057	528	0.0		83.88	40.74
0				0		40174
Diabetes pedigree	3.0	≤	Age 6		95	۲
<u> </u>	3.0	1.46	-			51
New prediction name						
Diabetes diagnosis   Training (70%)'s					e.	

Figure 6.35: Fusion prediction

BigML predictions are synchronous, i.e., when you send the input data, you get an immediate response.

4. Your prediction is automatically saved and you can find it in the predictions list view.

For **regression** fusions, the process is the same, but instead of the predicted classes and the probabilities you get a numeric value for the objective field along with an expected error as the certainty measure.

Note: this option is only available from the BigML Dashboard for fusions with less than 100 fields. If you want to perform single instance predictions for a higher number of fields, use the BigML API³.

³https://bigml.com/api/predictions

### 6.6.2.2 Batch Predictions

BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you need is the fusion you want to use to make predictions and a dataset containing the instances you want to predict. BigML will create a prediction for each instance in the dataset.

Follow these steps to create a batch prediction:

1. Click on BATCH PREDICTION option under the fusion 1-click action menu (Figure 6.36)



Figure 6.36: Create batch prediction using 1-click action menu

Sources Datasets Supervis	ed - Unsupervised - Pr	edictions	s 🕶 Tasks			Whizz	ML 🕶
		3 Fusi	ons		Ø	Î <	<b>5</b> .
Name	ţ	Туре	Objective	<b>#</b> 1,	\$	🗐 (j	
Diabetes diagnosis   Training (70%)'s 6 total models (deepnet: 1, ensemble: 1, logi	sticregression: 2 🖉 🧕	o0	Diabetes	22min			
237 total models (deennet: 30, ens	REDICT	00	Survived	1h 16min			
Titanic Survival	REATE BATCH PREDICTION	00	Survived	1h 28min			
	REATE FUSION USING THIS ONE	.0	Survived	1d 1h			
Diabetes diagnosis	IEW DETAILS		Diabetes	1d 22h			
Aventhmia diagnagia [filtered] -2	10VE TO	>	Arrhythmia	2d 23h			
Arrythmia diagnosis [filtered] 2 total models (logisticregression: 1, model:	1)	.0	Arrhythmia	2d 23h		1	
Arrythmia diagnosis [filtered] 1 total model (logisticregression: 1)		00	Arrhythmia	2d 23h			
Iris dataset - extended 1 total model (logisticregression: 1)		00	Species	2d 23h			1
Arrythmia diagnosis dataset's logistic reg 1 total model (logisticregression: 1)	ression	00	Arrhythmia	2d 23h			
Show 10 🗘 fusions	1 ta	o 10 of 62 fu	sions		K < 1	2 3 4 5	У

Alternatively, click on CREATE BATCH PREDICTION in the **pop up menu** of the list view (Figure 6.37).

Figure 6.37: Create batch prediction using pop up menu

2. Select the dataset containing all the instances you want to predict. The instances should contain the input values for the fields used by the fusion as input fields. From this view you can also select another fusion from the selector (see Figure 6.38).

Sources Datasets Supervised - Unsupervised - Predic	tions - Tasks	WhizzML 🔻
New Ba	tch Prediction	<b>5</b> - i
A A ∠ ≤ ∞		
Diabetes diagnosis   Training (70%)'s x v Diabetes diagnosis   Trai	Search dataset 30% Diabetis diagnosis dataset   Test (30% No sociel Microsociel	م م ()
Description:		
Prediction name: dataset with Diabetes diagnosTraining (70%)'s	Reset	

Figure 6.38: Select dataset for batch prediction

3. After the fusion and the dataset are selected, the batch prediction **configuration options** will appear along with a **preview** of the prediction output (a CSV file) (see Figure 6.39). The default output format includes all your prediction dataset fields and adds an extra column with the class predicted. See Subsection 6.6.3 for a detailed explanation of all configuration options.

		ons - Tasks	WhizzML ·
1 T	New Batch	h Prediction	<b>3</b> -
ରେ 🖍 🗶			
Diabetes diagnosis   Train	ning (70%)'s × *	Diabetis diagnosis dat	aset   Test (30%) × v
Diabetes	Thu, 28 Jun 2018 19:55:38	1 ⁰	Wed, 08 Jun 2016 15:39:3
	6 Jodels	8.1 KB size	10 231 fields instances
Description:	$\overline{\mathbf{O}}$	Description:	$\overline{\mathbf{O}}$
Preview of the prediction file (using Pregnancies, Glucose, Blood p 123, 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123, 123,	pressure,Skinfold,Insulin,BMI,Diab6 3,123,ABC,ABC,ABC 3,123,ABC,ABC,ABC,ABC	etes pedigree,Age,Diabetes,we	ight,Diabetes
123,123,123,123,123,123,123,123,123,123,	3,123,ABC,ABC,ABC		

Figure 6.39: Configuration options for fusion batch prediction

4. By default, BigML generates an output dataset with your batch predictions that you can later find in your datasets section of the BigML Dashboard. This option is active by default but you can deactivate it by clicking on the icon shown in Figure 6.40.

ources Datasets Supervi	ised - Unsupervised - Predic	tions - Tasks		WhizzML 🔻
	New Ba	tch Prediction		<b>5</b> - (
£1 PA ⊥ ≈	8 2			
Diabetes diagnosis   Trai	ining (70%)'s × v	Diabetis dia	gnosis dataset   Test (30%)	* *
Diabetes	Thu, 28 Jun 2018 19:55:38	1 <mark>9</mark> 1	1	Ved, 08 Jun 2016 15:39:39
	6 odels	8.1 KB size	10 fields	<b>231</b> instances
Description:	⊙	Description:		$\overline{\mathbf{O}}$
Preview of the prediction file (usin Pregnancies, Glucose, Blood 123, 123, 123, 123, 123, 123, 123, 123 123, 123, 123, 123, 123, 123, 123, 123,	pressure,Skinfold,Insulin,BMI,Di 3,123,ABC,ABC,ABC 3,123,ABC,ABC,ABC 3,123,ABC,ABC,ABC 3,123,ABC,ABC,ABC	abetes pedigree,Age,Dia	betes,weight,Diabetes	

Figure 6.40: Create a dataset from batch prediction

After you configure your batch prediction, click on the green button Predict to generate your batch prediction.

5. When the batch prediction is created, you will be able to **download the CSV file** containing all your dataset instances along with a prediction for each one of them. If you did not disable the option to create a dataset previously explained, you will also be able to access the **output dataset** from the batch prediction view (see Figure 6.41).

ources Da	tasets Superv	ised 🗸 Unsup	ervised 👻 🏼 P	redictions -	Tasks		١	WhizzML 🔻
· · ·	•••	Diabe	etis diagnos	.set   Test (30%	%) with Diabete	es diag	∽.	(≡) <b>*</b> , (
Dia	betes Diagnos	is   Training (70	%)'S	•	Diabetis Dia	ignosis Dataset	Test (30%)	•
Output preview Pregnancies		pressure,Skinfo	old,Insulin,BM	I.Diabetes ped	i anna Ana Diaka	tos wajaht Diabat		
8,183,64,0 10,115,0,0 4,110,92,0 1,189,60,2 7,100,0,0,0 7,107,74,0	5,0133,6,0,627,5 ,0,23,3,0,672,32 ,0,35,3,0,134,29 ,0,37,6,0,191,30 3,846,30,1,0,398 0,30,0,0,484,322 ,0,29,6,0,254,31 8,83,43,3,0,183	2,True,1,True ),False,1,True ),False,1,False 8,59,True,1,True ,True,1,False			Lgree, Age, Dlabe	es, we tyrit, o tubet	es	

Figure 6.41: Download batch prediction CSV file and output dataset

## 6.6.3 Configuring Fusion Predictions

BigML provides several options to configure your batch predictions such as the missing strategy (see Subsection 6.6.3.1), setting a probability threshold (see Subsection 6.6.3.2), default values for your missing numeric values (see Subsection 6.6.3.3), fields mapping (see Subsection 6.6.3.5), and output file settings (see Subsection 6.6.3.6).

#### 6.6.3.1 Missing Strategies

This option is available only when the fusion contains **models and/or ensembles** since the missing strategy has no effect for logistic regressions or deepnets.

When you create a new prediction, BigML will automatically navigate through the corresponding model or ensemble to find the leaf node that best classifies the new instance. However, it may just so happen that your new data (the instances you want to predict) does not have populated values for all the fields used in building the original ensemble. For example, imagine that you are trying to predict diabetes and you have the patient's glucose level and BMI (Body Mass Index) but not his blood pressure. If the model or ensemble arrives at a node where the blood pressure level is required, BigML can handle this missing value by using one of these two strategies:

- Last prediction: it returns the prediction value and probability of the parent node.
- **Proportional**: it combines all predictions beneath the current node based on the data distribution of their child nodes in order to compute the prediction value and probability.

For single predictions you can select either Missing strategies by clicking in the icons shown in Figure 6.42.

Sources Dataset	s Supervised	NEW → Unsupervis	ed - Predict	ions 👻 Tasks			WhizzML 👻
		Pre	edict using Dia	betes   Training (8	D%)	3.	(≡)⁵, (≣)
			Diabetes: ?		?		
	So S	Select a positive cla	ss v Proba	bility threshold: —	0	50%	
Glucose 0		248	35.44% 🗹	<b>BMI</b> 0.0	-0	83.88	19.36% 🗹
<b>Age</b> 6		96	16.11% 🗹	Diabetes pedigree		3.01	14.26% 🕑
Blood pressure	-0	152	7.75% 🗹	Pregnancies		21	3.31%
Skinfold 0		123	2.06%	Insulin 0		1057	) 1.72% 🕑
New prediction nam	e						
Diabetes   Training	g (80%)					E. P	redict

Figure 6.42: Missing strategies for fusion single predictions

For batch predictions you can find both options under the configuration panel as shown in Figure 6.43.

ources Datasets Supervised - Unsupervised - Predictio	ns 🔻 Tasks		WhizzML 👻
New Batch	Prediction		<b>G</b> - (
n fn ∠ ≤ 2			
Diabetes   Training (80%) × ×	Diabetes	Fest (20%)	x v
C Diabetes Tue, 10 Jul 2018 02:49:34	I		Fri, 29 Jun 2018 21:24:35
1 models	5.4 KB size		154 instances
Description:	Description:		$\overline{\mathbf{O}}$
දීසු Configure			۲
Select a positive class   Probability threshold:		50%	
Default numeric value: V Select a default value			
Excluded fields:			$\overline{\mathbf{O}}$
1 Make sure that the fields in the Fusion and Dataset match			
Fields mapping:		De	fault fields

Figure 6.43: Missing strategies for fusion batch predictions

### 6.6.3.2 Probability threshold

Probability thresholds usually makes sense when you want to minimize false positives at the cost of false negatives. The positive class will be predicted if its probability is greater than the given threshold; otherwise, the following class with greater probability will be predicted. This option is only available for classification fusions.

Follow these steps to configure a threshold for your batch prediction:

1. Select the positive class, i.e., the class for which you want to apply the threshold (Figure 6.44).

7.	Predict	using Diabetes diagnosis	Training (70	(≣)⁵- (≣)
	D	abetes: ?	?	
Select a positive class  False False	Probability threshold:	50%		All input fields: 🥳
0	21	Glucose           10         0		244 <b>122</b>
Blood pressure	152	Skinfold           0           76		123 <b>61</b>
nsulin 0	1057	Image: BMI         0.0           528		83.88 40.74
0.0	3.0	Age           1.46	0	95 <b>51</b>
New prediction name				
Diabetes diagnosis   Training	n (70%)'s		e.	Predict

Figure 6.44: Select the positive class

2. Set a probability threshold using the slider shown in Figure 6.45 and click Predict .

	Predic	t using Diabetes di	agnosis   Training (70	<b>C</b> 5.	, (≡)⁵, (
	[	Diabetes: ?	?		
True × • Proba	ability threshold: =		75%		
				All in	put fields: 🕑
Pregnancies		Gluc	cose		۲
°	21	10	0	244	122
Blood pressure		🗹 Skin	fold		5
0	152	76		123	61
nsulin		🕑 вмі			٤
0	1057	0. 528	0	83.88	40.74
Diabetes pedigree		🗹 Age			٤
0.0	3.0	1.46	0	95	51
lew prediction name					

Figure 6.45: Set a probability threshold

3. If the positive class probability is greater than the given threshold, it will be predicted; otherwise, the following class with greater probability will be predicted.



Figure 6.46: Get the prediction

You can also find the same options to set a threshold for batch predictions under the CONFIGURE panel (see Figure 6.47).

Sources Datasets Supervised <del>-</del>	Unsupervised - Predic	tions 🔻 Tasks		WhizzML 👻
1	New Bat	ch Prediction		<b>G</b> - i
£: Fa ∠ ≈ 8	3			
Diabetes diagnosis   Training (70%)!	S X V	Diabetis dia	agnosis dataset   Test (30%)	× v
Diabetes	Thu, 28 Jun 2018 19:55:38	ılî	W	ed, 08 Jun 2016 15:39:39
6 models		8.1 KB size		231 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
Configure				۲
True x v Probability	threshold:	75%		
Default numeric value:  Select a defaul	t value 🔻			
Excluded fields:				$\odot$
1 Make sure that the fields in the Fusior	and Dataset match			
Fields mapping:			Default	fields
2 [OPTIONAL] Customize prediction ou				

Figure 6.47: Configure a probability threhsold for batch predictions

### 6.6.3.3 Default Numeric Value

By using the **Default numeric value** before creating your batch prediction, you can easily replace all the missing numeric values in the dataset by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero** (see Figure 6.48).

ources Datasets Supe	ervised - Unsupervised - Predic	tions 👻 Tasks		WhizzML 🔻
Ŧ	New Bat	tch Prediction		<b>5</b> - (i
ል የል 🔟				
Diabetes diagnosis	Training (70%)'s x	Diabetis diag	nosis dataset   Test (30%)	×
Diabetes	Thu, 28 Jun 2018 19:55:38	ıl		Wed, 08 Jun 2016 15:39:39
	6 models	8.1 KB size	10 fields	231 instances
Description:	$\odot$	Description:		$\overline{\mathbf{O}}$
ີ່ຜູ້ Configure				۲
True × v	Probability threshold:	75%		
Default numeric value: 🛿	Select a default value			
Excluded fields:	Maximum Mean			⊚
1 Make sure that the field	Median Minimum et match			
Fields mapping:	Zero		Defaul	t fields 💿

Figure 6.48: Configure Default numeric value for batch prediction

#### 6.6.3.4 Excluded Fiels

This option allows you to exclude a set of fields from the prediction calculation but at the same time keep them in the output file and dataset (see Figure 6.49).

urces Datasets Supervised ✔ Unst	upervised - Predictio	ns 🔻 Tasks	_	WhizzML 🔻
12	New Batch	Prediction		<b>5</b> - (
£: F£ ∠ ≈ 8				
Diabetes diagnosis   Training (70%)'s	× v	Diabetis diag	nosis dataset   Test (30%	i) × *
Diabetes	Thu, 28 Jun 2018 19:55:38	I		Wed, 08 Jun 2016 15:39:39
6 models		8.1 KB size	10 fields	231 instances
Description:	$\overline{\mathbf{O}}$	Description:		$\overline{\mathbf{O}}$
a Configure				۲
Select a positive class	shold:	50%		
Default numeric value: 🛛 Select a default valu	e 🔻			
Excluded fields:				۲
Excluded fields:				
× Pregnancies				



### 6.6.3.5 Fields Mapping

You can specify which input fields of the fusion match with the fields in the dataset contaning the instances you want to predict. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** by clicking on the green switcher. Additionally, you can **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the batch prediction (see Figure 6.50).

12	New Ba	tch Prediction		∽.
ରେ 🖍 🔟 😆				
Diabetes diagnosis   Training (76	0%)'s × v	Diabetis dia	agnosis dataset   Test (30%)	× v
🕸 🗤 🕂 Diabetes	Thu, 28 Jun 2018 19:55:38	I	Wed, 0	8 Jun 2016 15:39:3
6 models		8.1 KB size	10 fields	231 instances
Description:	$\bigcirc$	Description:		$\overline{\mathbf{O}}$
a Configure				۲
Default numeric value:  Select a de	efault value			
Excluded fields:	efault value v			⊘
				©
Excluded fields:			Apply this ma	
Excluded fields:  1 Make sure that the fields in the Fu		Dataset fields	Apply this ma Auto-mapping by: NAME CO	ap 🛞
Excluded fields:  Make sure that the fields in the Fu Fields mapping:		Dataset fields 1. Pregnancies		ap 🛞
Excluded fields: 1 Make sure that the fields in the Fu Fields mapping: fields	sion and Dataset match	1. Pregnancies 2. Glucose	Auto-mapping by: NAME Control 123 x v	
Excluded fields: 1) Make sure that the fields in the Fu Fields mapping: fields 1. Pregnancies	sion and Dataset match	1. Pregnancies	Auto-mapping by: NAME C	

Figure 6.50: Configure the fields mapping for batch prediction

# Note: Fields mapping from the BigML Dashboard is limited to 200 fields. For batch predictions with a higher number of fields, map your fields using the BigML API⁴.

#### 6.6.3.6 Output Settings

Batch predictions return a CSV file containing all your instances and the final predictions. Tune the following settings to customize your prediction file (see Figure 6.51):

- **Separator**: this option allows you to choose the best separator for your output file columns. The default separator is the comma. You can also select the semicolon, tab, or space.
- New line: this option allows you to set the new line character to use as the line break in the generated csv file: "LF", "CRLF".

⁴https://bigml.com/api/batchpredictions#bp_batch_prediction_arguments

- Output fields: by clicking on the list icon next to the separator selector, you can include or exclude all your dataset fields from your output file. You can also individually select the fields you want to include or exclude using the multiple output fields selector. Note: a maximum of 100 fields can be displayed in this selector, but all your dataset fields will be included in the output file by default unless you exclude them.
- Headers: this option includes or excludes a first row in the output file (and in the output dataset) with the names of each column (input field names, prediction column name, probability column name, etc.). By default, BigML includes the headers.
- Prediction column name: customize the name for your predictions column. By default, BigML takes the name of the fusion's objective field.
- **Probability**: this option allows you to include an additional column with the probability for the predicted class. By default it is not included in your ouput file. For **regression** fusions, you will find the expected error instead of the probability.
- Probability column name: customize the name for the probability column if you include it in the output file. BigML sets "probability" as the default name. For regression fusions, you will find the expected error column name.
- Individual model predictions: this includes all the per-model predictions composing the fusion. That will add a column per model, named <prediction_name>_n where n is the position of the model in the model list in the fusion, starting at 1.
- All class probabilities: this includes all the probabilities of the objective field classes per instance. This option will add *n* extra columns, one by class in the objective field. This option does not exist for **regression** fusions.
- Field importances: this option allows you to include a column for each of the field relative importances for the fusion predictions (taking into account the decision trees, ensembles, and deepnets composing the fusion). This option will add a column per field, named "<field_name> importance".
   If the fusion only contains logistic regressions, these field importances cannot be calculated.

, (comma) Prediction column name a,b,c Output Fields: x Pregnancies		• ﷺ •• ••	
a,b,c Output Fields:			
Output Fields:			
* Pregnancies			
	123 x Glucose	123 🗶 Blood press	sure 123
× Skinfold	123 x Insulin	123 × BMI	123
* Diabetes pedigree	123 × Age	123 × Diabetes	ABC
view of the prediction file (using regnancies, Glucose, Blood p 23, 123, 123, 123, 123, 123, 123 23, 123, 123, 123, 123, 123, 123 23, 123, 123, 123, 123, 123, 123	oressure,Skinfold,Insulin,BMI,Diak 5,123,ABC,ABC,ABC 5,123,ABC,ABC,ABC 5,123,ABC,ABC,ABC 5,123,ABC,ABC,ABC 5,123,ABC,ABC,ABC	petes pedigree,Age,Diabetes,weig	ht,Diabetes

Figure 6.51: Fusion output settings for batch predictions

## 6.6.4 Visualizing Fusion Predictions

Fusion predictions visualizations change depending if you are predicting a **single** instance (Subsection 6.6.4.1), or multiple instances using the **batch predictions** option (Subsection 6.6.4.2).

### 6.6.4.1 Single Predictions

For single predictions, find the predicted class given the input fields values at the top of the form along with its probability (see Figure 6.52).



Figure 6.52: Fusion single prediction

Below the prediction, there's a histogram representing the rest of the objective field **class probabilities**. All the class probabilities must sum to 100%. Show or hide this view by clicking on the icon highlighted in Figure 6.53. You can see up to seven different classes at the same time; if you have more than seven classes, you can see the others by clicking on the **arrows** icons. **Export** this view in PNG, CSV, or JSON format by clicking on the corresponding icons (see Figure 6.53).



Figure 6.53: Fusion all class probabilities

For **regression** fusions, instead of the class probabilities you will get the predicted numeric value for the objective field.

#### 6.6.4.1.1 Prediction explanation

Prediction explanation helps understand why a fusion makes a certain prediction. This is very useful in many applications, and the reasons behind a fusion's prediction are often as important as the prediction itself.

BigML prediction explanation is based on Shapley values. For more information, please refer to this research paper: A Unified Approach to Interpreting Model Predictions [3].

For any classification or regression fusion, you can request the explanation for the prediction by clicking the prediction explanation icon and then click Predict (see Figure 6.54).

Sources Datasets Supervised - Unsuper				WhizzML 🔻
Pre	edict using Diabe	tes diagnosis	Training (70	(≡)⁵-
	Diabetes: ?		?	
Select a positive class    Probability threshold:		50%		
				All input fields: 🕑
Pregnancies	≤	Glucose		•
021	10	0	0	244
Blood pressure	<b></b>	Skinfold		
0 152	76	0	0	123 <b>61</b>
Insulin	€	BMI		
0 1057	528	0.0	0	83.88
Diabetes pedigree	€	Age		
0.0 3.0	1.46	6	6	⁹⁵ <b>51</b>
New prediction name				te the <b>explanation</b> the prediction
Diabetes diagnosis   Training (70%)'s				Predict

Figure 6.54: Explain prediction

The prediction explanation represents the most important factors considered by the fusion in a prediction given the input values. Each input value will yield an associated importance, as you can see Figure 6.55. The importances across all input fields should sum to 100%.

Sources Datas	sets Super	vised 👻 Unsupervised	Predictions	Tasks	WhizzML 🗸
8 7.		Diabo	etes diagnosis   Tra	ining (70%)'s	<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li></ul>
		Dia	betes: True	86.819	10
PREDICTION EXPLAN	ATION				
Input data		Importance			
BMI	40.74				47.54% +
Diabetes pedigree	1.46				26.95% +
Age	51.00				16.73% +
Pregnancies	10.00				5.18% +
Glucose	122.00				2.68% +
Insulin	528.00				0.00%
Blood pressure	76.00				0.00%
Skinfold	61.00				0.00%



For some input fields you will see a "+" icon next to the importance. This is because the importance may not be directly associated with the input value, i.e., it can be explained by other reasons. In the Figure 6.56 below, the importance of the field "Pregnancies" is not explained by this field being equal to 10, rather, it is because this field value is greater than 5.

		B	Diabetes diag	gnosis   Training	(70%)'s	C	5- (≡)⁵- (
			Diabetes: Tr	rue	86.81%		
REDICTION EXPLANA	ITION						PNG CSV IS
nput data		Importance					
BMI	40.74						47.54%
Diabetes pedigree	1.46						26.95% 4
ge	51.00						16.73%
regnancies	10.00						5.18% 4
lucose	122.00					Detailed Exp	lanation 2.58%
nsulin	528.00					>5	5.18%
lood pressure	76.00						0.00%

Figure 6.56: See the detailed explanation

The prediction explanation for fusions is calculated using the results of over a thousand distinct predictions using random perturbations of the input data. For this reason, the calculation of the explanation may take some time to be computed.

Note: the input field importances in the prediction explanation are different from the overall field importances of the fusion. A field can be very important for the fusion but insignificant for a given prediction.

#### 6.6.4.2 Batch Prediction

After creating your batch prediction, you get a **CSV file** and, optionally, an **output dataset**. Both outputs are explained in the following subsections.

#### 6.6.4.2.1 Output CSV file

The batch prediction generates a CSV file containing your **predictions** for each of your dataset instances in the last column (see Figure 6.57).



Figure 6.57: Download batch prediction CSV file

You can configure several options to **customize your CSV file**. You can find a detailed explanation of those options in Subsection 6.6.3.6.

See an output CSV file example in Figure 6.58. The column **class** in this example contains the final prediction (it is named by default as your fusion's objective field). In this case, we are predicting whether a person is a good or a bad candidate for holding a credit. This file has been configured to also contain the **probability** for each prediction.

```
duration,age,amount,purpose,class,probability
24,26,5433,used car,good,0.88785
36,42,8086,new car,bad,0.55526
24,28,1376,radio/tv,good,0.8385
48,31,6758,radio/tv,bad,0.73576
26,30,7966,used car,good,0.7201
12,42,2577,furniture/equipment,good,0.67644
36,30,4455,business,good,0.52227
18,32,1442,new car,bad,0.75488
9,22,276,new car,good,0.57819
```

Figure 6.58: An example of a fusion batch prediction CSV file

#### 6.6.4.2.2 Output Dataset

By default, BigML automatically creates a dataset out of your batch prediction. You can disable this option by configuring your batch prediction (see Subsection 6.6.3.6). You will find the output dataset in your batch prediction view as shown in Figure 6.59.



Figure 6.59: Batch prediction output dataset

In the output dataset, you can find an additional **field** (named by default as your fusion's objective field) containing the **class predicted** for each one of your instances (see Figure 6.60). If you configured your batch prediction to include the prediction **probabilities** and **all class probabilites**, you will be able to find them in the last fields of your output dataset.

	Diabe	tis diagno	sset   Test (3	0%) with Dia	abet 🎭 🝊 (≡)⁵ (
<u>A</u>					Q
Name	†ј Туре	Count	t↓ Missing	t _↓ Errors	tistogram
Pregnancies	123	231	0	0	III
Glucose	123	231	0	0	
Blood pressure	123	231	0	0	
Skinfold	123	231	0	0	haddillillillillilli
insulin	123	231	0	0	111111111111111111111111111111111111111
BMI	123	231	0	0	
Diabetes pedigree	123	231	0	0	illinna
Age	123	231	0	0	
Diabetes	ABC	231	0	0	
weight	ABC	231	0	0	
Diabetes	ABC	231	0	0	
Show 100 🗘 fields		1 */	11 of 11 fields		K < 1 > :

Figure 6.60: Fusion batch prediction output dataset

## 6.6.5 Consuming Fusion Predictions

You can fully use single and batch predictions via the BigML API and bindings. The following subsections explain both tools.

### 6.6.5.1 Using Fusion Predictions via the BigML API

Fusion predictions have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, and delete single and batch predictions.

In the example below, see how to create a single prediction using a fusion and define the input data once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/prediction?$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"fusion": "fusion/50650bdf3c19201b64000020",
                     "input_data": {"000001": 3, "000002":4.5, "000003":5}}}'
```

For more information on using predictions through the BigML API, please refer to the documentation⁵.

### 6.6.5.2 Using Fusion Predictions via BigML Bindings

You can also create, configure, retrieve, list, update, and delete single and batch predictions via **BigML bindings**, which are libraries aimed to make it easier to use the BigML API from your language of choice including Python, Node.js, Java, Swift and Objective-C. See below an example to create a fusion with the Python bindings.

```
from bigml.api import BigML
api = BigML()
prediction = api.create_prediction(
    "fusion/50650bdf3c19201b64000020",
    {"credit_amount": 5, "duration": 2.5})
```

For more information on BigML bindings, please refer to the bindings page⁶.

## 6.6.6 Descriptive Information

Each fusion prediction has an associated **name**, **description**, **category**, and **tags**. You can find a brief description of each concept in the following subsections. The MORE INFO menu option displays a panel that provides editing options (see Figure 6.61).

⁵https://bigml.com/api/predictions

⁶https://bigml.com/tools/bindings

Sources Datasets Supervised -	Unsupervised - Predictions - Tasks	WhizzML 👻
<b>≙ ₩</b> .	Diabetes diagnosis   Training (70%)'s	(≡) ⁵ , (□),
DETAILS		$\overline{\mathbf{O}}$
INFO		۲
Name: Diabetes diagnosis   Training (70%)'s Description:	Category: Healthcare Tags:	
Predict if a patient has diabetes	diagnosis diabetes healthcare	
PRIVACY		$\odot$

Figure 6.61: Fusion prediction descriptive information

#### 6.6.6.1 Name

If you do not specify a **name** for your predictions, BigML assigns a default name depending on the type of predictions:

- Single predictions: the name always follows the structure "<fusion name>".
- **Batch predictions**: BigML combines your prediction dataset name and the fusion name: "<fusion name> with <dataset name>".

Prediction names are displayed on the list and also on the top bar of a prediction view. Prediction names are indexed to be used in searches. Rename your predictions at any time from the MORE INFO menu.

The name of a prediction cannot be longer than 256 characters. More than one prediction can have the same name even within the same project, but they will always have different identifiers.

#### 6.6.6.2 Description

Each prediction also has a **description** that is useful for documenting your Machine Learning projects. Predictions have the same description as the fusion used to create them.

Descriptions can be written using plain text and also markdown⁷. BigML provides a simple markdown editor that accepts a subset of markdown syntax (see Figure 6.62).

⁷https://en.wikipedia.org/wiki/Markdown

'ou can add formatting and links using B I I≣ & ↔	a simple markuown lai	iguage:	
Predict if a patient has dia	abetes		
<b>Description:</b> Predict if a patient has diabetes			

Figure 6.62: Markdown editor for fusion descriptions

Descriptions cannot be longer than 8192 characters.

#### 6.6.6.3 Category

Each prediction has an associated **category** taken from the fusion used to create it. Categories are useful to classify predictions according to the domain which your data comes from, which helps when you use BigML to solve problems across industries or multiple customers.

A prediction category must be one of the categories listed on table Table 4.5.

#### 6.6.6.4 Tags

A prediction can also have a number of **tags** associated with it. These tags help to retrieve the prediction via the BigML API or to provide predictions with some extra information. Your prediction inherits the tags from the fusion used to create it. Each tag is limited to a maximum of 128 characters. Each prediction can have up to 32 different tags.

## 6.6.7 Fusion Predictions Privacy

The link displayed in the **Privacy** panel is the private URL of your prediction, so only a user logged into your account is able to see it. Neither single predictions nor batch predictions can be shared by using a secret link (see Figure 6.63).

Sources Datasets Super	vised - Unsupervised	Predictions      Tasks	Whizz	ML 🛨
a ₽.	Diab	etes diagnosis   Training (70%)'s	<b>८५</b> - (≡) [€] -	0
DETAILS				◙
INFO				◙
PRIVACY				۲
Private link   Only you can view th	is prediction logged into your a	ccount.		
	A https://str	ato.dev.bigml.com/dashboard/prediction/5b354a6f9	83efc15900000bc	



## 6.6.8 Moving Fusion Predictions to Another Project

When you create a prediction, it will be assigned to the same **project** where the original fusion is located. You cannot move predictions between projects as you do with other resources.

## 6.6.9 Stopping Fusion Predictions

Single predictions are **synchronous** resources, so you cannot cancel them during the creation since you get the result immediately.

On the other hand, batch predictions are **asynchronous** resources, so you can stop their creation before the task is finished. Use the DELETE BATCH PREDICTION option from the **1-click action menu** (Figure 6.64) or from the **pop up menu** on the list view.

Sources Datasets	Supervised - Unsupervised - Predictions - Tasks	WhizzML 👻
	Diabetis diagnosset   Test (30%) with Diabetes diag 🥵	(≡)⁵~ (i)
	BATCH PREDICTION AGAIN	
	BATCH PREDICTION USING ANOTHER FUSION	
	DELETE BATCH PREDICTION	
	Processing	
	Your request is being processed and we'll generate a prediction soon	

Figure 6.64: Stop fusion batch prediction from 1-click action menu

A modal window will be displayed asking you for confirmation. If you stop the prediction during its creation you won't be able to resume the same task again. So if you want to create the same prediction, you will have to start a new task.

	Are you sure you want to delete this prediction?
	If you delete this prediction, you will no longer have access to its data and you will need to recreate it.
Dia	Cancel Delete

Figure 6.65: Fusion delete prediction confirmation

## 6.6.10 Deleting Fusion Predictions

You can DELETE your **single or batch predictions** from the predictions view, using the **1-click action menu** (see Figure 6.66) or using the **pop up menu** on the predictions list view (see Figure 6.67).

A 🐨			Diabetes diagnosi	is   Training (70%)'s	<ul> <li>(≡)⁵.</li> </ul>
			Diabetes: True	NEW PREDICTION	
REDICTION EXPLANA	πον	Importance			
змі	40.74				47.54%
Diabetes pedigree	1.46				26.95%
ge	51.00				16.73%
regnancies	10.00				5.18%
alucose	122.00				2.68%
nsulin	528.00				0.00%
lood pressure	76.00				0.00%
Skinfold	61.00				0.00%

Figure 6.66: Delete fusion prediction from 1-click menu

Sources	s Datasets Supervised - Unsupervise	d 🔻 🏻 Pre	dictions - Tasks			WhizzML 🔻
Ŧ	Ē		Predictions			â Q
ŵ	Name	ų.	Objective	4	Prediction	t, 🛗
83	Diabetes diagnosis   Training (70%)'s operating kind=probability, 8 inputs	/ 🗵	Diabetes		True	9m
23	Diabetes diagnosis		liabetes		True	22mi
C2*	Diabetes diagnosis Delete PREDICTION threshold for True at 81% probability, 4 inputs		Jiabetes		False	24m
7 <b>%</b>	Diabetes diagnosis   Training (70%)'s operating kind=probability, 8 inputs		Diabetes		True	40mi
82	Wine Sales operating kind=probability, 5 inputs		Total Sales		78.15923	1
8 <b>%</b>	Titanic Survival v1 operating kind=probability, 3 inputs		Survived		FALSE	1
83	Titanic Survival operating kind=probability, 3 inputs		Survived		FALSE	1
C2	Iris dataset - extended operating kind=probability, 4 inputs		Species		Iris-versicolor	1w 1
2 <b>%</b>	Titanic Survival_two_fields operating kind=probability, 1 inputs		Survived		FALSE	1w 1
83	Stumble_upon.csv operating kind=probability, 24 inputs		label		0	1w3
how 10	predictions	1 to 10	of 581 predictions			< < 1 2 3 4 5 > >

Figure 6.67: Delete fusion prediction from pop up menu

A modal window will be displayed asking you for confirmation. Once a prediction is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

re you sure you want to delete th	nis prediction?
you delete this prediction, you will no longer ecreate it.	r have access to its data and you will need to
	Cancel Delete

Figure 6.68: Delete fusion prediction confirmation

## 6.7 Consuming Fusions

Similarly to other models in BigML, fusions are white-box models, so you can **download** them and use them locally to make predictions. You can also create and consume your fusions programmatically via the **BigML API and bindings**. The following subsections explain these three options.

## 6.7.1 Downloading Fusions

You can download your fusions in several programming languages including JSON PML, Python or Node.js. By downloading your fusions you will be able to compute **predictions locally**, free of latency and at no cost. Click on the download icon in the top menu (see Figure 6.69), and select your preferred option (see Figure 6.70.)

Sources Datasets Supervi			WhitzML + Download Actionable Fusion
	Diabetes diagnosis	s   Training (70%)'s	⊕
6	DIABETES	9	FALSE
			Ø
<b>Y</b> BMI 123	· · · · · · · · · · · · · · · · · · ·		PREDICTION
		-	Probability
60 -		min.	max.

Figure 6.69: Click download icon

Complete actionable fusion	:				
Python	× *				
Python					
Node.js	ings				
<pre>from bigml.fusio from bigml.api in # Downloads and # if it hasn't b fusion = Fusion(</pre>	tps://github.com n import Fusion mport BigML generates a loca een downloaded p 'fusion/5b353d3a i=BigML("teresa	n/bigmlcom/pyt al version of previously. 1983efc1595004	the FUSION, eb7',	61c06471".	

Figure 6.70: Select language to download your fusions

## 6.7.2 Using Fusions Via the BigML API

Fusions have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them for predictions.

In the example below, see how to create a fusion using two existing models once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/fusion?\$BIGML_AUTH" \
    -X POST \
    -H 'content-type: application/json' \
    -d '{"models":["model/5af06df94e17277501000010"
                     "model/5af06df84e17277502000019"
                    "deepnet/5af06df84e17277502000016"
                    "ensemble/5af06df74e1727750100000d"]}'
```

For more information on using fusions through the BigML API, please refer to the documentation⁸.

## 6.7.3 Using Fusions Via the BigML Bindings

You can also create and use fusions via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift and Objective-C. See below an example to create a fusion with the Python bindings.

```
from bigml.api import BigML
api = BigML()
```

```
<sup>8</sup>https://bigml.com/api/fusions
```

For more information on BigML bindings, please refer to the bindings page⁹.

## 6.8 Fusion Limits

There are some limits that apply to the creation of any BigML resource. These are limits based on the number of classes, terms and items that can be considered to create your models. In the case of fusions, the single model limits apply along with some specific limits for the fusion creation and visualization:

- Maximum models: you can select a maximum of 1,000 models to create your fusion.
- PDP limits: the fusion Partial Dependence Plot will not be displayed if:
  - The fusion only has text or items fields as inputs (because only categorical and numeric fields can be plotted on the PDP axes)
  - The objective field has more than 200 categories.
  - The fusion reaches a certain number of models defined by the formula: (decision trees  $+5 \times$  ensembles + logistic regressions  $+10 \times$  deepnets)  $\leq 50$ .

In these cases, the model list will be the default view and the icon to access the PDP will be disabled (see Figure 6.71).

6					Titanic Surviva	I		1	<b>5</b> -	(≡)⁵- (ĵ
Your fusion conta than the maxin show the PDP logistic reg 5'ensembles + 1	mum allov ?: (single t gressions	wed to rees + + ²³⁷	DELS	OBJECTIVE FIELD		FIELDS 5	5		FIELDS MAP	
					5	37	<b>F 1</b> 70	100	<b>X</b>   30	
Num	nber $\mathbf{f}_{\downarrow}$	Model 🖞	Name							t↓ Weight t↓
	1	ភា	Titanic Survival 512-node, pruned,	deterministic order						1
-	2	ŵ	Titanic Survival 512-node, pruned,	deterministic order						1
	3	វា	Titanic Survival 512-node, pruned,	deterministic order						1
	4	ŵ	Titanic Survival 512-node, pruned,	deterministic order						1
ę	5	ណ	Titanic Survival 512-node, pruned,	deterministic order						1
	6	ŵ	Titanic Survival 512-node, pruned,	deterministic order						1
	7	ភា	Titanic Survival 512-node, pruned,	deterministic order						1
8	8	ŵ	Titanic Survival 512-node, pruned,	deterministic order						1
ę	9	ភា	Titanic Survival 512-node, pruned,	deterministic order						1
1	10	ŵ	Titanic Survival 512-node, pruned,	deterministic order						1
Show	w 10	c models			1 to 10 of 237 models			К	< 1 2 3	4 5 > >

Figure 6.71: The PDP cannot be displayed

It may happen that the number of component models does not reach the maximum but they are too big to load the fusion predictions in a reasonable time, in this case, BigML will display a warning message (see Figure 6.72).

⁹https://bigml.com/tools/bindings

Y	· ·	Υ.X.	PREDICT	ION
			- min.	max.
	The PDP cannot be di	isplayed, because your mo predictions in a reasonabl	del is too big to calculate the le time	S All fields:
			confidence	

Figure 6.72: The PDP cannot be displayed because the predictions take too much time to be computed

Note: the PDP limits only affect the fusion visualization, i.e., even if your dataset reaches those limits, you can still create the fusion, evaluate it and use it to make predictions.

Note2: in the case a fusion has more than 100 input fields, only the top 100 fields ordered by importance will be displayed in the PDP view.

## 6.9 Descriptive Information

Each fusion has an associated **name**, **description**, **category**, and **tags**. The following subsections provide a brief description for each concept. In Figure 6.73, you can see the editing options provided in the MORE INFO menu.

Sources Datasets Supervised -	Unsupervised   Predictions  Tasks	WhizzML 👻
ີ ເ <mark>ພີ່</mark> •••••	🚫 Diabetes diagnosis   Training (70%)'s 🛛 📋 💠	ૐ- (≡)⁵- ①_
DETAILS		$\odot$
INFO		۲
Name: Diabetes diagnosis   Training (70%)'s	Category:	Y
Description:	Tags:	
Patient data to predict diabetes	diagnosis diabetes patients	
PRIVACY		$\overline{\mathbf{O}}$

Figure 6.73: Edit fusion descriptive information

## 6.9.1 Fusion Name

Each fusion has a name that is displayed in the fusion list view and also on the top bar of the fusion view. Fusion names are indexed to be used in searches. A fusion inherits the name from the first model used to create it. Change it using the MORE INFO menu option on the right corner of the fusion view. The name of a fusion cannot be longer than **256** characters. More than one fusion can have the same name even within the same project, but they will always have different identifiers.

## 6.9.2 Description

Each fusion also has a **description** that it is very useful for documenting your Machine Learning projects.

Descriptions can be written using plain text and also markdown¹⁰. BigML provides a simple markdown editor that accepts a subset of markdown syntax (see Figure 6.74).

¹⁰https://en.wikipedia.org/wiki/Markdown

You can add formatting and links using a simple markdown lang	made.
B $I \equiv \mathscr{O} \diamond \diamond$	
Patient data to predict diabetes	
Description:	
Patient data to predict diabetes	

Figure 6.74: Markdown editor for fusion descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

## 6.9.3 Category

A **category** can be associated with each fusion. Categories are useful to classify fusions according to the domain which your data comes from. This is helpful when you use BigML to solve problems across industries or for multiple customers.

A fusion category must be one of the 24 categories listed on Table 6.1.

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 6.1: Categories used to classify fusions on BigML

## 6.9.4 Tags

A fusion can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide fusions with some extra information. Each tag is limited to a maximum of 128 characters. Each fusion can have up to 32 different tags.

### 6.9.5 Counters

For each fusion, BigML also stores a number of counters to track the number of other resources that have been created using the fusion as a starting point. In the fusion view, you can see a menu option that displays counters for evaluations, single and batch predictions, and the fusions created using this fusion. It also allows you to quickly jump to all the resources of each type (see Figure 6.75).

	••••	Q	Diabetes di	agnosis   Trai	ning (70%)'s	Ē	G 🝊 (=	·)*_ (
	VIEW 4 PREDICTIONS	FROM THIS FUSION	E FIELD		FIELDS		FIELDS MAP	
[	2 VIEW 2 BATCH PREDI	CTIONS FROM THIS FUS	I TES		9		FALSE	
	NO EVALUATIONS FRO	OM THIS FUSION YET						
	NO FUSIONS THAT INC	CLUDE THIS FUSION YET						0
YE	BMI 123	3	-	£3				
						PREDICTI	JN	

Figure 6.75: Counters for fusions

## 6.10 Fusion Privacy

Privacy options for a fusion can be defined in the **More Info** panel, displayed in Figure 6.76. There are two levels of privacy for BigML fusions:

- **Private**: only accessible by authorized users (the owner and those who have been granted access by him or her).
- **Shared**: by enabling the secret link you will get two different links to share your model. The first one is a sharing link that you can copy and send to others so they can visualize and interact with your model. The second one is a link to embed your model directly on your web page.

••••	Titanic Survival	Ē	$\mathbf{\Phi}$	<b>\$</b> -	(≡)⁵-	(
DETAILS						0
INFO						0
						0
						(
Private link   Only you can view this	fusion logged into your account.	2222000000				(
	https://bigml.com/dashboard/fusion/5b2bb759983efc2	222200000				e
	https://bigml.com/dashboard/fusion/5b2bb759983efc2					

Figure 6.76: Fusions privacy

## 6.11 Moving Fusions to Another Project

When you create a fusion within a project, it will be assigned to this project.

Fusions can only be assigned to a single project. However, you can move fusions between your Dashboard projects. The menu option to do this can be found in two places:

1. In the fusion view, click the MOVE TO... option within the **1-click action menu** and select another project or create a new one (see Figure 6.77).

Sources Datasets Supervised	Visupervised - Predictions	r Tasks	WhizzML 🔻
A	🚫 Titanic Sur	vival 📋 🗘	ૐ- (≡)⁵- ③
TOTAL MODELS	OBJECTIVE FIELD	BATCH PREDICT	
		CREATE FUSION USING THIS ONE	
Y Joined ABC	NEW PROJECT	MOVE TO PREDICTION	_
	PROJECTS CUSTOMER CHURN		Probability
	BIGML ARCHIVE	min.	max.
Southampton	PREDICTIVE MAINTENANCE	FALSE	
		TRUE	
Queenstown		All classes	v

Figure 6.77: Change project from 1-click action menu

2. In the fusion list view, click the MOVE TO... option within the **pop up menu** and select another project or create a new one (see Figure 6.78).

Sources Datasets	Sources Datasets Supervised - Unsupervised - Predictions - Tasks					WhizzML 👻		
			Fusions		Ø	Î <	<b>5</b>	
Name	t,	Туре	Objective	t,	<b>\$</b>			
Wine Sales 2 total models (model: 2)		٨	Total Sales	3d 5h		2	1	
Iris_missing 4 total models (deepnet: 1, er	nsemble: 1, logisticregression: 1	00	Species	4d 4h		2		
Iris_missing v1 2 total models (logisticregression: 2)		00	Species	4d 7h		2		
Iris_missing 1 total model (logisticregress	ion: 1)	00	Species	4d 7h				
Titanic Survival 1 total model (logisticregress	ion: 1)	00	Survived	4d 7h				
2 total models (logisticrogr		00	Survived	4d 7h	1			
Titanic Survival	CREATE BATCH PREDICTION		NEW PROJECT	4d 7h				
1 total model (logisticregre	CREATE FUSION USING THIS ONE	PROJ		4d 8h	1			
Arrythmia diagnosis	VIEW DETAILS		USTOMER CHURN	4d 8h				
Stumble_upon.csv 2 total models (logisticregr	мочето	P	REDICTIVE MAINTENANCE	4d 8h				
Show 10 🗘 fusions		11 to 2	0 of 58 fusions		K < 1	2 3 4 5	5 > >	

Figure 6.78: Change project from pop up menu

Note: you cannot move fusions to an organization project since the fusion is composed by many models and it cannot be separated from them.

## 6.12 Stopping Fusions

You can stop the creation of a fusion before the task is finished by clicking the DELETE FUSION option from the **1-click action menu** (see Figure 6.79), or from the **pop up menu** in the fusion list view (see

### Figure 6.80).



Figure 6.79: Stop fusion creation from 1-click action menu

Sources Datasets	Supervised - Unsupervised -	Predi	ictions <del>-</del> Tasks	_		Whizz	zML 🔻
			Fusions		Ø	(	<b>5</b> -
Name	ţ,	Туре	Objective	<b>***</b> 1,	\$	<b>(</b>	
Stumble_upon.csv Queued	1 🗵	-	-	2min			
Arrythmia diagnosis [filte 2 total models (logisticregr	PREDICT	•0 <b>1</b>	Arrhythmia	1h 47min			
Arrythmia diagnosis [filte 2 total models (logisticregr	CREATE BATCH PREDICTION	•0 <b>1</b>	Arrhythmia	1h 49min			
Arrythmia diagnosis [filte 1 total model (logisticregre	CREATE FUSION USING THIS ONE	o0	Arrhythmia	1h 49min			
<b>Iris dataset - extended</b> 1 total model (logisticregre	VIEW DETAILS	00	Species	1h 52min			
Arrythmia diagnosis data 1 total model (logisticregre	MOVE TO	>	Arrhythmia	1h 53min			

Figure 6.80: Stop fusion creation from pop up menu

A modal window will be displayed asking you for confirmation and if you want to also delete all the models composing the fusion. The models that belong to another fusion or to an OptiML will not be deleted. If you stop the fusion during its creation, you will not be able to resume the same task. If you want to create the same fusion, you will have to start a new task.

shboard	The All	_
ets [	Are you sure you want to delete this fusion?	
	If you delete this fusion it will be permanently removed. However, by default the resources created by this fusion will not be removed unless you choose the option below.	
iltered] agression iltered] agression iltered] gression	<ul> <li>Please confirm this operation. Every resource created by this fusion will be deleted permanently. This action cannot be undone.</li> <li>Type "DELETE" in the following input box to confirm that you understand.</li> <li>I want to this fusion and all of the resources it created.</li> <li>The models that belong to another Fusion or OptiML will not be deleted.</li> </ul>	
gression: ataset's	Cancel Delete	

Figure 6.81: Confirmation message to delete a fusion

## 6.13 Deleting Fusions

You can delete your fusions by clicking on the DELETE FUSION option from the **1-click action menu** (see Figure 6.82) or using the **pop up menu** on the fusion list (see Figure 6.83).

Sources Datasets Superv	vised - Unsupervised -	Predictions -	Tasks	WhizzML 🔻
<b></b>		Titanic Surviva	al 📋 🗘	0 ໕, (≡)⁵, ()
TOTAL MODELS	OBJECTIVE FIELD		BATCH PREDICT	
			CREATE FUSION USING THIS ONE	
Y Joined ABC	-	Ęž	MOVE TO PREDICTION	
			- min.	Probability max.
Southampton			FALSE	
			TRUE	

Figure 6.82: Delete fusions from 1-click action menu
Sources Datasets	Supervised - Unsupervised	<ul> <li>Predi</li> </ul>	ctions 🛨 Tasks			Whizzl	ИL <del>-</del>
			Fusions			Î <	<b>5</b> -
Name	ţ	Туре	Objective	t,	<b>\$</b>	<b>I</b>	
Wine Sales 2 total models (model: 2)	r (8	入	Total Sales	3d 5h		2	1
Iris_missing 4 total models (deepnet: 1,		•O <b>I</b>	Species	4d 4h		2	
Iris_missing v1 2 total models (logisticregr	CREATE BATCH PREDICTION		Species	4d 7h		2	
Iris_missing 1 total model (logisticregre	CREATE FUSION USING THIS ONE		Species	4d 7h			
Titanic Survival 1 total model (logisticregre	VIEW DETAILS	00	Survived	4d 7h			
Titanic Survival v1 2 total models (logisticregr	MOVE TO	>	Survived	4d 7h	1		
Titanic Survival 1 total model (logisticregres	ssion: 1)	00	Survived	4d 7h			

Figure 6.83: Delete fusion from pop up menu

A modal window will be displayed asking you for confirmation and if you want to also delete all the models composing the fusion. The models that belong to another fusion or to an OptiML will not be deleted. Once a fusion or its models are deleted, they are permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

shboard	The All	_	
ets	Are you sure you want to delete this fusion?		
	If you delete this fusion it will be permanently removed. However, by default the resources created by this fusion will not be removed unless you choose the option below.		Í
iitered] sgression iltered] sgression iltered]	<ul> <li>Please confirm this operation. Every resource created by this fusion will be deleted permanently. This action cannot be undone.</li> <li>Type "DELETE" in the following input box to confirm that you understand.</li> <li>I want to this fusion and all of the resources it created.</li> <li>The models that belong to another Fusion or OptiML will not be deleted.</li> </ul>		
gression gression ataset's	Cancel Delete		

Figure 6.84: Confirmation message to delete a fusion

# 6.14 Takeaways

This chapter explains fusions in detail. Here is a list of key points:

- A fusion is a supervised Machine Learning algorithm used to solve classification and regression problems.
- A fusion is built by selecting multiple models in BigML and using to make an evaluation, a single prediction, or a batch prediction (see Figure 6.85).
- · You can create a fusion using models, ensembles, logistic regression and/or deepnets.
- You can assign different weights to the models composing a fusion.
- All models need to have the same objective field.
- The PDP view provides a visual way to analyze a field's impact on predictions given certain values for the rest of the fields.
- For classification problems, you get all the objective field class probabilities along with the predicted class.
- · For regression problems, you get the objective field predicted values.
- · BigML lists all the models composing the fusion along with their weights.
- You need to evaluate your fusion's performance using data that the fusion has not seen before.
- The ultimate goal in building a fusion is being able to make predictions with it.
- BigML allows you to quickly make predictions for single instances by providing a form containing the fields used by the fusion, so you can easily set the input data and get an immediate response.
- BigML batch predictions allow you to make simultaneous predictions for multiple instances. All you
  need is the fusion you want to use to make predictions and a dataset containing the instances for
  which you want to obtain predictions.
- · You can configure your batch predictions output file settings.
- · You can download your fusion to perform local predictions.
- You can add descriptive information to your fusion (name, description, tags, and category).
- You can move your fusions between projects.
- · You can share your fusions with other people using the secret link.
- You can **stop** your fusions creation by deleting them.
- You can permanently delete an existing fusion.



Figure 6.85: Fusion Workflow



# **Evaluations**

# 7.1 Introduction

BigML evaluations provide an easy way to measure and compare the performance of classification and regression models (i.e., models, ensembles, logistic regressions, deepnets, and fusions created using supervised learning algorithms). The main purpose of evaluations is twofold:

- First, obtaining an estimation of the model's performance in production (i.e., making predictions for new instances the model has never seen before).
- Second, providing a framework to compare models built using different configurations or different algorithms to help identify the models with best predictive performance.

The basic idea behind evaluations is to take some test data different from the one used to train a model and create a prediction for every instance. Then compare the actual objective field values of the instances in the test data against the predictions and compute several performance measures based on the correct results as well as the errors made by the model.

The usual way to obtain some test data is to split a dataset into two disjoint subsets: a training set and a test set. You can easily do this from your BigML Dashboard by using the 1-click menu option that automatically splits your dataset into a random 80% subset for training and 20% for testing, or if you prefer, you can also configure those percentages. Chapter 7 in **Datasets with the BigML Dashboard document** [9] explains how to do this.

Depending on whether you evaluate a **classification** or a **regression** model different metrics and visualizations are provided as **Subsection** 7.2.1 and **Subsection** 7.2.2 describe. In BigML you can also perform **cross-validation**, another popular model evaluation technique explained in **Subsection** 7.3.6. Moreover BigML provides some tools so you can **compare** several **evaluations** built with different algorithms and configurations explained in <u>Section</u> 7.6.

BigML evaluations are first-class citizens. This means that they can be created via the BigML API and can also be queried automatically (you can find an example in Subsection 7.7.1). This allows you to **automate workflows** when you want to iteratively change your model parameters and see how the performance is altered.

The evaluations section in the BigML Dashboard is found in the third tab under the models' menu (see Figure 7.1.) This section contains all your model's evaluations ordered by creation date so most recent evaluations are found at the top of the list. Order your evaluations by **Name**, by **Type** (classification or regression and cross-validation icons below), by **Performance** (f-measure or r-squared explained in Subsection 7.2.1.3 and Subsection 7.2.2 respectively), by **Age** (time since the evaluation was created), **Dataset Size** (the testing dataset weight) and **Instances** (number of instances in the testing dataset). (See Figure 7.2.) The icons on the left of the evaluations names in the **evaluation list view** allow you to go to the original resources used to create the evaluation (the model, ensemble, logistic regression, deepnet, or fusion and the testing dataset).

Sources Datasets Models	Clusters Anomalies Associations Predic	tions Tasks		S	Cripts 🔻
MODELS	Sources		-	- {** <b>}[†] P⁺</b>	D*
Type S Nam EVALUATIONS		÷ 🛱	•	â \$	ıll °
Fictional Wine Sales			8m 1w	51.9 KB	1
CSV Titanic Survival			8m 1w	78.0 KB	2
US Car Accidents in 2011			8m 1w	685.5 KB	6
Premier League 2011-2012 Se	ason		8m 1w	24.7 KB	1
Churn in the Telecom Industry	/		8m 1w	270.4 KB	10
Arrhythmia Diagnosis			8m 1w	533.6 KB	1
2012 POTUS Winner by Count	ty .		8m 1w	262.9 KB	1
Diabetes Diagnosis			8m 1w	26.8 KB	3
Show 10 0 sources	21 to 28 of 28 sources			K < 1 2 3	>>

Figure 7.1: Evaluations section in the BigML Dashboard

Sources	Data	sets Evaluations -	Clusters Anomalies	Associations	Predictions 1	asks	So	cripts -
				Evaluations			Q	4
ណ	ıll	Name		🗘 Type 🔇	Performance \$	<b>#</b> \$	🔒 🗘 Ins	stances
f 💦	ılî	Evaluation of Fictional W	/ine Sales dataset with F	木		0min	10.1 KB	318
frå	ıl <mark>e</mark>	Evaluation of Fictional W	line Sales dataset with F	木		3min	10.1 KB	318
5. 1	ıl <mark>o</mark>	Evaluation of Fictional W	line Sales dataset with F	木		4min	10.1 KB	318
få	ıl <mark>e</mark>	Evaluation of Diabetes d	iagnosis dataset - balanc	00		19min	5.4 KB	154
Kî.	ılî	Evaluation of Diabetes d	iagnosis dataset - balanc	00		19min	5.4 KB	154
ŕå	ıl <mark>e</mark>	Evaluation of Diabetes d	iagnosis dataset - sample	00		21min	5.4 KB	154
Kî.	ılı	Evaluation of Diabetes d	iagnosis dataset - sample	00		21min	5.4 KB	154
how 10	ᅌ evalu	ations	1 to	7 of 7 evaluations			K < 1	$\rightarrow$ >1

Figure 7.2: Evaluation list view

In the **evaluation list view**, you can search evaluations by name by clicking the SEARCH menu option in the top right corner. You can also access the **1-click action menu** shown in Figure 7.3:

- EVALUATE A MODEL: create a new evaluation by selecting a model and a testing dataset (see Subsection 7.3.1)
- EVALUATE AN ENSEMBLE: create a new evaluation by selecting an ensemble and a testing dataset (see Subsection 7.3.2)
- EVALUATE A LOGISTIC REGRESSION: create a new evaluation by selecting a logistic regression and a testing dataset (see Subsection 7.3.3)
- EVALUATE A DEEPNET: create a new evaluation by selecting a deepnet and a testing dataset (see Subsection 7.3.4)
- EVALUATE A FUSION: create a new evaluation by selecting a fusion and a testing dataset (see Subsection 7.3.5)

• **COMPARE EVALUATIONS**: compare two existing evaluations side by side (see Subsection 7.6.1)

• COMPARE MULTIPLE EVALUATIONS: compare multiple existing evaluations (see Subsection 7.6.2)

Source	es D	atasets Supervised - Unsupervised -	Predictions -	Tasks		W	/hizzML 🔻
			Evaluation	5		Ô	۹ 🗳
វា	ill	Name	†↓ Туре	t Performance	EVALUATE A MODE	L	
C3	ılî	Iris dataset - extended vs. Iris dataset v2 operating kind=probability	00		EVALUATE AN ENSE		
83	ılî	Stumble_upon.csv vs. Stumble_upon.csv operating kind=probability	00		EVALUATE A LOGIS		ION
C3^	ılî	Wine Sales vs. Fictional Wine Sales operating kind=probability	*		EVALUATE A FUSIO		
83	ılî	Titanic Survival vs. Titanic Survival operating kind=probability	00		EVALUATE A TIME S		
8	ılı	Stumble_upon.csv vs. Stumble_upon.csv operating kind=probability	00		COMPARE MULTIPL	E EVALUATIO	NS
к <mark>Р</mark>	ılî	Diabetes diagnosis 512-node, pruned, operating kind=probability	00		5m 2w	27.0 KB	76
<b>_</b> *	ılî	Loan risk evaluation operating kind=probability	00	_	5m 3w	716.0 KB	4K
° <mark>8</mark>	ılî	Fictional Wine Sales vs. Fictional Wine Sales 512-node, 10-model bootstrap decision forest, oper	<b>*</b>		5m 3w	50.4 KB	1K
° <mark>8</mark>	ılî	Fictional Wine Sales vs. Fictional Wine Sales 512-node, 10-model bootstrap decision forest, oper	<b>^</b>		⊃ 5m 3w	50.4 KB	1K
° <mark>8</mark>	ılî	Diabetes diagnosis dataset   Training (70%) v4 vs 512-node, 31-model bootstrap decision forest, thres	_ []		5m 3w	8.1 KB	23
how 1	0 ᅌ ev	valuations	1 to 10 of 461 evaluations	5	I< <	1 2 3 4	↓ 5 <b>&gt; &gt;</b>

Figure 7.3: Evaluation list view 1-click action menu

Note: time series evalautions are explained in the document Time Series with the BigML Dashboard [12].

A **Name**, **Description**, **Category**, and **Tags** are associated with each evaluation, which can be helpful to retrieve and document your projects. (See Section 7.9.) **Share** your evaluation with others by using the secret link (see Section 7.10.)

An evaluation is associated with the same project which the model, ensemble or logistic regression belongs to. **Move** an evaluation between projects (see Section 7.11,) or **delete** it permanently from your account. (See Section 7.13.)

Finally, you can see the corresponding icons used to represent a single **evaluation** and **cross-validation** evaluations in Figure 7.4 and Figure 7.5, respectively.



Figure 7.4: Evaluations icon



Figure 7.5: Cross-validation icon

Depending on the evaluations Type, you can find the following icons in the evaluation list view:

Figure 7.6: Classification evaluations icon



Figure 7.7: Regression evaluations icon



Figure 7.8: Classification cross-validation icon



Figure 7.9: Regression cross-validation icon

# 7.2 Understanding Evaluations

In this section, we are going to describe the technicalities behind evaluations. The main goal of evaluating your model is to measure its **predictive performance**. BigML provides two different ways to measure your model performance: by creating single evaluations or cross-validation evaluations.

- Single evaluations are available for models, ensembles, logistic regressions, deepnets, and fusions. The basic idea is to make predictions for instances for which the objective field values are known, but the model has not seen before. The dataset containing those instances is called the testing dataset. Based on the comparison between the predicted and the actual values of the testing dataset, a set of performance measures are calculated. The usual way to obtain the testing dataset is to split the original dataset into two disjoint subsets: a training set and a test set. You can easily do this by using the 1-click menu option that automatically splits your dataset into a random 80% subset for training and 20% for testing. (Subsection 7.1 in Datasets with the BigML Dashboard document [9] explains how to do this.)
- **Cross-validation** evaluations are available for models, ensembles, logistic regressions, and deepnets. In particular, BigML uses k-fold cross-validation¹. To create a cross-validation evaluation you just need a dataset as input. BigML then automatically splits your dataset in k complementary subsets. One of the subsets is used for evaluation while the rest of the data is used for training the model (the remaining k 1 subsets). This process is performed k times, each time using different parts of the data for training and testing the models. Finally, cross-validation yields k different

¹https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation

models and k evaluations. The results of the k evaluations are averaged to obtain the final performance measures. Cross-validation evaluations usually yield more accurate results than single evaluations since they avoid the potential error derived from randomly selecting a too optimistic testing dataset.

Both types of evaluations yield the same measures except for the evaluation curves, which will not be shown for the final averaged cross-validation. However, different performance measures will be obtained depending on the objective field type: numeric or categorical. In the following subsections you can find a detailed explanation of **classification measures** and **regression measures** provided by BigML. You can also find a brief subsection at the end mentioning the specificities of **cross-validation evaluation measures**.

# 7.2.1 Classification Measures

The performance of a classification model is usually represented by a confusion matrix, which contains the actual and the predicted values by the model. The evaluation measures are calculated based on the confusion matrix and the concept of positive and negative classes.

In the following subsections you can find a brief explanation of **positive and negative** concepts, the **confusion matrix**, the **probability threshold**, the **classification measures** and the **evaluation curves** provided by BigML.

#### 7.2.1.1 Positive and Negative Classes

To understand the confusion matrix and the classification measures explained in Subsection 7.2.1.2 and Subsection 7.2.1.3 respectively, it's important to have a general idea of the meanings of positive and negative classes.

In Machine Learning, by convention, the objective field classes are often referred to as **positive** and **negative** classes. The positive class is the class that is more important to accurately predict; e.g., if you are predicting cancer and you have two classes for the objective field, "true" and "false", you can afford some mistakes in predicting when cancer is "false", but it's essential to identify the "true" cases. In other words, the positive class should be the one that has a greater cost of making a prediction error. The rest of classes will then be considered negative classes. The positive class is usually the minority class and the negative class is the majority class in binary classification problems because in many cases, it is often more interesting to predict the rare cases rather than the most evident and common ones.

Considering a class positive or negative is not related to its labels or meaning, so in many cases the positive class can be the "bad" outcome and the negative class the "good" outcome. For example, again consider disease diagnosis, where "true" indicates the patient has the disease, and it is usually considered the positive class.

#### 7.2.1.2 Confusion Matrix

A common method to analyze the model predictive performance is the confusion matrix. The confusion matrix is a table containing the predictions and the actual values for the objective field classes so you can visualize the correct decisions as well as the errors made by the classifier.

In BigML, the **columns** represent the **actual values** and the **rows** represent the **predictions** for each of the classes by default. You can transpose rows by columns by clicking the switcher. (See Figure 7.10.) The intersection between actual values and predicted values yields four possible situations:

- True Positives (TP): positive instances correctly classified
- False Positives (FP): negative instances classified as positive
- True Negatives (TN): negative instances correctly classified as non-positive
- False Negatives (FN): positive instances classified as negative

A model with n classes will yield a confusion matrix with n columns and n rows (see Figure 7.10). However, confusion matrices for the evaluation curves (see Subsection 7.2.1.5) will always yield 2 columns and 2 rows because negative classes are aggregated as if they were one class (you can find more information  $here^2$ ).



Figure 7.10: Confusion Matrix example

The cell colors in the confusion matrix indicate the TP, FP, TN and FN values. In the diagonal of the table you can find the correct predictions, the TP and TN.³ See Subsection 7.2.1.4 to find out how to select the positive class.



Figure 7.11: Cell colors indicate the class selected as positive

#### 7.2.1.3 Classification Measures

BigML provides different metrics to measure your model's performance :

- Accuracy
- Precision
- Recall
- F-measure
- Phi Coefficient
- Macro-averages
- Kendall's Tau
- Spearman's Rho

²https://en.wikipedia.org/wiki/Confusion_matrix

³Note that all negative classes are aggregated so the TN may include incorrect predictions for the negative classes. E.g., imagine three classes, a, b and c where the positive class is a while b and c are the negative classes. The instances of the b class incorrectly predicted as c as well as the instances of the c class predicted as b will be considered TN since they are "negative instances correctly classified as non-positive".

You can find an explanation of each measure in the following subsections. All the measures except the Kendall's Tau and the Spearman's Rho are derived from the confusion matrices and they change according to the positive class and the threshold selected as explained in Subsection 7.2.1.4. Some of these measures are used to display the evaluation curves explained in Subsection 7.2.1.5.

#### 7.2.1.3.1 Accuracy

Accuracy is calculated as the number of correctly classified instances over the total instances evaluated.

TP + TN



Figure 7.12: Accuracy example

Accuracy remains a popular measure for model performance since it is very easy to calculate, but for many real life problems it is too simplistic and misleading. One of the most obvious is when the model has to deal with unbalanced classes. For example, suppose that we get 90% of accuracy in a binary classification model for which you have 900 instances for one of the classes and 100 for the other one. A 90% accuracy is reachable just by classifying all the 1,000 instances as the majority class. This is why it is very important to take into account two more measures, **Precision** and **Recall** (explained below).

#### 7.2.1.3.2 Precision

**Precision** is the percentage of correctly predicted instances over the total instances predicted for the positive class. (See Figure 7.13.)

$$\mathsf{Precision} = \frac{TP}{TP + FP}$$



Figure 7.13: Precision example

#### 7.2.1.3.3 Recall

Recall is the percentage of correctly classified instances over the total actual instances for the positive class. (See Figure 7.14.)

$$\mathbf{Recall} = \frac{TP}{TP + FN}$$

					TP FN	FP
ACTUAL VS. PREDICTED	False	True	ACTUAL	RECALL	F	Phi
False		25	142	82.39%	0.78	0.37
True	42	47	89	52.81%	0.58	0.37
PREDICTED	159	72	231	67.60% AVG. RECALL	0.68 AVG. F	0.37 AVG. Pr
PRECISION	73.58%	65.28%	69.43% AVG. PRECISION	71.00% ACCURACY		

Figure 7.14: Recall example

### 7.2.1.3.4 F-measure

The **F-measure**, also called the **F-score**⁴, is the balanced harmonic mean between **Precision** and **Recall**. The **F-measure** is often a more useful metric than accuracy since poor performance in either **Precision** or **Recall** will result in a low **F-measure** value. It can range between 0 and 1. Higher values indicate better performance.



Figure 7.15: F-measure example

# 7.2.1.3.5 Phi Coefficient

**Phi Coefficient**, also called the Mathews Correlation Coefficient⁵, it is the correlation coefficient between the predicted and actual values. It returns a value between -1 and 1. A coefficient of -1 negative correlation between predictions and actual values; a 0 indicates the prediction is not any better than random, and a coefficient of 1 indicates a perfect prediction.

Phi Coefficient = 
$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$



Figure 7.16: Phi Coefficient example

⁴https://en.wikipedia.org/wiki/F1_score

⁵https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

BigML also reports the **maximum phi coefficient**, which is the highest phi coefficient given all possible thresholds (see Subsection 7.2.1.4). You can find it as a mark in the threshold slider as shown in Figure 7.17.

Sources Datasets	Supervised - Unst	upervised 👻 Predic	tions <del>v</del> Tasks			WhizzML 🗸
a (₩° <b>T</b> . ••••	•	.oan risk dataset   1	Training (80%) vs. L	oan risk	چ.	(≡)∕~
Loan Risk D	ataset   Training (80	%) 💿	D Dataset	does not exist		
						19 <b>(</b> %
Positive class: bad		v	Probabili	ty threshold:	).7148	31%
				ТР	FN FP TI	•
ACTUAL VS. PREDICTED	bad	good	ACTUAL	RECALL	F	Phi
bad		14	121	88.43%	0.80	0.71
good	38	241	279	86.38%	0.90	0.71
PREDICTED	145	255	400	87.40% AVG. RECALL	0.85 AVG. F	0.71 AVG. Phi
PRECISION	73.79%	94.51%	84.15% AVG. PRECISION	87.00% ACCURACY		

Figure 7.17: Maximum Phi Coefficient

To find out more about classification performance measures, refer to this paper [5] written by BigML's VP of Machine Learning algorithms, Charles Parker.

#### 7.2.1.3.6 Macro-averages

As it is explained in the previous subsections, classification measures are computed per class, except in the case of **Accuracy** which is the only measure that is always computed for the overall model.

BigML computes the average of per class measures to measure the overall model performance. Those global statistics are called the **macro-averages** of the measures since they are computed by giving equal weight to all classes. You can find them in the **evaluation view** under the names of **Average Precision**, **Average Recall**, **Average F-Measure** and **Average Phi** as shown in Figure 7.18.



Figure 7.18: Macro-averaged measures

Read more on macro-averaging in this paper [4].

#### 7.2.1.3.7 Kendall's Tau

The Kendall's tau and the Spearman's rho coefficients (see Subsection 7.2.1.3.8) are the only measures that do not have a formula based on confusion matrices. The Kendall's tau is based on all possible

pair of rankings considering each instance in the testing dataset. It measures the degree of correlation between the ranked instances and it can take values between -1 and 1. You can find a more detailed explanation on how the Kendall's tau coefficient is computed in this section.

Each instance in the testing dataset has an **actual class** for the objective field and a **probability score** result of the model predictions. The positive class is assigned a value of 1 and the negative class a value of 0. If for two different instances a and b, score(a)>score(b) and class(a)>class(b) or score(a)<score(b) and class(a)<class(b), the pair of instances is "concordant" because the ordering of scores matches the actual classes, if it doesn't, the pair is "discordant".

Letting *C* be concordant pairs and *D* discordant pairs, the tau coefficient is calculated as follows:

Kendall's tau = 
$$\frac{C - D}{C + D}$$

BigML specifically computes Kendall's tau-b coefficient⁶, which makes an adjustment in the denominator for pairs that are tied. A pair is tied when the score(a)=class(a) or score(b)=class(b). In this case, the pair is neither concordant nor discordant, so the coefficient denominator needs to be modified to keep the range of values [-1, 1].



Figure 7.19: Kendall's Tau-b coefficient

# 7.2.1.3.8 Spearman's Rho

Similar to the Kendall's tau explained int he previous subsection, the Spearman's rho⁷ coefficient measures the degree of correlation between the model predictions and the testing dataset actual values. It computes the Pearson correlation coefficient⁸, between the ranks of the instances and it can take values between -1 and 1. Values closer to 1 indicate a perfect positive correlation which means a better

⁶https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient#Tau-b

⁷https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

⁸https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

performing model. On the other hand, closer values to -1 indicate a worse performing model. A value of 0 indicates that the model is no better than another model making random predictions. See the formula below:

Spearman's rho = 
$$\rho_{rg_X, rg_Y} = rac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}}$$

The  $cov(rg_X, rg_Y)$  is the covariance of the ranked values and the  $\sigma_{rg_x}$  and  $\sigma_{rg_y}$  are the standard deviations of the ranked values.



Figure 7.20: Spearman's Rho coefficient

#### 7.2.1.4 Confidence, Probability and Vote Thresholds

**Classification models** in BigML always return a **confidence** and/or a **probability** for each prediction, i.e. a percentage between 0% and 100% that measures the certainty of the prediction. See <u>Subsection 1.2.6.2</u>, <u>Subsection 2.2.1.2</u>, and <u>Section 4.2</u> to know how BigML calculates probabilities for models, ensembles and logistic regression. Deepnets and fusions also return a probability measure.

When evaluating or predicting with your model you can set a **probability or confidence threshold** for a selected class, known as the **positive class**, so the model only predicts the positive class if the probability or the confidence is greater than the threshold set, otherwise it predicts the negative class. For example, imagine the following diabetes predictions for three different patients:

Patients	Diabetes False	Diabetes True
Patient 1	80%	20%
Patient 2	10%	90%
Patient 3	60%	40%

Table 7.1: Example of diabetes prediction probabilities for three different patients

Without setting any threshold, just by looking at the probabilities for each predicted class, patients 1 and 3 would be predicted as "False" and patient 2 as "True". However, if we select "True" as the positive class and we set a probability threshold of 30%, patient 2 and 3 will be predicted as "True", and only patient 1 will have a "False" prediction.

For decision forests (see Subsection 2.2.1), you can also set a **vote threshold**, i.e., a threshold based on the percentage of models in the ensemble voting for the positive class. The type of threshold (confidence, probability or vote threshold) can be configure before creating the evaluation (see Subsection 7.4.2).

Setting a threshold is specially useful when you want to minimize false positives at the cost of false negatives. By setting a probability or confidence threshold you can easily increase the minority class predictions.

Different **thresholds** produce different **confusion matrices**, hence **different metrics** for the same evaluation. These different metrics according to the threshold set can be seen in the evaluation curve views provided by BigML (see <u>Subsection 7.2.1.5</u>).

In some cases, it is likely that several thresholds yield the same matrix, above all for small test datasets. In BigML you can select any **threshold** and the **positive class** by using the options shown in the Figure 7.21. By setting different thresholds you will see the values for the confusion matrix and the metrics changing accordingly. Single points in the evaluation curves correspond to different thresholds.

Sources Datasets S	upervised - Unsuper	vised  Predictions	s ▼ Tasks			WhizzML 🔻
		oan risk data data	set   Training (80%) v	s. Loa	6	<b>5</b> - (≡) ⁵ - (€)
Loan Risk Da	ata Dataset   Training	g (80%) 🗢	Loan Rist	k Data Datase	ət   Test (20%	)
Positive class: bad		•	Probability	threshold:	8	50%
					TP FN FP	TN 👰
ACTUAL VS. PREDICTED	bad	good	ACTUAL	RECALL	F	Phi
bad	27	41	68	39.71%	0.48	0.30
good	18	114	132	86.36%	0.79	0.30
PREDICTED	45	155	200	63.03% AVG. RECALL	0.64 AVG.	0.30 F AVG. Phi
PRECISION	60.00%	73.55%	66.77% AVG. PRECISION	70.50% ACCURACY		
100% 90%						779 easure
70% 70% 70% 70% 70% 70% 70% 70% 70% 70%	<i>,</i>		60.0% Precision	39.7 Rec		0.2957 Phi coefficient
Ure Positive Rate (19)			13.6%	22. % positive i		176.5% Lift
20%	ROC AUC: 0.7716		47.9% K-S statistic	0.36 Kendall		0.4456 pearman's Rho
	30% 40% 50% 60% False Positive Rate (FPR)		100%			

Figure 7.21: Select the probability threshold and the positive class

The greater the threshold, fewer instances will be predicted for the positive class. By setting a **a threshold of 100%**, all instances are predicted as the negative class (see Figure 7.22).

	Dervised Unsupervis				WhizzML *
	Loa	an risk data datas	set   Training (80%) vs.	Loa	. (≡)⁵₊ (
Loan Risk Dat	a Dataset   Training (	(80%) 🗢	Loan Risk	Data Dataset   Test (	20%) 🕤
					<b>0</b> 0 8
Positive class: bad		•	Probability th	reshold:	6 1009
				TP FN	FP TN
ACTUAL VS. PREDICTED	bad	good	ACTUAL	RECALL	F Phi
bad	0	68	68	0.00%	
good	0	132	132	100.00%	0.80 -
PREDICTED	0	200	200	50.00% AVG. RECALL	AVG. F AVG. Phi
PRECISION	-	66.00%	AVG. PRECISION	66.00% ACCURACY	
90% 80%		<u> </u>	66.09 Accuracy		0 F-measure
%0000 (bdu)			_ Precision	0.0% Recall	– Phi coefficient
40% 40% 40%			0.0% FPR	0.0% % positive instances	294.1%
20%	ROC AUC: 0.7716		47.9% K-S statistic	0.3648 Kendall's Tau	0.4456 Spearman's Rho
0%% 10% 20% 30 Fi	% 40% 50% 60% alse Positive Rate (FPR) 0		00%		

Figure 7.22: Probability threshold of 100%

On the other hand, by setting a **0% threshold** all instances are predicted as the positive class (see Figure 7.23).

Sources Datasets	Supervised  Unsupe	ervised   Predictions	s ▼ Tasks		WhizzML 🔻
<b>₽ ₽</b>		_oan risk data data	set   Training (80%) vs.	Loa	<b>⊘</b> _~ (≡) [*] , ③
Loan Risk D	Data Dataset   Trainin	ng (80%) 💿	Loan Risk	Data Dataset   Test	(20%) 💿
					<b>0</b> 0 &
Positive class: bad		¥	Probability th	reshold:	0%
				TP F	N FP TN
ACTUAL VS. PREDICTED	bad	good	ACTUAL	RECALL	F Phi
bad	68	0	68	100.00%	0.51 -
good	<b>132</b> 200	0	200	0.00% 50.00% AVG. RECALL	
PRECISION	34.00%	-	AVG. PRECISION	34.00% ACCURACY	AVG. F AVG. Phi
100% 90% 80%					0.5075 F-measure
%00'001 (Hell) 60% 60% 60% 60% 60% 60% 60% 60%			34.0% Precision	100.0% Recall	– Phi coefficient
All			100.0% FPR	100.0% % positive instances	100.0% Lift
20%	ROC AUC: 0.771		47.9% K-S statistic	0.3648 Kendall's Tau	0.4456 Spearman's Rho
0%0% 10% 20%	30% 40% 50% 60% False Positive Rate (FPR)		100%		

Figure 7.23: Probability threshold of 0%

# 7.2.1.5 Evaluation curves

As explained in Subsection 7.2.1.4, setting different thresholds can result in different confusion matrices and different metrics. The best way to evaluate how your model performs for all the possible thresholds is to plot the metrics in different charts:

- Precision-Recall curve
- ROC curve
- Gain curve & K-S statistic
- Lift curve

Single points for each curve represent a probability threshold for the positive class selected.

# 7.2.1.5.1 Precision-Recall Curve

The Precision-Recall curve visually represents the **trade-off**⁹ between both measures for the positive class. Precision and recall are inversely related, i.e., for the same model you can increase recall using a lower threshold for the positive class, but it will usually result in a decrease in precision, and vice versa. You can find the formulas of both measures in Subsection 7.2.1.3.

A **high precision** and a **high recall** are represented by points near the upper right corner of the chart (1,1), thus the greater the area under the precision-recall curve the better. BigML provides two different area calculations:

• **PR AUC**: the Area Under the Curve (AUC) is calculated taking into account the exact curve shape (Figure 7.24).



Figure 7.24: Area Under the Curve for the Precision-Recall curve

• **PR AUCH**: the Area Under the Convex Hull is calculated taking into account the convex shape of the curve where no other points lay above the curve. You can visualize it by clicking the option shown in Figure 7.25,

⁹https://en.wikipedia.org/wiki/Precision_and_recall



Figure 7.25: Area Under the Convex Hull for the Precision-Recall curve

The **appropiate balance** between precision and recall needs to be decided in a per-case basis according to the **costs** associated wit **false positives** and **false negatives**.

# 7.2.1.5.2 ROC Curve

The ROC space graphically represents the existing **trade-off**¹⁰ between the **recall** (or sensitivity) and **specificity** for classification problems. The **recall** is obtained by calculating the **True Positive Rate** (**TPR**), i.e. the ratio of instances that has been correctly classified for the positive class. The **False Positive Rate** (**FPR**), equivalent to **1-specificity**, is the percentage of negative class instances that have been incorrectly classified.

You can obtain the TPR and FPR by normalizing the confusion matrix results:

$$\label{eq:FPR} \begin{split} \mathsf{TPR} &= \mathsf{Recall} = \frac{TP}{TP+FN} \\ \mathsf{FPR} &= \frac{FP}{TN+FP} \end{split}$$

 $^{^{10} {\}tt https://en.wikipedia.org/wiki/Sensitivity_and_specificity}$ 



Figure 7.26: The ROC curve

The diagonal of the chart divides the space; all the points found in the upper left part of the chart (where TPR>FPR) can be considered good results, and all those found in the diagonal or below (where TPR=<FPR) are bad results. The diagonal represents a model that has the same performance as choosing a class for each point at random.

Similar to the precision-recall curve, the Area Under the Curve¹¹ (AUC) for an evaluation, is the area beneath the evaluation's ROC curve in the ROC space (see Figure 7.27). Higher AUC values indicate a better classifier performance; however, in extreme cases, such as AUC=1, it may reflect an overfitting problem.

¹¹https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve



Figure 7.27: The ROC AUC

You can also visualize the Area Under the Convex Hull by clicking the highlighted option in Figure 7.30.



Figure 7.28: The ROC AUCH

# 7.2.1.5.3 Gain Curve & K-S statistic

The Gain curve (or Cumulative Gain curve) represents the relationship between the **percentage of correct predictions** for the positive class and the effort needed to achieve them measured as the **percentage of instances predicted**. The y-axis in the Gain curve is equivalent to the recall as well as the True Positive Rate (TPR) and the x-axis is the percentage of positive class instances. The formulas for these metrics are:

$$\label{eq:Gain} \begin{split} \text{Gain} = \text{Recall} = \text{TPR} = \frac{TP}{TP+FN} \\ \text{\% of Positive Instances} = \frac{TP+FP}{TP+FP+TN+FN} \end{split}$$

Similar to the ROC curve, the diagonal of the chart represents the results of a random model. All the points above the diagonal can be considered good results. The closer a point is to the upper left corner (0,1), the better.

Along with the Gain curve, BigML also provides in the same chart the **Negative Cumulative Response curve** (represented by the black curve shown in Figure 7.29). The Negative Cumulative Response curve represents the percentage of instances incorrectly predicted as positive, so it is equivalent to the False Positive Rate (FPR) as explained in the section on ROC curves (Subsection 7.2.1.5.2):

Negative Cumulative Response = 
$$FPR = \frac{FP}{TN + FP}$$

Along with the Gain and Negative Cumulative response curve, BigML provides the calculation of the **Kolomogorov Smirnov statistic (K-S statistic)**. It measures the maximum difference between the TPR and the FPR over all possible thresholds:

K-S statistic =  $\max(TPR - FPR)$ 

The K-S statistic is an indicator of how well the model separates the positive from the negative classes. A K-S statistic of 100% indicates a perfect separation and a model that classifies everything correctly. Higher values for the K-S statistic indicate a higher quality model.



Figure 7.29: The Gain curve

#### 7.2.1.5.4 Lift Curve

The Lift curve shows the goodness of fit of your **model** compared to a **random class assignment** given a sample of positive instances. The **Lift** is plotted in the y-axis and it is calculated as the ratio between the result predicted by your model and the result using no model. The x-axis represents again the **percentage of correct predictions**. The formulas for these metrics are:

$$\label{eq:Lift} \text{Lift} = \frac{Precision}{\frac{PositiveInstances}{TotalInstances}} = \frac{\frac{TP}{TP+FP}}{\frac{TP+FN}{TP+FP+TN+FN}}$$
 % of Positive Instances = 
$$\frac{TP+FP}{TP+FP+TN+FN}$$

The horizontal line in the chart indicating a 100% lift (see Figure 7.30) represents a model that makes random predictions.



Figure 7.30: The Lift curve

For a more detailed explanation of the Gain and Lift charts, please refer to this article¹².

# 7.2.2 Regression Measures

When the objective field of the model, ensemble, deepnet, or fusion is numeric the resulting evaluation includes the **regression measures** explained below.

#### 7.2.2.1 Mean Absolute Error

The **Mean Absolute Error** is the mean of the model prediction errors for each instance. It is computed as the average of the absolute values of the differences between the target variable predicted by the model (y') vs. the actual values (y). Letting N be the total number of instances evaluated, then:

Mean Absolute Error 
$$= rac{\sum_n |y'_n - y_n|}{N}$$

# 7.2.2.2 Mean Squared Error

The **Mean Squared Error** is similar to the **Mean Absolute Error**, but the differences between predictions and actual values are squared. It is computed as the average of the squares of the differences between

 $^{^{12}} https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/$ 

the target variable predicted by the model (y') vs. the actual values (y). Letting N be the total number of instances evaluated, then:

Mean Square Error 
$$=rac{\sum_n (y'_n-y_n)^2}{N}$$

#### 7.2.2.3 R Squared

The  $R^2$ , also called the coefficient of determination¹³, measures how much better the model is than always predicting the mean value of the target variable  $(\bar{y})$  in the test set. It can take values up to 1. Values below 0 indicate the model is worse than predicting the mean; a value of 0 means the model is not any better than predicting the mean; and 1 means the model perfectly fits the data. Although an  $R^2 = 1$  may not necessarily be desirable (since it can be a symptom of overfitting), higher values for  $R^2$ usually mean better performance.

$$R^{2} = 1 - \frac{\sum_{n} (y_{n}' - y_{n})^{2}}{\sum_{n} (y_{n} - \bar{y}_{n})^{2}}$$

### 7.2.3 Cross-Validation Measures

BigML cross-validation yields k different models and k evaluations. To get a single estimation of the model's performance, the results of the k evaluations are averaged to obtain the final cross-validation measures (per class and overall measures). Consequently, cross-validation evaluations have the same measures as single classification and regression evaluations. (See Subsection 7.2.1 and Subsection 7.2.2.)

Additionally, apart from the averages, you will also find the **standard deviation** for each classification and regression measure. (See Subsection 7.5.4.)

Cross-validation evaluations do not include the evaluation curves, but will in a future release.

# 7.3 Creating Evaluations

The process to create an evaluation is different if you want to create a single **evaluation** or a **cross-validation** evaluation:

- To create a single evaluation, you need two resources: a testing dataset (different from the one used for training) and a model, an ensemble, a logistic regression, deepnet, or fusion. All three processes follow a similar logic. You can find a separate explanation of each one in the following subsections.
- To create a **cross-validation** evaluation, you just need a dataset. BigML allows you to create cross-validation for **models**, **ensembles**, **logistic regressions**, **deepnets** and **fusions**. This process is explained in Subsection 7.3.6

#### 7.3.1 Model Evaluations

To evaluate a model, you can use any of the following options from the BigML Dashboard:

• Click EVALUATE A MODEL in the 1-click action menu from the evaluation list view. (Figure 7.31.)

¹³https://en.wikipedia.org/wiki/Coefficient_of_determination

			Evaluations			Q	4
ŝ	ill	Name	≎ Type	Performance	evaluate.	MODEL	
2	ıl	Evaluation of Fictional Wine Sales dataset with F	木			AN ENSEMBLE	
2	ılî	Evaluation of Fictional Wine Sales dataset with F	木		COMPARE	EVALUATIONS	010
6 <mark>9</mark>	ılî	Evaluation of Fictional Wine Sales dataset with F	木		40min	10.1 KB	318
ŝ	ılî	Evaluation of Diabetes diagnosis dataset - balanc	00		55min	5.4 KB	154
6 <mark>9</mark>	ılî	Evaluation of Diabetes diagnosis dataset - balanc	00		55min	5.4 KB	154
<b>8</b>	ılî	Evaluation of Diabetes diagnosis dataset - sample	00		56min	5.4 KB	154
¢,	1 ⁰	Evaluation of Diabetes diagnosis dataset - sample	00		57min	5.4 KB	154

Figure 7.31: Evaluate model from evaluation list view

This option redirects you to the **New Evaluation** view where you need to select a model and a testing dataset. (See Figure 7.32.) From this view you can also select an ensemble by clicking the ensemble icon above the model selector.

Sources	Datasets Evaluations -	Clusters Anomalies	Associations	Predictions	Tasks	Scripts 👻
A Il	хî ²	Ne	w Evaluation			<b>\$</b> - i
ŝ	fa					
53	Search model dia Diabetes diagnosis dataset - sample	۵ ۹ (80 00%) model		Search datase	et	T
Evaluation Evaluatio	Diabetes diagnosis dataset - sample Diabetes diagnosis dataset's model	e (80.00%)'s model			Reset	Evaluate

Figure 7.32: Select model and dataset

• Click EVALUATE in the 1-click action menu from the model view. (See Figure 7.33.)



Figure 7.33: Evaluate model from 1-click action menu

Alternatively, click EVALUATE in the pop up menu from the model list view (see Figure 7.34).

Sourc	ces Datas	sets Models -	Clusters	Anoma	lies	Associations	Prediction	s Task	s			Executio	ons 🕶
						Mode	ls						0
III	Name			\$	Туре	Objective	\$		۵	\$	\$	\$	00 ¢
ılî	Fictional V	Vine Sales dataset - sa	ample (80.00	۲	٨	Total Sales		34mi	n 40.3	KB	0	0	1
ı۱۹	Diabetes	PREDICT BY QU	ESTION			Diabetes		48mi	n 21.6	бKB			1
ılı	Diabetes	PREDICT	PREDICTION		00	Diabetes		50mi	n 21.6	KB			1
ı۱۴	Diabetes				00	Diabetes		52mi	n 27.0	KB			
ılî	Arrythmi	VIEW DETAILS			00	Arrhythmia		2r	n 536.2	KB		1	
Show	10 ᅌ mod	MOVE TO		2		1 to 5 of 5 model	S				1	< < 1	> >

Figure 7.34: Evaluate model from pop up menu

By using any of these options, you will be redirected to the **New Evaluation** view where the model will be pre-filled in the selector and you only have to choose the testing dataset. If you previously split your original dataset into two subsets (one for training and another for testing) using the 1-click menu option from your dataset view, BigML will automatically select the corresponding testing dataset. Finally, click the Evaluate green button to perform the evaluation. (See Figure 7.35.)

ources Datasets	Evaluations -	Clusters Anomalies	Associations	Predictions	Tasks	Scripts 🗸
11 ² 22 11		Ne	w Evaluation			<b>\$</b> - ()
ស៊ា 🕅						
Diabetes dia	gnosis dataset - sampl	e (80.00%)'s mo × 🔻		Diabetes dia	gnosis dataset - samp	ele (20.00%) × ×
	etes Tu	e, 03 May 2016 09:16:21 +0000	ılî			Tue, 03 May 2016 09:16:07
21.6 KB size				5.4 KB size		154 instances
Description:		$\odot$	Desc	ription:		$\overline{\mathbf{O}}$
Configure						$\odot$
valuation name:						
	s diagnosis dataset	- sample with Diabe			Reset	Evaluate

Figure 7.35: Evaluation with pre-filled model and dataset information

# 7.3.2 Ensemble Evaluations

To evaluate an ensemble you can use the following options from the BigML Dashboard:

• Click EVALUATE AN ENSEMBLE in the 1-click action menu from the evaluation list view (Figure 7.36).

Sources	Data	sets Evaluations -	Clusters	Anomalies	Asso	ociations		Predictions	Та	sks			Script	s T
					Eval	uations	;						Q,	ᠿ.
វណ	ill	Name			\$	Туре	٥	Performance	\$		EVALUATE A	MODEL		
с ⁰	ıl <mark>o</mark>	Evaluation of Diabetes of	liagnosis data	set - sample		00					EVALUATE A	N ENSEMBLE		
ŕå	ılî	Evaluation of Fictional V	Vine Sales data	aset with F		٨		—			COMPARE E	EVALUATIONS		010
ŕå	ıl <mark>e</mark>	Evaluation of Fictional V	Vine Sales data	aset with F		٨		_			1d 2h	10.1 KB		318
к ⁰	ıl <mark>e</mark>	Evaluation of Fictional V	Vine Sales data	aset with F		٨		_			1d 2h	10.1 KB		318
é 🖧	ılı.	Evaluation of Diabetes of	liagnosis data	set - balanc		o0		—			1d 2h	5.4 KB		154
5	ıl <mark>o</mark>	Evaluation of Diabetes of	liagnosis data	set - balanc		o0					1d 2h	5.4 KB		154
é 🖧	ılı	Evaluation of Diabetes of	liagnosis data	set - sample		o0					1d 2h	5.4 KB		154
к ⁰	ılı	Evaluation of Diabetes of	liagnosis data	set - sample		o0		_			1d 2h	5.4 KB		154
how 10	ᅌ evalu	lations		1 to 3	8 of 8 eva	aluations						IK	< 1 >	>1

Figure 7.36: Evaluate ensemble from evaluation list view

This option takes you to the **New Evaluation** view where you need to select an ensemble and a testing dataset. (See Figure 7.37.) From this view, you can also select a model by clicking the model icon above the ensemble selector.

Sources	Datasets Evaluations -	Clusters A	nomalies	Associations	Predictions	Tasks	Executions -
	б ⁰ .		Ne	w Evaluation			<b>5</b> - (i)
វា	Êŵ						
fa	Search ensemble		A Q		Search datas	set	×
Evaluation Evaluatio	Diabetes diagnosis dataset - sa	mple (80.00%) ense mple (80.00%)'s en	emble			Res	set Evaluate

Figure 7.37: Select ensemble and dataset

• Click EVALUATE in the 1-click action menu from the ensemble view- (Figure 7.38.)



Figure 7.38: Evaluate ensemble from 1-click action menu

Alternatively, click EVALUATE in the pop up menu from the ensembles list view (Figure 7.39).

Source	es Datasets Ensembles Clusters	An	omalies	Associations	Predict	ions	Tasks			Executio	ons 🔻
			6.6	Ensembles							0
III	Name	\$	Type 🗘	Objective	\$		\$	<b>Å</b> \$	\$	<b>\$</b>	
ılî	Fictional Wine Sales dataset - sample (80.00	۲	$\checkmark$	Total Sales		1h 4n	nin	1.8 MB	0	0	0
ılî	Fictional		⋆	Total Sales		1h 5n	nin	4.0 MB			1
ılî	Fictional		٨	Total Sales		1h 6n	nin	402.6 KB			1
ıl <mark>e</mark>	Diabetes CIEW DETAILS		o0	Diabetes		1h 8n	nin	215.7 KB			
ılî				Diabetes		1h 23n	nin	215.7 KB			1
ılî	Diabetes dragnosis dataset - sample (ou.ou	2		Diabetes		1h 24n	nin	215.7 KB			1
ılî	Diabetes diagnosis dataset's ensemble		00	Diabetes		1h 25n	nin	269.8 KB			
Show 1	0 ᅌ ensembles		1 to	7 of 7 ensembles						K < 1	> >1

Figure 7.39: Evaluate ensemble from pop up menu

By using any of these two options, you will be redirected to the **New Evaluation** view where the ensemble will be pre-filled and you only have to choose the testing dataset. If you previously split

your original dataset into two subsets (one for training and another for testing) using the 1-click menu option from your dataset view, BigML will automatically select the corresponding testing dataset. Finally, click the Evaluate green button to perform the evaluation. (See Figure 7.40.)

1 [°] s [°]			No.	v Evaluation		<b>3</b> - (
- III - P2				VEVAluation		
សា ៩ស						
Diabete	s diagnosis datas	et - sample (80.00°	%)'s ens × 🔻		etes diagnosis datase	t - sample (20.00%) × ×
£\$ ₀01 ⊕ ¤	liabetes	Tue, C	3 May 2016 09:32:46	ıl		Tue, 03 May 2016 09:16:07
215.7 KB size	10 fields	614 instances	10 models	5.4 KB size	10 fields	154 instances
Description:			◙	Description		$\overline{\mathbf{O}}$
Configure						$\odot$
valuation name:						
Evaluation of Diah	etes diagnosis	dataset - sampl	e with Diabe			Reset Evaluate

Figure 7.40: Evaluation with pre-filled ensemble and dataset information

# 7.3.3 Logistic Regression Evaluations

To evaluate a logistic regression, you can use these options from the BigML Dashboard:

• Click EVALUATE A LOGISTIC REGRESSION from the **1-click action menu** from the evaluation list view. (See Figure 7.41.)

Source	es Data	sets Evaluations -	Clusters	Anomalies	Asso	ociations	Predictions	Т	Fasks	Scrip	ots 👻
					Eval	uations				Q,	₲-
ŝ	ıll	Name			\$	Туре 🗧	Performance	٥	EVALUATE A MODEL		
×	ılî	Evaluation of Loan risk	data dataset  1	Training wi		00			EVALUATE AN ENSEMBLE		
	_								EVALUATE A LOGISTIC RE		
Show 1	0 ᅌ evalı	lations		1 to 1	l of 1 ev	aluations			COMPARE EVALUATIONS		

Figure 7.41: Evaluate logistic regression from evaluation list view

This option redirects you to the **New Evaluation** view where you need to select a logistic regression and a testing dataset. (See Figure 7.42.)

Sources Datasets	Evaluations - Clus	sters Anomalies	Associations	Predictions	Tasks	Scripts -
a   ıî   ≾?		Ne	w Evaluation			<b>5</b> - i
<u>කි</u> එක .	X					
US car acci US car acci I all $\bigoplus$ Injur 741.0 KB	lents dataset's logistic reg y Severity Wed, 25 16 fields	ression x x	,	Search datas	et more character	۸ م
Description:		َ⊘				
Evaluation name:						
Evaluation of US car	accidents dataset's logi	stic with dataset			Reset	Evaluate

Figure 7.42: Select logistic regression and dataset

• Click EVALUATE from the logistic regression 1-click action menu. (Figure 7.43.)



Figure 7.43: Evaluate logistic regression from 1-click action menu

Alternatively, click EVALUATE in the **pop up menu** from the logistic regression list view. (Figure 7.44.)

Sources	Datasets	Logistic Regressions 🔻	Clusters Ar	nomalies Associ	ations Prediction	s Tasks		Scri	pts 🗸
			Log	gistic Regressio	ns				Q
ill	Name		\$	Objective	≎ 🛗 ≎	₿ ≎	00 \$	\$	\$
ılî	US car accidents da	ataset's logistic regression	8	Injury Severity	1h 28min	741.0 KB	0	1	0
ılı	Arrythmia diagnosi	_		Arrhythmia	1h 29min	536.2 KB			
ılî	Loan risk data data	CREATE BATCH PREDIC	CTION	class	1h 31min	135.8 KB			
ılî	Loan risk data data			class	22h 53min	135.8 KB		2	
ılî	Arrythmia diagnosi		RESSION	Arrhythmia	1d 1h	536.2 KB			
ılî	Loan risk data data			class	1d 3h	135.8 KB		2	

Figure 7.44: Evaluate logistic regression from pop up menu

By using any of these options, you will be redirected to the **New Evaluation** view where the logistic regression will be pre-filled in the selector and you only need to choose the testing dataset. If you previously split your original dataset into two subsets (one for training and another for testing) using the 1-click menu option from your dataset view, BigML will automatically select the corresponding testing dataset. Finally, click the Evaluate green button to perform the evaluation. (See Figure 7.45.)

Sources Datasets	Evaluations -	Clusters Anomalies	Associations	Predictions	Tasks	Scripts 🗸
11 ² 52		Ne	w Evaluation			ۍ (
r 👬						
Loan risk dat	a dataset   Training (80	%)'s logistic reg × v		Loan risk dat	a dataset   Test (20%)	× •
	We	d, 25 May 2016 15:04:04 +000				Wed, 25 May 2016 15:03:55
108.6 KB size	21 fields	800 instances	2	27.2 KB size	21 fields	200 instances
Description:		$\overline{\mathbf{O}}$	Desc	ription:		$\overline{\mathbf{O}}$
State Configure						◙
Evaluation name:						
Evaluation of Loan risk	k data dataset   Trai	ning with Loan risk (			Reset	Evaluate

Figure 7.45: Evaluation with pre-filled logistic regression and dataset information

# 7.3.4 Deepnet Evaluations

To evaluate a deepnet, you can use these options from the BigML Dashboard:

• Click EVALUATE A DEEPNET from the **1-click action menu** from the evaluation list view. (See Figure 7.46.)

			valuations			Ô	۹ 🖒
Ω	ıll	Name	≎ Type ≎	Performance	EVALU	JATE A MODEL	
<u></u>	ıľ	Happiness score 2015-2016 vs. Happiness 2017 beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	入	_	EVALU	JATE AN ENSEMBLE	
-0	-0	Fictional Wine Sales   Training (80%) vs. Fictional Win	1		EVALU	JATE A LOGISTIC REGR	RESSION
< ⁸	ılî	beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	<u>^</u>	_	EVALU	JATE A TIME SERIES	
< <mark>.</mark>	ılî	Churn in the telecom industry   Training (80%) vs. Chu beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	00			JATE A DEEPNET	
<u></u>	ılî	Churn in the telecom industry   Test (20%) vs. Churn i beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	00	_		ARE EVALUATIONS	IATIONS
< <mark>S</mark>	ılî	Happiness score 2015-2016 vs. Happiness 2017 max training time=1800, missing values	木	_		14h 28.8 K	
<b>8</b>	ılî	Happiness score 2015-2016 vs. Happiness 2017 512-node, 10-model bootstrap decision forest, plurality	木		5d 2	20h 28.8 K	(B 15
2	ılî	Happiness score 2015-2016 vs. Happiness 2017 512-node, 10-model bootstrap decision forest, plurality	木	•	5d 2	21h 28.8 K	(B 155
< <mark>e</mark>	ılî	FMEL_Dataset.csv [extended]   Training (80%) vs. FM max training time=1800, missing values	00	-	) 1w	5d 564.4 K	(B 7K-
Ľ	ılî	FMEL_Dataset.csv [extended]   Training (80%) vs. FM beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	00	-	) 1w	5d 564.4 K	(B 7K-
< ^Q	ılî	FMEL_Dataset.csv [extended]   Training (80%) vs. FM beta1=0.9, beta2=0.999, epsilon=1e-08, learning rate=0	00		• 1w	5d 564.4 K	(B 7K-

Figure 7.46: Evaluate deepnet from evaluation list view

This option redirects you to the **New Evaluation** view where you need to select a deepnet and a testing dataset. (See Figure 7.47.)

1 T	New Evaluation	<b>3</b> - (
£1 F£1 ∠ ≤	ha	
Diabetes diagnosis   Training	g (80%) × ×	×
Diabetes	Fri, 22 Sep 2017 12:20:11	
21.6 KB 9 size fields	614 instances	
size fields	instances	

Figure 7.47: Select deepnet and dataset

• Click EVALUATE from the deepnet 1-click action menu. (See Figure 7.48.)



Figure 7.48: Evaluate deepnet from 1-click action menu

Alternatively, click EVALUATE in the pop up menu from the deepnet list view. (Figure 7.49.)

			×	Deepnets				Ô	Q
11	Name	٥	Туре	Objective	⇒ 🛱	<b>ê</b> \$	\$	\$	08
I <mark>R</mark>	FMEL_Dataset.csv [extended]   Training (80%) None hidden layers, default numeric value=zero		o0	Result	6d 4h	2.2 MB			
I <mark>R</mark>	FMEL_Dataset.csv [extended]   Training (80%) max training time=1800, missing values		o0	Result	1w 5d	2.2 MB	1		1
I <mark>P</mark>	FMEL_Dataset.csv [extended]   Training (80%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin	8	00	Result	1w 5d	2.2 MB	0	0	1
I <mark>R</mark>	FMEL_Datas predict		o0	Result	1w 5d	2.2 MB			1
l <mark>î</mark>	Forest Cover         CREATE BATCH PREDICTION           max training t         EVALUATE		o0	Cover_Type	1w 5d	1.5 MB			1
I <mark>P</mark>	Forest Cover beta1=0.9, be		00	Cover_Type	1w 5d	1.5 MB			1
I ^Q	Churn in the DELETE DEEPNET	,	00	Churn	1 w 5d	218.7 KB	1		
I ^Q	Churn in the MOVE TO initial accumulator value=0.1, learning rate powe	)	o0 <b>1</b>	Churn	1w 5d	218.7 KB			1
I <mark>R</mark>	Churn in the telecom industry   Test (20%) beta1=0.9, beta2=0.999, epsilon=1e-08, learnin		o0	Churn	1 w 5d	54.7 KB			
R	Churn in the telecom industry   Training (80%) initial accumulator value=0.1, learning rate powe		00	Churn	1w 5d	218.7 KB			1

Figure 7.49: Evaluate deepnet from pop up menu

By using any of these options, you will be redirected to the **New Evaluation** view where the deepnet will be pre-filled in the selector and you only need to choose the testing dataset. If you previously split your original dataset into two subsets (one for training and another for testing) using the 1-click menu option from your dataset view, BigML will automatically select the corresponding testing dataset. Finally, click the Evaluate green button to perform the evaluation. (See Figure 7.50.)

Sources Datasets Super	ised ▼ Unsupervised ▼ Predictions ▼ Tasks	WhizzML 🔻
9 TT	New Evaluation	<b>3</b> - 3
කි 🗚 🔟	*	
Churn in the telecom	ndustry   Training (80%) × v	n in the telecom industry   Test (20%) × ×
	Mon, 25 Sep 2017 14:03:49	Tue, 12 Sep 2017 18:31:22
218.7 KB size	19 2,666 54.7 KB fields instances size	20 667 fields instances
Description:	ODescription:	$\odot$
ុំដ្ឋ Configure		$\odot$
Evaluation name:		
	Training (80%) vs. Churn in the t	Reset

Figure 7.50: Evaluation with pre-filled deepnet and dataset information

# 7.3.5 Fusion Evaluations

To evaluate a fusion, you can use these options from the BigML Dashboard:

• Click EVALUATE A FUSION from the **1-click action menu** from the evaluation list view. (See Figure 7.51.)

Source	es D	atasets Supervised - Unsupervised -	Predictions -	Tas	ks			WhizzN	ЛL <del>-</del>
			Evaluation	ıs			節	Q,	3
ណ	ill	Name	†ј Тур	e îţ	Performance	EVALUATE	A MODEL		
23	ılî	Iris dataset - extended vs. Iris dataset v2	n	1			AN ENSEMBLE		
		operating kind=probability		-		EVALUATE	A LOGISTIC REGRE	SSION	
<u>R</u>	ılî	Stumble_upon.csv vs. Stumble_upon.csv operating kind=probability	٥			EVALUATE	A DEEPNET		
20	ılî	Wine Sales vs. Fictional Wine Sales			_	EVALUATE			
<i>S</i> .		operating kind=probability				EVALUATE	A TIME SERIES		
8	ılî	Titanic Survival vs. Titanic Survival operating kind=probability	٥		-		EVALUATIONS		
3	ılî	Stumble_upon.csv vs. Stumble_upon.csv operating kind=probability	٥			COMPARE I	MULTIPLE EVALUA	TIONS	
Kî ^Q	ılî	Diabetes diagnosis 512-node, pruned, operating kind=probability	٥			▶ 5m 2w	27.0 KB		768
/*	ılî	Loan risk evaluation operating kind=probability	٥		_	5m 3w	716.0 KB		4K-
2	ılî	Fictional Wine Sales vs. Fictional Wine Sales 512-node, 10-model bootstrap decision forest, oper				5m 3w	50.4 KB		1K-
8	ılî	Fictional Wine Sales vs. Fictional Wine Sales 512-node, 10-model bootstrap decision forest, oper	/			5m 3w	50.4 KB		1K+
2	ılî	Diabetes diagnosis dataset   Training (70%) v4 vs 512-node, 31-model bootstrap decision forest, thres	-1			5m 3w	8.1 KB		231
iow 1	0 ᅌ en	valuations 1	1 to 10 of 461 evaluation	ns			K < 1 2 3	4 5	××

Figure 7.51: Evaluate fusion from evaluation list view

This option redirects you to the **New Evaluation** view where you need to select a fusion and a testing dataset. (See Figure 7.52.)

Sources	Datasets Supervised - Unsupervised - Pr	redictions - Tasks		WhizzML 🚽
A 17	oc N	ew Evaluation		<b>5</b> . (i)
វវ	fa 1 z z 2 🗠			
	earch fusion	Search dataset	Ŧ	
	tit Q	Titanic Survival		
	Titanic Survival 35 total models (deepnet: 1, ensemble: 5, logisticregressio	35 total models (deepnet: 1, ensemble: 5, logisticregression: 29)		
Evaluation fusion vs	Titanic		Reset	
	Titanic Survival 1 total model (logisticregression: 1)			
	Titanic Survival v1 2 total models (logisticregression: 2)			

Figure 7.52: Select fusion and dataset

• Click EVALUATE from the fusion 1-click action menu. (See Figure 7.53.)
		Titanic Survival	📋 🗘	
TOTAL MODELS	OBJECTIVE FIELD		PREDICT	
35	SURVIVED		BATCH PREDICTION	
			EVALUATE	
			CREATE FUSION USING THIS ONE	C
			DELETE FUSION	
Y Fare today 123	· -	£}	MOVE TO	

Figure 7.53: Evaluate fusion from 1-click action menu

Alternatively, click EVALUATE in the pop up menu from the fusion list view. (Figure 7.54.)

Sources Datasets	Supervised - Unsupervised -	Pred	ictions 🕶 Tasks			Whizz	ML 🗸
			Fusions		Ø	Î <	<b>\$</b> -
Name	t _i	Туре	Objective	t,	*	<b>1</b>	
Titanic Survival 35 total models (deepnet: 1, ens	semble: 5, logisticregression: 🖉 😣	o0	Survived	22h 19min			
Diabetes diagnosiscxc 66 total models (ensemble	-	o0	Diabetes	1d 19h			
Arrythmia diagnosis [filte 2 total models (logisticregr		00	Arrhythmia	2d 20h			
Arrythmia diagnosis [filte 2 total models (logisticregr		00	Arrhythmia	2d 20h		1	
Arrythmia diagnosis [filte 1 total model (logisticregre		00	Arrhythmia	2d 20h			
Iris dataset - extended 1 total model (logisticregre		>	Species	2d 20h			1
Arrythmia diagnosis dataset's 1 total model (logisticregression		o0	Arrhythmia	2d 20h			
Diabetes iosjd 38 total models (ensemble: 38)		o0	Diabetes	2d 21h			
Iris_missing 4 total models (deepnet: 1, ense	mble: 1, logisticregression: 1	o0	Species	2d 22h			
Titanic 1 total model (logisticregression	E 1)	o0	Survived	3d 1h			
Show 10 ᅌ fusions		1 to 10	of 59 fusions		I< < 1	2 3 4 5	××

Figure 7.54: Evaluate fusion from pop up menu

By using any of these options, you will be redirected to the **New Evaluation** view where the fusion will be pre-filled in the selector and you only need to choose the testing dataset. Finally, click the **Evaluate** green button to perform the evaluation. (See Figure 7.55.)

Sources Datasets Supervised	Unsupervised - Predictions - Tasks	WhizzML 🔻
e 🐨	New Evaluation	<b>5</b> - ()
n fn ∠ ≤ 0		
Titanic Survival	× •	Survival dataset × v
Survived	Wed, 27 Jun 2018 18:18:21	Fri, 26 Feb 2016 12:38:05
35 models	75.3 KB size	5 2,208 fields instances
Description:	• Description:	$\odot$
Configure		$\overline{\mathbf{O}}$
Evaluation name:		
Titanic Survival vs. Titanic Survival dataset	t	Reset

Figure 7.55: Evaluate a fusion using a testing dataset

### 7.3.6 Cross-Validation Evaluations

In BigML, you can use k-fold cross-validation¹⁴ to evaluate your models, ensembles, logistic regressions, and deepnets. Cross-validation evaluations are implementing in BigML as a WhizzML script and they can be found in BigML Gallery:

- 1. Go to the scripts Gallery¹⁵ where you will find five different scripts to perform cross-validation:
  - **Basic 5-fold cross-validation**¹⁶: performs a 5-fold cross-validation for models with default model configuration options. Learn the default options for models in Subsection 7.4.5.2. (See Figure 7.56.)

¹⁴https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation ¹⁵https://bigml.com/gallery/scripts

¹⁶ https://bigml.com/dashboard/script/57989125af447f2234000003



Figure 7.56: Basic 5-fold cross-validation

• **Model's k-fold cross-validation**¹⁷: performs cross-validation for models. You can configure the *k*-fold parameter and the model inputs. Learn about the configurable inputs for models in Subsection 7.4.5.2. (See Figure 7.57.)

Model's k-fold of whizzml	cross-validation
k-fold cross-validation for	r models
Evaluations Models	Cross-validation
13.6 KB	
<b>Ģ</b> 0 <b>k</b> 11	Ef8in⊠

Figure 7.57: Model's k-fold cross-validation

• Ensemble's k-fold cross-validation¹⁸: performs cross-validation for ensembles. You can configure the *k*-fold parameter and the ensemble inputs. Learn about the configurable inputs

¹⁷https://bigml.com/dashboard/script/579a2973af447f2234000602 ¹⁸https://bigml.com/dashboard/script/57989115af447f2234000000

Ensemble's k-fold cross-validation
k-fold cross-validation for ensembles
Evaluations Ensembles Cross-validation
16.6 KB
🖵 o 🖡 7 🛛 E f 8 in 🖂

#### for ensembles in Subsection 7.4.5.3. (See Figure 7.58.)

Figure 7.58: Ensemble's k-fold cross-validation

• Logistic regression's k-fold cross-validation¹⁹: performs cross-validation for logistic regression. You can configure the *k*-fold parameter and the logistic inputs. Learn about the configurable inputs for logistic regressions in Section 4.4. (See Figure 7.59.)



Figure 7.59: Logistic regression's k-fold cross-validation

¹⁹https://bigml.com/dashboard/script/5aba720eeba31d2795001d6f

• Deepnet's k-fold cross-validation²⁰: performs cross-validation for deepnets. You can configure the k-fold parameter and the deepnet inputs. Learn about the configurable inputs for deepnets in Section 5.4. (See Figure 7.60.)



Figure 7.60: Deepnet's k-fold cross-validation

2. By clicking the script preview you can inspect script's details such as the source-code, the script input and the output. (See Figure 7.57.) You can find additional documentation about WhizzML scripts here²¹.

²⁰ https://bigml.com/user/whizzml/gallery/script/5a9ed849eba31d3b480005b1

²¹https://bigml.com/whizzml

☐ □ □ ■ □	(=) Model's k-fold cross	-validation
2 3 (define MODEL_OP1 4 5 6 7 8	<pre>l eventually be defined as a library. fIONS ["balance_objective"     "missing_splits"     "pruning"     "weight_field"     "objective_weights"     "node_threshold"]) OPTIONS (concat MoBL_OPTIONS     ["sample_rate"     "replacement"     "replacement"     "sample_rate"     "replacement"     "mediacement"     "sample_rate"     "replacement"     "sample_rate"     "replacement"     "sample_rate"     "replacement"     "sample_rate"     "sample_rate"     "sample_rate"     "sample_rate"     "replacement"     "sample_rate"     "samp</pre>	0
k-fold cross-validation for mode	eis	
Inputs		۲
dataset-id	alt	Select the dataset for training/test the model
k-folds	123	Select the number of folds to split the dataset
objective-id	text	Objective field ID, or empty for default dataset objective field
missing-splits	boolean	Sets the model's missing_splits flag
stat-pruning	boolean	Sets statistical pruning flag
Outputs		0
cross-validation-output		Average of evaluations results
	Lic	cense: Creative Commons Zero V1.0

Figure 7.61: Cross-validation script view

3. **Clone** your preferred script for FREE. You can clone it from the script preview by clicking the FREE button. (See Figure 7.62.)



Figure 7.62: Clone script from preview

Alternatively, you can clone it from the script view by clicking the FREE or Buy buttons. (See Figure 7.63.)

☐ □ □ ▶ 11	(=) Model's k-fo	old cross-validation	(	
Source code				۲
2 3 (define MODEL_OPTIONS 4 5 6 7 8 9 (define ENSEMBLE_OPTIC 10	<pre>tually be defined as a library. ["balance_objective"    "missing_splits"    "pruning"    "weight_field"    "objective_weights"    "node_threshold"]) NS (concat MODEL_OPTIONS    ["sample_rate"         "renlocement"    "renl</pre>			0
k-fold cross-validation for models				
Inputs				$\overline{\mathbf{O}}$
Outputs				۲
cross-validation-output	MC CR		Average of evaluations resu	ults
		License: Creative Commons Ze	ero V1.0	

Figure 7.63: Clone script from script view

A modal window will appear asking you for confirmation. (See Figure 7.64.)

Clone th	nis script
	You are about to clone this script
	Cancel Clone

Figure 7.64: Confirmation message to clone script

4. Once you clone the script, you will be redirected to the Execution view to set your inputs. You need to select a dataset and optionally, you can configure the rest of the inputs. If you do not configure them, they will take the default values. You can find an explanation of all your inputs in Subsection 7.4.5. (See Figure 7.65.)



Figure 7.65: Configure cross-validation inputs

5. Once you have selected the dataset, click Execute . (See Figure 7.66.)

(= <mark>;}</mark>	•	(=) Model's k-fold cros	s-validation	Q (5.	(≡)5,
		()			
Source code					∍
k-fold cross-validation for mo	dels				
Inputs – Set up them to	start an execution				
dataset-id	ll	Diabetis diagnosis dataset	v	Select the dataset for training/test model	the
k-folds	123	5		Select the number of folds to split dataset	the
objective-id	text			Objective field ID, or empty for defa dataset objective field	ault
missing-splits	boolean	False	Ψ	Sets the model's missing_splits fla	ig
stat-pruning	boolean	False	v	Sets statistical pruning flag	
Outputs		ſ			⊽
New Execution name:				_	
Model's k-fold cross-va	lideticule evenution			Reset	Execute

Figure 7.66: Execute cross-validation script

6. Once you execute the script, you can check the progress of your script in the execution view where you will find the **elapsed time**, the **total resources** generated and the script **log messages**. (See Figure 7.67.)

Sources Datasets	Models - Clusters	Anomalies Associations	Predictions Tasks	Executions *
<b>a</b> ( <u></u> ≡) [•] ••••		Model's k-fold cross-	validation's exe	(≡) ⁵ . (Ξ)
- Of	ELAPSE 0 00 DWYS HOURS	00 20	L RESOURCES	LOG MESSAGES
Source code				$\overline{\mathbf{O}}$
k-fold cross-validation for n	nodels			
Inputs				$\overline{\mathbf{O}}$
Outputs				
cross-validation-outpu	t		Average of evaluations	results
Console				$\overline{f O}$
Resources				TOTAL: 0

Figure 7.67: Cross-validation execution progress

7. Finally, cross-validation yields *k* different models and *k* different evaluations. The results of the single evaluations are averaged to obtain the final model performance measures. Access the final **cross-validation** evaluation containing the averaged measures by clicking on the evaluation ID

link in the **Outputs** section. Read more about cross-validation measures in Subsection 7.2.3. The k intermediary resources can be found in the same view under the **Resources** panel. (See Figure 7.68.)

a (= ⁹ ) ••••	(III) 1-click Cr	oss-Validation (models)'s ex	∽ (≡)5-
	ELAPSED TIME	TOTAL RESOURCES	LOG MESSAGES
	0 00 00 24 DAYS HOURS MINS SECS	15	0 0 0
Source code			$\overline{\mathbf{O}}$
1-click Cross-Validation (models)			
inute			
•			•
nputs Nutputs			
•	evaluation/573c20f	7 <b>b95b395724000039</b> 🔍 Cross Validatio	
utputs cross-validation-output	evaluation/573c20f	7b95b395724000039 Q Cross Validatio	
utputs	evaluation/573c20f	7b95b395724000039 Q Cross Validatio	on result

Figure 7.68: Cross-validation output and resources

You can perform again any new cross-validation by clicking on your cloned script listed under the scripts tab in the BigML Dashboard. (See Figure 7.69.)

Sources	Datasets	Models 🔻	Clusters	Anomalies	Associations	Predictions	Tasks	_	Scripts -
					(=) Script	S			Q_ (≡) [*] .
Type 💠	Name						\$ 🛗 🔻	<b>Å</b> \$	(II) \$
(≡ <mark>;</mark>	Model's k-fold o	ross-validation	1				14min	13.6 KB	1
Show 10	scripts				1 to 1 of 1 scripts			K	

Figure 7.69: Cross-validation script in scripts list view

# 7.4 Evaluation Configuration Options

BigML allows you to configure different options for single evaluations and for cross-validation evaluations. For cross-validation, the configuration options are called inputs.

In the following subsections you will find an explanation of each single **evaluation** configuration options. The last subsection (Subsection 7.4.5) details **cross-validation** inputs.

For single **evaluations**, all the options are found under the **configuration panel** that appears once you select the model and the testing dataset as shown in Figure 7.70.

ត៌ 🕅				
Diabetes diagnosis dataset - sample (80.00%)'s n	mo × ×	Diabetes diag	gnosis dataset - 20.00% s	sample × v
Diabetes Tue, 03 May 2016	09:16:21 +0000	ıl		Tue, 03 May 2016 09:16:0
	14 ances	5.4 KB size	10 fields	154 instances
Description:	$\odot$	Description:		⊽
Configure				C

Figure 7.70: Evaluation configuration panel

Note: some configuration options may change depending on whether you are evaluating a model, an ensemble, a logistic regression, a deepnet, or a fusion.

### 7.4.1 Missing Strategies

This option is only available for models, ensembles, and fusions that contain models and/or ensembles; for logistic regressions and deepnets the option to include missing values is configured when training the model.

When the testing dataset contains instances with missing values for models and ensembles, BigML can take two different approaches to return a prediction for those instances:

• Last prediction: when a missing value is found in the testing data for a decision node, the prediction returned will be the one from the parent of the missing split. (See Figure 7.71.)



Figure 7.71: Last prediction icon

• **Proportional missing strategy**: when a missing value is found in the testing data for a decision node, the prediction is calculated based on all subtrees predictions, taking into account the proportion of data in each subtree. (See Figure 7.72.)



Figure 7.72: Proportional missing strategy icon

BigML uses **Last prediction** as the default strategy. Select your preferred option by clicking the corresponding icon shown in Figure 7.73.

	Sources Datasets Evaluations Clusters Anomalies Associations Predictions Tasks S	icripts 🔻
	A If A New Evaluation C	• i
	53 Fa	
	Diabetes' dataset   Training (80%) ensemble × *	K Y
	Image: Wed, 21 Oct 2015 18:29:13         Wed, 21 Oct 2015 18:29:13	18:22:46
	927.5 KB 9 614 43 5.4 KB 10 154 instances models size fields instances	
	Description:            O         Description:	$\odot$
	ିୟୁ Configure	۲
Missing strategies	- & & 3	
_	Make sure that the fields in the Ensemble and Dataset match	
	Fields mapping: Default fields	◙
	2 [OPTIONAL] Select a sample	
	Sampling: 154 instances	◙
	Advanced sampling: Default settings	$\odot$
	Advanced ordering: Default settings	$\overline{\mathbf{O}}$
	Evaluation name:	
	Evaluation of Diabetes' dataset   Training (8 with Diabetes' da Reset Evaluat	e

Figure 7.73: Missing strategies for evaluations

Note: if the model or ensemble has been trained with Missing splits (learn more about this parameter in Ensembles chapter in Subsection 1.4.4), the Missing strategy may not have any impact since missing values would already be considered by the model or the ensemble as valid values.

### 7.4.2 Select the Voting Strategy for Decision Forests

For Decision Forests ensembles, you can select the voting strategy you want to calculate the evaluation. See a complete explanation about the three options that BigML provides in Subsection 2.6.3.2. Selecting a given voting strategy will also determine the threshold to calculate the curves (see Subsection 7.2.1.4).

- **Probability**: For **classification** ensembles per-class probabilities are averaged taking into account all trees composing the ensemble. The class with higher probability is the winner class. For **regression** ensemble the probability option averages the predictions of the trees composing the ensemble.
- **Confidence**: For **classification** ensembles per-class confidences are averaged taking into account all trees composing the ensemble. The class with higher confidence is the winner class. For **regression** ensembles the confidence option averages the predictions of the trees composing the ensemble
- Votes: For classification ensembles each tree prediction is considered as one vote. The "votes" of a given class is the percentage of trees in the ensemble that vote for that class. For regression ensembles the votes option averages the predictions of the trees composing the ensemble. It gives the same results as the probability

urces Datasets Supervise	ed 👻 Unsupervised 👻 Predie	ctions 🛨 Tasks	v	VhizzML 👻
1.1	New	Evaluation		<b>\$</b> - (
£1 ₽£1 ₹	<u>~</u>			
Diabetes diagnosis datase	t   Training (70%) v4 × ×	Diabetis diagr	nosis dataset   Test (30%)	* *
Diabetes	Wed, 13 Dec 2017 17:58:27	ıl	Wed, 08 Jun 20	016 15:39:39
18.9 KB 9 size fields	537 31 instances models	8.1 KB size	10 231 fields instance	es
Description:	$\odot$	Description:		◙
Con Uses all classes probabilities predict				۲
Excluded fields:				◙
1 Make sure that the fields in the	e Ensemble and Dataset match			
Fields mapping:			Default fields	$\odot$
2 [OPTIONAL] Select a sample				
Sampling:			231 instances	$\odot$
Dataset advanced sampling:			Default settings	$\overline{\mathbf{O}}$

Figure 7.74: Select the voting strategy to calculate the evaluation

Note: for single decision tree evaluations you can also choose between probabilities and confidences to calculate the thresholds of the evaluation curves.

# 7.4.3 Fields Mapping

You can specify which fields in the model, ensemble, logistic regression, deepnet, or fusion match with the fields in the testing dataset. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** by clicking on the green switcher shown in Figure 7.75. You can also **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the evaluation.



Figure 7.75: Fields Mapping for evaluations

Note: the fields mapping from the BigML Dashboard has a limit of 200 fields; for evaluations with a higher number of fields, you can use from the BigML API²² if you need to map your fields.

# 7.4.4 Sampling Your Dataset

Sometimes you do not need all the data contained in your testing dataset to generate your evaluations. If you have a very large dataset, sampling may be a good way of getting faster results. BigML allows you to take a sample before creating an evaluation so you do not need to create a different dataset. You can configure the sampling options detailed in the following subsections. (See Figure 7.76.)

### 7.4.4.1 Rate

The **Rate** is the proportion of instances to include in your sample. You can set any value between 0% and 100%. Defaults to 100%.

### 7.4.4.2 Range

Specifies a subset of instances from which to sample, e.g., choose from instance 1 until 200. The **Rate** you set will be computed over the **Range** configured. This option may be useful when you have temporal data, and you want to train your model with historical data, and test it with the most recent one to check if it can predict based on time.

### 7.4.4.3 Sampling

By default, BigML selects your instances for the sample by using a random number generator, which means two samples from the same dataset will likely be different even when using the same rates and row ranges. If you choose deterministic sampling, the random-number generator will always use the same seed, thus producing repeatable results. This lets you work with identical samples from the same dataset.

### 7.4.4.4 Replacement

Sampling with replacement allows a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once. By default, BigML generates

²²https://bigml.com/api/evaluations#ev_evaluation_arguments

samples without replacement.

### 7.4.4.5 Out of bag

This argument will create a sample containing only out-of-bag instances for the currently defined rate. If an instance is not selected as part of a sample, it is considered out of bag. Thus, the final total percentage of instances for your sample will be 100% minus the rate configured for your sample (when replacement is false). This can be useful for splitting a dataset into training and testing subsets. It is only electable when a sample rate is less than 100%.

		Evaluation	∽-
	n fn		
	Diabetes' dataset   Training (80%) ensemble × ×	Diabetes' dataset   Test (20%)	×
	C Diabetes Wed, 21 Oct 2015 18:29:13	Wed, 2	1 Oct 2015 18:22
	927.5 KB 9 614 43 size fields instances models	5.4 KB 10 size fields	
	Description:	Description:	۲
	Configure		
	8 8 3 1 do		
	Make sure that the fields in the Ensemble and Dataset match		
	Fields mapping:	Default maj	
	2 [OPTIONAL] Select a sample		
	Sampling:	154 instance	s (
Rate -	Rate: 100%	SAMPLING 100%	
Rate - Range	100%		
	Advanced sampling: Range: 154 instances 1 RANGE	100% Custom settir SAMPLING REPLACEMENT OUT OF L	ıgs @
	Advanced sampling: Range: 154 instances	100% Custom settir	ıgs 🤅



# 7.4.5 Cross-Validation Configuration Options

Depending on the cross-validation script, you will be able to configure different inputs. As it was mentioned in Subsection 7.3.6, you can find three different types of cross-validation scripts in the BigML Gallery. You can find a separate explanation of each one in the following subsections.

### 7.4.5.1 Basic 5-fold cross-validation

Basic 5-fold cross-validation script only has one configurable input: the **dataset-id**. (See Figure 7.77.) You can click on the selector and search the dataset by name. Once you have selected the dataset you can execute the script and it automatically splits your dataset into k = 5 subsets creating five different models and five evaluations using default inputs. Next Subsection 7.4.5.2 explains the default inputs for models.

ources Datasets Models - Clus	ters Anomalies Associations Predictions	Tasks Scripts -
(≡;; 60, 0000	Easic 5-fold cross-validation	<b>↔</b> (=) ⁵ - (
Source code		$\overline{\mathbf{O}}$
5-fold cross-validation of a model created with c	lefault parameters	
Inputs – Set up them to start an executi	on <b>A 1 input</b> with undefined value	
dataset-id	A Select a dataset	Select the dataset to train/test the model
Outputs		•
New Execution name:		
Basic 5-fold cross-validation's executi	on	Reset Execute

Figure 7.77: Select a dataset to execute a basic 5-fold cross-validation

#### 7.4.5.2 Model's k-fold cross-validation

You can configure the following inputs for your model's k-fold cross-validation (see Figure 7.78):

- dataset-id: select an existing dataset by clicking on the selector and searching the dataset by name.
- **k-folds**: set the number of partitions for your dataset. Your dataset will be split into *k* subsets creating *k* different models and *k* different evaluations. By default *k*=5, a *k*=10 is also commonly used.
- **objective-id**: set the ID of the field you want to be the <u>objective field</u>. You can find the field ID in your dataset view by mousing over the field type. It can be a categorical or numeric field. If no ID is given, BigML takes the dataset default objective field. (See Subsection 1.4.1.)
- missing-splits: set this input to true to consider missing values as valid values in your model. By default, it is set to false. (See Subsection 1.4.4.)
- **stat-pruning**: apply statistical pruning to all the tree nodes in order to avoid overfitting. By default, this option is disabled and the smart pruning is enabled. (See Subsection 1.4.3.)
- **balance-objective**: enable this option to let BigML automatically balance the classes of the objective field. This is only available for classification models. By default this option is disabled. (See Subsection 1.4.6.1.)
- **weight-field**: weigh the instances considering the values of one field. You need to input the field ID. The selected field must be numerical and it must not contain missing values. This is valid for both regression and classification models. (See Subsection 1.4.6.3.)
- objective-weights: set a specific weight for each class of the objective field. You need to list the weights in the same order that classes are found in your dataset histogram, e.g., to weight red, blue and yellow classes you need to input [2,3,1]. If a class is not listed, it is assumed to have a weight of 1. Weights of 0 are also valid. This option is only available for classification models. (See Subsection 1.4.6.2.)
- **node-threshold**: set a threshold for the nodes so the model stops growing. You can set a value between 3 and 2,000. By default it is set to -1 which indicates that no threshold applies. This parameter is also useful to avoid overfitting. (See Subsection 1.4.5.)

(≡;		(=) Model's k-fold cross-validation	₲ ₲. (≡),
Source code			6
k-fold cross-validation for models	i		
nputs – Set up them to sta	rt an execution	A 1 input with undefined value	
dataset-id	ılı	A Select a dataset	Select the dataset for training/test the model
k-folds	123	5	Select the number of folds to split the dataset
objective-id	text		Objective field ID, or empty for default dataset objective field
nissing-splits	boolean	False	Sets the model's missing_splits flag
stat-pruning	boolean	False	Sets statistical pruning flag
Outputs			6

Figure 7.78: Model's k-fold cross-validation configuration

### 7.4.5.3 Ensemble's k-fold cross-validation

You can configure the following inputs for your ensemble's k-fold cross-validation (see Figure 7.79):

- dataset-id: select an existing dataset by clicking on the selector and searching the dataset by name.
- **k-folds**: set the number of partitions for your dataset. Your dataset will be split into *k* subsets creating *k* different ensembles and *k* different evaluations. By default *k*=5, a *k*=10 is also commonly used.
- **objective-id**: set the ID of the field you want to be the <u>objective field</u>. You can find the field ID in your dataset view mousing over the field type. It can be a categorical or numeric field. If no ID is given, BigML takes the dataset default objective field. (See Subsection 2.4.1.)
- **number-of-models**: total number of models in the ensemble. You can choose between 2 and 1,000 models. By default it is set to 10 models. (See Subsection 2.4.4.)
- **missing-splits**: set this input to true to consider missing values as valid values in your ensemble. By default it is set to false. (See Subsection 2.4.6.2.)
- **stat-pruning**: apply statistical pruning to all the tree nodes in order to avoid overfitting. By default this option is disabled and the smart pruning is enabled. (See Subsection 1.4.3.)
- **balance-objective**: enable this option to let BigML automatically balance the classes of the objective field. This is only available for classification ensembles. By default this option is disable. (See Subsection 2.4.8.1.)
- **weight-field**: weigh the instances considering the values of one field. You need to input the field ID. The selected field must be numerical and it must not contain missing values. This is valid for both regression and classification ensembles. (See Subsection 2.4.8.3.)
- **objective-weights**: set a specific weight for each class of the objective field. You need to list the weights in the same order that classes are found in your dataset histogram, e.g., to weight red, blue and yellow classes you need to input [2,3,1]. If a class is not listed, it is assumed to have a

weight of 1. Weights of 0 are also valid. This option is only available for classification ensembles. (See Subsection 2.4.8.2.)

- **node-threshold**: when the number of computed nodes is greater than this threshold, model growth stops. You can set a value between 3 and 2,000. By default it is set to -1 which indicates that no threshold applies. This parameter is useful to avoid overfitting. (See Subsection 2.4.6.3.)
- **sample-rate**: set a percentage of the dataset to build each single tree. By default it is set to 100% with replacement. (See Subsection 2.4.9.1.)
- replacement: enable this parameter to allow a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once. For ensembles, you need to set this parameter to true if your sampling rate is 100% to ensure single trees are built using different subsets of your dataset. (See Subsection 2.4.9.3.)
- randomize: set this input to true to use Random Decision Forests algorithm instead of Bagging as the algorithm to build the ensemble. (See Subsection 2.4.3.)
- **seed**: write any character to randomize the ensemble sampling. By leaving this input in blank the random-number generator will always use the same seed, producing repeatable results. By default it is configured to create random samples. (See Subsection 2.4.9.)

	dels - Clusters	Anomalies	Associations	Predictions	Tasks	-		Scrip	
	•	(=) Ensemi	DIE'S K-TOID CR	oss-validation	1	4		- (≡) ^{\$} -	
Source code									◙
k-fold cross-validation for ense	mbles								
Inputs – Set up them to s	tart an execution	1 input with	undefined value						
dataset-id	ll	A Select a dat	aset		Ŧ	Select the dataset for model	or training/tes	st the	
k-folds	123	5				Select the number of dataset	f folds to spli	t the	
objective-id	text					Objective field ID, or default objective of t			
number-of-models	123	10				Select the number of ensemble	f models in th	10	
missing-splits	boolean	False			¥	Sets the model's mi	ssing_splits f	lag	
Outputs	_					1			◙
New Execution name:									
Ensemble's k-fold cross-	validation's execu	tion				Reset		Execute	

Figure 7.79: Ensemble's k-fold cross-validation configuration

# 7.5 Visualizing Evaluations

After evaluations are created, you can visualize them in the BigML Dashboard.

Different visualizations are provided for **classification** (see Subsection 7.5.2) and **regression** (see Subsection 7.5.3) evaluations. Both types of evaluations shared the **original resources information** at the top of the view (see Subsection 7.5.1).

**Cross-validation** visualizations are very similar to single evaluations, the main differences are explained in Subsection 7.5.4.

# 7.5.1 Original Resources Information

For single evaluations, the original model and testing dataset used to create the evaluation will appear at the top of the evaluations view along with their corresponding icons so you can come back to see the resources. (See Figure 7.80.)

urces Datasets Supervised	<ul> <li>Unsupervised</li></ul>	ons 🔻 Tasks	WhizzML
₽ <b>₽</b> ~ ••••	90% Diabetes diag	nosis dataset with Diabetis	s 📋 🤣₊ (≡)⁵₊
Diabetes Diagnosis D	ataset   Training (70 🙆	Diabetis Diagn	osis Dataset   Test (30%)
ිකී 📶 🕂 Diabetes	Fri, 09 Jun 2017 16:34:40	1 R	Wed, 08 Jun 2016 15:39:
18.9 KB 9 size fields	537 20 instances models	8.1 KB size	10 231 fields instances
Description:	$\odot$	Description:	•
Description: Sampling and Ordering options:	© ⊘	Description: Sampling and Ordering op	
	-		
Sampling and Ordering options:	۲	Sampling and Ordering op	otions:
Sampling and Ordering options:	537	Sampling and Ordering op	231
Sampling and Ordering options:	537 100%	Sampling and Ordering op	231 100%

Figure 7.80: Example of an ensemble evaluation top view information

The information for each resource includes some of the parameter values used to create them:

- For models, ensembles, logistic regressions and deepnets:
  - Model type (regression or classification).
  - Objective field name.
  - Model size.
  - Number of fields: used to build the model.
  - Number of instances: used to build the model.
  - Description (see Section 1.10.)
  - Sampling and ordering options (see Subsection 1.4.7 and Subsection 1.4.8.)
    - * Instances: number of sampled instances.
    - * Sample rate: percentage of the training dataset used to build the model.
    - * Range: instances selected to build the model.
    - * Sampling: deterministic or random.
    - * Replacement: true or false.
    - * Out of bag: true or false.
    - * Ordering: linear, random or deterministic.
- Ensembles also include:
  - Type: Decision Forests or Boosted Trees (See Subsection 2.4.3)
  - Number of models: total single models in the ensemble
- For testing **datasets**:

- Dataset size
- Number of fields: in the testing dataset
- Number of instances: in the testing dataset
- Description: see Subsection 7.9.2
- Sampling and ordering options (See Subsection 7.4.4.)
  - * Instances: number of sampled instances
  - * Sample rate: percentage of the testing dataset used to create the evaluation
  - * Range: instances selected to sample the testing dataset
  - * Sampling: deterministic or random.
  - * Replacement: true or false.
  - * Out of bag: true or false.
  - * Ordering: linear, random or deterministic.

### 7.5.2 Classification Evaluations

For classification models, BigML provides several views that are accesible by clicking in the corresponding icons at the top menu.

#### 7.5.2.1 General Evaluation

This first visualization provides the **general confusion matrix** which contains the correctly classified instances as well as the errors made by the model for each of the objective field classes whithout applying any threshold.

In the table rows you can find the **predictions** for the **positive class** and the **negative classes**, and in the **columns** the **actual instances** in the testing dataset. You can transpose rows by columns by clicking on the switcher shown in Figure 7.81. You can also download the confusion matrix in Excel format by clicking the option shown in Figure 7.81.

	D	iabetes diagnosis	dataset   Training (8	0%) vs	<b>5</b> -	(≡) ^{\$} , (
Diabetes Diag	nosis Dataset   Tra	ining (8 💿	Diabete	s Diagnosis Dataset	Test (20º	%) 🔊
						. 0
Positive class: All classes		Ŧ				
				<b>1</b>	FN FP TI	<b>P</b>
ACTUAL VS. PREDICTED	False	True	ACTUAL	RECALL	F	Phi
False	85	14	99	85.86%	0.82	0.46
True	23	32	55	58.18%	0.63	0.46
PREDICTED	108	46	154	72.02% AVG. RECALL	0.73 AVG. F	0.46 AVG. Phi
PRECISION	78.70%	69.56%	74.13% AVG. PRECISION	75.97% ACCURACY		
	76.0% Accuracy			0.7275 F-measure		
74.19 Precision	ó		.0%		611	

Figure 7.81: Confusion matrix view

If you select a postive class in the selector, the confusion matrix cells will be painted according to the True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). By hovering a class in the table you get also these colors Figure 7.82.

Diabetes Diagnosis Dataset   Training (8 O Diabetes Diagnosis Dataset   Training (8	ataset   Test (20%) 💿
	a <b>l</b> 0 d
psitive class: True 💌	
	TP FN FP TN
ACTUAL VS. PREDICTED False True ACTUAL RECALL	F Phi
False 85 14 99 85.86%	0.82 0.46
True 23 32 55 58.18%	0.63 0.46
PREDICTED 108 46 154 72.02%	0.73 0.46 AVG. F AVG. Phi
PRECISION 78.70% 69.56% 74.13% 75.97% AVG.PRECISION 76.07%	
76.0% 0.633 Accuracy F-measure	
69.6% 58.2% Precision Recall	0.4611 Phi coefficient

Figure 7.82: Select a positive class

In the metrics below the confusion matrix you will see the performance measures for the class selected as the positive class such as **Precision**, **Recall**, **F-Measure**, **Phi Coefficient** and **Accuracy**. If you select "All classes", you will get the averages for the overall model. You can also compared these metrics to other two types of models: one using the **mode** as its prediction and the other predicting a **random** class of the objective field. At the very least, you would expect your model to outperform these weak benchmarks. (See Figure 7.83).

Diab	etes Diagr	nosis Dataset   T	raining (8	ତ	Diabete	s Diagnosis Data	aset   Test (20	9%) 🔊
		$\mathcal{O}$					Hide m	etrics comparing v
							mo	de-based decision
tive class:	rue		Ŧ					
							TP FN FP	TN
ACTUAL VS. PR	DICTED	False	True		ACTUAL	RECALL	F	Phi
False		85	14		99	85.86%	0.82	0.46
True		23	32		55	58.18%	0.63	0.46
PREDICT	ED	108	46		154	72.02% AVG. RECALL	0.73 AVG. F	0.46 AVG. Phi
PRECISIO	N	78.70%	69.56%		74.13% AVG. PRECISION	75.97% ACCURACY		
MODEL 76.0%		MODE 64.3%	RANDOM 50.0%		MODEL 0.6337	MODE 0		.438
10.0%			00.070		0.0007	-	0	.+00
		Accuracy				F-measure		
	MODE	RANDOM	MODEL	MODE	RANDOM	MODEL	MODE	RANDOM
MODEL	WODE							

Figure 7.83: Performance metrics and benchmark models view

The confusion matrix will display up to **five different classes**. If your model contains more than five classes in the objective field, you can download the confusion matrix in Excel format by clicking the option shown in Figure 7.84.



Figure 7.84: Download confusion matrix

### 7.5.2.2 Evaluation curves

BigML offers four different evaluation curves: the **ROC curve**, the **Precision-Recall curve**, the **Gain curve** and the **Lift curve**. All of them are explained in detailed in <u>Subsection 7.2.1.5</u>.

You can select the **positive class** and the **threshold** for each curve and you will see the confusion matrix values and the metrics changing in real-time (see Figure 7.85).



Figure 7.85: Select the evaluation curve, the positive class and the threshold

The confusion matrices for the curves have always two columns and two rows although there are more than two classes in the objective field. This is because the postive class is the one predicted given a certain threshold and if the threshold is not met, the rest of classes are aggregated as a single "Negative class" and predicted instead. Read more about positive and negative classes in Subsection 7.2.1.2).

From each curve visualization you can:

- · Download the confusion matrix in Excel format
- Download the chart in PNG format with or without legends.

Diabetes Dia	ignosis Dataset   Trai	ning (8 ତ	Diabetes I	Diagnosis Dataset   T	est (20%)
					00
sitive class: False		Ŧ	Probability t	threshold:	50
				E TP FN	FP TN
ACTUAL VS. PREDICTED	False	True	ACTUAL	RECALL	F Phi
False	87	12	99	87.88%	0.82 0.46
True	25	30	55	54.55%	0.62 0.46
PREDICTED	112	42	154	71.21% AVG. RECALL	0.72 0.46 AVG. F AVG. Phi
PRECISION	77.68%	71.43%	74.55% AVG. PRECISION	75.97% ACCURACY	
80%		ا مــــــــــــــــــــــــــــــــــــ	76.0 ¹		SIMAGE SIMAGE WITH LEGENDS F-measure
70%			77.7% Precision	87.9% _{Recall}	0.4564 Phi coefficient
60% 50% 40% 30%			45.5% FPR	72.7% % positive instances	120.8%
20%	ROC AUC: 0.7941		50.9% K-S statistic	0.4009 Kendall's Tau	0.4883 Spearman's Rho

Figure 7.86: Export options for the threshold confusion matrices and the evaluation curves

# 7.5.3 Regression Evaluations

You can visualize the **regression measures** in the green boxed histograms: **Mean Absolute Error**, **Mean Squared Errror** and **R Squared**. Read more about regression measures in Subsection 7.2.2.



Figure 7.87: Performance measures for regression models, ensembles, deepnets, and fusions

By default, BigML provides the measures of two other types of models to compare against your model performance. One of them uses the **mean** as its prediction and the other predicts a **random** value in the range of the objective field. At the very least, you would expect your model to outperform these weak benchmarks. You can remove the benchmarks by clicking the icons shown in Figure 7.88.



Figure 7.88: Compare regression models against random and mode values

# 7.5.4 Cross-Validation Visualization

Since cross-validation measures are calculated by averaging the single k evaluations, you cannot find the confusion matrix and evaluation curves explained in Subsection 7.5.2. Instead, the main metrics are plot in different histograms. Additionally, you can see the **standard deviation** per measure under the sigma symbol icon. You can find a cross-validation example for a classification model below.



Figure 7.89: Cross-validation view for classification measures

At the top of the view, you can find the list of the single evaluations used to compute the cross-validation measures by clicking on the **Evaluations** panel. (See Figure 7.90.)

Sources Datasets <b>Evaluations -</b> Clusters Anomalies	s Associations	Predictions	Tasks	Scripts 👻
	-validation Evalu	ation	Ê	. ૐ- (≡)∕- Û
Evaluations				۲
Evaluation of Diabetes v2 - extended's model with Diabetes	s v2 - extended			
Evaluation of Diabetes v2 - extended's model with Diabetes	s v2 - extended			
Evaluation of Diabetes v2 - extended's model with Diabetes	s v2 - extended			
Evaluation of Diabetes v2 - extended's model with Diabetes	s v2 - extended			
Evaluation of Diabetes v2 - extended's model with Diabetes	s v2 - extended			
A/B				
©∰ #0	All Classes		¥	
AV	ERAGE ACCURACY			
100% 75% 50% 25% 25%	70.82%		σ	
	CV MODEL			
AVERAGE PRECISION			AVERAGE RECALL	
100% 75% 68.06% 50% 25% 0%	σ 100% 75% 50% 25% 0%		68.38%	σ
CVMODEL			CVMODEL	

Figure 7.90: Cross-validation single evaluations

# 7.6 Evaluation Comparison

Evaluation comparison allows you to identify the algorithms and configurations that yield the highest performance. In BigML you can compare two evaluations side by side or multiple evaluations simultaneously. Both options are explained in the following subsections.

# 7.6.1 Compare Evaluations Side By Side

In BigML, you can easily compare two evaluations side by side by clicking COMPARE EVALUATION in the **1-click action menu** option from the evaluation view. (Figure 7.91). This option is deprecated for the new evaluations that include the evaluation curves explained in Subsection 7.2.1.5.

Sources	Datasets	Supervised -	w) Unsupervised ▼	Predictions -	Tasks			WhizzML 🔻
The ROC cu	irve and other e	valuation curves car	not be displayed for previ	ously built evaluati	ons, please	e make the evaluation again if you want to	visualize them.	)
	••••		Evaluation of	of Loan risk d	ata   mo	del balanced	<b>G</b> -	(≡)≁_
						RE-EVALUATE		
	Loop Diel	Data   Trainin	g (80%) Model	ె	0	COMPARE EVALUATIONS		୕
229	Loan Risi	CData   Trainin	g (80%) Model	Ŭ	009	DELETE EVALUATION		
Configu	uration				- L	MOVE TO		୕
oomiga								0
	Confusio	n Matrix						⊽
			A/I	B	>			
				All Classes		٣		
				ACCURAC	Y			
			100% 75% 50% 25% 0%	67.00	%			
				MODE	EL.			

Figure 7.91: Compare evaluations using 1-click action menu

Alternatively, you can use the COMPARE EVALUATIONS menu option, from the **pop up menu**. (Figure 7.92.)

			Eva	aluations			Ô	Q 👩
ណ	ill	Name	4	≎ Type	Performance :	≎ # ≎	<u>م</u> \$	Instances
6 <u>6</u> 7	ılî	Evaluation of Iris dataset - sample (80.00%)'s	wi	00		4d 21h	921 bytes	30
° <mark>8</mark>	ıl <mark>e</mark>	Evaluation of Iris dataset - sample (80.00%)'s	wi	00	•	4d 21h	921 bytes	30
2 <mark>2</mark>	ılî	15% Iris dataset - sample (80.00%)'s with Iris o	la	o0		6d 22h	921 bytes	30
22 2	ılî	30% iris-setosa- Iris dataset - sample (80.00%)	5	00		6d 22h	921 bytes	30
<u></u>	ılî	90% (k) US car accidents dataset - sample wit	h	00		1w	741.0 KB	4K+
۴	ılî	90% Diabetes diagnosis dataset with Diabetis	di 🗙	00		1w 2d	8.1 KB	231
2 <mark>2</mark> 3	ılî	90% Diabetes c VIEW DETAILS		00		1w 2d	8.1 KB	231
2 2 2	ılî	30% Iris datase ↔ COMPARE EVALUATIONS	UATL	00		1w 2d	921 bytes	30
29. 21.	ılî	Evaluation of B		00		1w 3d	38.3 MB	25K+
° <mark>8</mark> °	ıl	Evaluation of tr		>		1w 3d	51.7 KB	760

Figure 7.92: Compare evaluations using pop up menu

You can also click on the COMPARE EVALUATIONS list view from the **1-click action menu** in the evaluation list view. (Figure 7.93).

Sources	Data	sets Supervised  Vinsupervised  Prec	dictions •	Tasks					WhizzN	il •
			Eval	uations				Ô	Q,	∽.
ណ	ıll	Name	\$	Type 🗘	Performance	00 ⁺	EVALUATE A MODEL			
ŕå	ılî	Evaluation of Iris dataset - sample (80.00%)'s wi		00		00	EVALUATE AN ENSE	MBLE		
							EVALUATE A LOGIST	IC REGRE	SSION	
£	ılî	Evaluation of Iris dataset - sample (80.00%)'s wi		00	•		COMPARE EVALUAT	IONS		
68	ılı	15% Iris dataset - sample (80.00%)'s with Iris da		00			COMPARE MULTIPLE	EEVALUA	TIONS	
ŕå	ılı	30% iris-setosa- Iris dataset - sample (80.00%)'s		00			6d 22h	921 bytes		30
×	ılı	90% (k) US car accidents dataset - sample with		00	-		1w	741.0 KB		4K+

Figure 7.93: Compare evaluations using 1-click action menu from list view

Using any of the options you will be redirected to the **Compare Evaluations** view where you will be able to select the evaluations you want to compare. (Figure 7.94.)

Sources	Datasets	Evaluations -	Clusters	Anomalies	Associations	Predictions	Tasks	Scripts 🗸
A II	r 12			Com	oare Evaluatio	ons		
	Evaluation	of Diabetes diagnosis	s dataset - san	nple ( × •		Search evalu	uation	<u>^</u> م
00 E	valuation Of	Diabetes Diagno	sis Dataset ·	Sam 🕞		n on Please enter		
					With	e sume sumpling	parameters.	

Figure 7.94: Select the other evaluation

# Note: you can only compare two evaluations that use the same testing dataset and the same evaluation configuration; otherwise evaluations are not comparable.

The performance measures of your selected evaluations will be displayed side by side. BigML automatically computes the variation of the right-hand-side evaluation compared to the left-hand-side one. You can find those variations next to each performance measure of the right-hand-side evaluation. You can also select another evaluations from the evaluations selectors. (See Figure 7.95.)



Figure 7.95: Compare evaluations side by side

# 7.6.2 Compare Multiple Evaluations

The main goal of comparing multiple evaluations is to analyze different model performances by plotting their evaluation results in a chart. BigML evaluation comparison offers a flexible and visual way to compare multiple **classification models** built using different algorithms (models, ensembles, logistic regression, deepnets, or fusions) and/or different configurations as long as the **testing dataset is the same**. Cross-validation evaluations are not eligible for the Evaluation comparison tool since they use different testing datasets to compute the averaged measures.

You can compare multiple **classification** evaluations using the option from the **1-click action menu** as shown in Figure 7.96. Previously built evaluations to the evaluation curves explained in Subsection 7.2.1.5 do not show this option anymore.

	Ipervised - Unsup	ervised  Prediction:	- Tasks			WhizzML
₩. ••••		Diabetes diagnosis	dataset   Training (8	30%) vs	<b>5</b> -	(≡) ^{\$} -
			RE-E	VALUATE		
(D) ()			Сом			
Diabetes Dia	gnosis Dataset   Tr	raining (8 💿	DELE	TE EVALUATION		C
	1 1		моч	ETO		
						0 0 a
ositive class: All classes		v				
All classes		×		ПП ТР	FN FP TN	
All classes	False	True	ACTUAL	RECALL TP	FN FP TN	
ACTUAL VS. PREDICTED	False 85		ACTUAL 99		FN FP TN F 0.82	<b>P</b>
ACTUAL VS. PREDICTED		True		RECALL		Phi
ACTUAL VS. PREDICTED	85	True 14	99	RECALL 85.86%	0.82	Phi 0.46

Figure 7.96: Compare multiple evaluations using 1-click action menu

You can also click on the COMPARE MULTIPLE EVALUATIONS list view from the **1-click action menu** in the evaluation list view. (Figure 7.97).

Sources	a Data	sets Supervised   Predict	ions	new Tasl	s			٧	VhizzML 🔻
			Eval	uations	;			ŵ	ৎ 👶
ŝ	ill	Name	\$	Туре	٥	Performance \$	₩ ≎	<b>ê</b> \$	Instances
×	ılî	Happiness survey dataset - sample (80.00%) vs. Hap L1 regularized (c=1.0), auto-scaled, eps=0.1575, stats, b		00			1d	612.0 KB	10K+
<u> </u>	ılî	Happiness survey dataset - sample (80.00%) vs. Hap L2 regularized (c=1), bias, auto-scaled, missing values		00			1d	612.0 KB	10K+
få	ılî	Happines: Sur Happiness survey dataset - ^{OVA} ) s. Hap 20-iteratio sample (80.00%) vs. Happiness Sample		00			1d	612.0 KB	10K-
ŕå	ılı	Happines: Sun survey sample (20.00%) 0%) vs. Hap 10-model, statistics 512-node, pruned statistic		00			1d	612.0 KB	10K-
к <mark>2</mark>	ılî	Happiness survey dataset - sample (80.00%) vs. Hap 512-node, pruned	8	00			1d	612.0 KB	10K-
к <mark>9</mark>	ılî	Happiness surverse VIEW DETAILS		.0			1d	612.0 KB	10K
K1	ılî	US car acciden 512-node, prune 512-node, prune Billion Delete evaluation		.0			1d 8h	148.5 KB	11
۶	ılî	US car acciden					1d 8h	148.5 KB	11
ŕå	ılî	Loan risk data dataset - sample (80.00%) vs. Loan ris No extra information		.01			1d 8h	27.2 KB	20
۶	ılî	Loan risk data dataset - sample (80.00%) vs. Loan ris No extra information		o0			1d 8h	27.2 KB	20
Show 10	ᅌ evalu	uations 11 to 20	of 384	evaluations	3		к	<b>〈</b> 1 2 3 4	5 > >

Figure 7.97: Compare multiple evaluations using 1-click action menu from the list view

A chart displaying the curves for the current evaluation will be displayed. You can select any of the four evaluation curves offered by BigML: **Precision-Recall curve**, **ROC curve**, **Gain curve** and **Lift curve**. (See Figure 7.98.) You can find a detailed explanation of the evaluation curves in Subsection 7.2.1.5.



Figure 7.98: Select an evaluation curve

You can plot different evaluations in the chart by clicking the evaluation selector shown in Figure 7.99. You will only be able to select other **comparable evaluations**, i.e., other evaluations that were built using the **same testing dataset**. BigML allows to compare up to 150 evaluations simultaneously selecting them in groups of 20 evaluations from the selector.



Figure 7.99: Select evaluations to compare

Your selected evaluations will appear in the **chart** with different colors. The evaluation curves and metrics will be calculated according to the **positive class** selected. You can select to **sort** your evaluations in the legend by any of the available **metrics**: the ROC AUC, the PR AUC, the K-S statistic, the Kendall's Tau or the Spearman's Rho. In the legend, for each evaluation, you will have the **name**, the icon of the model used, the **sorting metric**, and the **configuration parameters** used to create the model. By hovering over the evaluations in the legend you will be able to hide or remove them at any time.



Figure 7.100: Manage evaluations from the legend

Apart from the ROC AUC and PR AUC metrics shown in the legend for, you will also be able to see the **PR AUCH** and **ROC AUCH** by hovering over the evaluation names. Again, refer to Subsection 7.2.1.5 for a detailed explanation about the AUCH (Are Under the Convex Hull).



Figure 7.101: See the PR AUCH and ROC AUCH by hovering over the evaluation name in the table

If you select the Gain curve you will also obtain the **K-S statististic** by hovering over the evaluation name in the legend. (See Subsection 7.2.1.5 for a detailed explanation).



Figure 7.102: See the K-S statistic by hovering over the evaluation name in the table

The panel below the chart, contains a table with all the selected evaluations, their metrics and a set of model parameters to help you identify which algorithm and configuration performs better (see Figure 7.103):

- ROC AUC: the Area Under the Curve of the ROC curve.
- PR AUC: the Area Under the Curve of the Precision-Recall curve.
- K-S: the K-S statistic of the Gain curve.
- Tau: the Kendall's tau coefficient.
- · Rho: the Spearman's rho coefficient.
- **Model Type**: if the resource evaluated is a single model, ensemble, logistic regression, deepnet, or fusion.
- Number of nodes: maximum number of splits set for training the model or ensemble.
- **Number of models**: number of models set for training the ensemble. For models, it will always be 1.
- % of Bagging: percentage of the dataset instances used for training the single trees composing the ensemble.
- Randomize: if the single trees composing an ensemble take a random sample of fields per split.
- Boosted: if the ensemble is composed of Boosted trees.
- Balanced: if the model or ensemble has been built previously balancing the objective field.
- **Missing splits**: if the model or ensemble has been built taking into account the missing values in the dataset.
- **Default Numeric**: if the logistic regression has been built replacing the missing values by the field mean, maximum, minimum or zero.
- · Auto-scaled: if the fields in the logistic regression have been auto-scaled.
- Bias: if the logistic regression includes the bias term.

You can find a detailed explanation of each configuration parameter depending if the model is a single tree (Section 1.4), an esemble (Section 2.4), a logistic regression (Section 4.4), or a deepnet (Section 5.4) in the corresponding sections.



Figure 7.103: Table containing all the evaluations and the model parameters

#### 7.6.2.1 Exporting Evaluation Comparison

You can **export the multiple comparison chart** in PNG format with or without legends by clicking the icon shown in Figure 7.104. The legend includes the **evaluations names**, the selected **sorting metric** and the **model configuration** options.



Figure 7.104: Export multiple comparison chart in PNG format

## 7.7 Consuming Evaluations

BigML allows you to create and consume your evaluations programmatically via the **BigML API and bindings**. The following subsections explain both tools.

#### 7.7.1 Using Evaluations Via the BigML API

You can perform all the evaluation actions explained in this document via the BigML API such as creating, configuring, retrieving, listing, updating, and deleting evaluations.

See below how to create an evaluation just using a model and a testing dataset once you have properly set the BIGML_AUTH environment variable to contain your authentication credentials:

Apart from all the BigML Dashboard actions, you can also create an evaluation using **multiple models** and **multiple datasets** via the BigML API:

- Provide multiple model IDs, and the models will be evaluated as an ensemble and you can select your preferred voting strategy option. (See Subsection 7.4.2.)
- Provide a list of datasets identifiers, and the corresponding datasets will be concatenated and used as input. You can also sample the resulting dataset. (See Subsection 7.4.4.)

For more information on using evaluations through the BigML API, please refer to documentation²³.

#### 7.7.2 Using Evaluations Via the BigML Bindings

You can create and use evaluations via BigML bindings which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings for several languages including Python, Node.js, Java, Swift or Objective-C. You can find an example to create an evaluation with Python bindings below.

For more information on BigML bindings, please refer to the bindings page²⁴.

## 7.8 Evaluation Limits

Similarly to the rest of resources in BigML, evaluations have some limits in the number of classes, terms, and items allowed per field:

- · Classes: a maximum number of 1,000 distinct classes per field is allowed.
- **Terms**: BigML can handle up to 1,000 tokens in total. In case multiple text fields are defined, then the token limit per field is evenly divided by the number of text fields, e.g., a dataset with two text fields would result in 500 terms per text field. BigML selects those terms with most significant frequency, discarding both those that appear either too often or too infrequently. Moreover, a maximum of 256 characters are allowed.
- Items: a maximum number of 10,000 distinct items per field is allowed.

## 7.9 Descriptive Information

Each evaluation has an associated **name**, **description**, **category**, and **tags**. The following subsections provide a brief description for each concept. See the options that the **More info** menu option gives to edit them (Figure 7.105).



Figure 7.105: Panel to edit an evaluation name, description, category, and tags

²³https://bigml.com/api/evaluations

²⁴https://bigml.com/tools/bindings

#### 7.9.1 Evaluation Name

Each evaluation name is displayed on the list view and also on the top bar of an evaluation view. Evaluation names are indexed to be used in searches. When you create an evaluation, it gets a default name. You can change it using the MORE INFO menu option on the right corner of the evaluation view (see Figure 7.105) The name of an evaluation cannot be longer than **256** characters. More than one evaluation can have the same name even within the same project, but they will always have different identifiers.

#### 7.9.2 Evaluation Description

Each evaluation also has a **description** that it is very useful for documenting your Machine Learning projects. Evaluations take the description of the models used to create them.

Descriptions can be written using plain text and also markdown²⁵. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 7.106.)

Edit description	×
You can add formatting and links using a simple markdown language B $I \models = \infty$	e
Write your **description** here	
Description: Write your <b>description</b> here	
Cance	Update

Figure 7.106: Markdown editor for evaluations descriptions

Descriptions cannot be longer than 8192 characters and can use almost any character.

#### 7.9.3 Evaluation Category

A **category**, taken from the model used to create the evaluation, is associated with each evaluation. Categories are useful to classify evaluations according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers.

An evaluation category must be one of the categories listed on Table 7.2.

²⁵https://en.wikipedia.org/wiki/Markdown

Category
Aerospace and Defense
Automotive, Engineering and Manufacturing
Banking and Finance
Chemical and Pharmaceutical
Consumer and Retail
Demographics and Surveys
Energy, Oil and Gas
Fraud and Crime
Healthcare
Higher Education and Scientific Research
Human Resources and Psychology
Insurance
Law and Order
Media, Marketing and Advertising
Miscellaneous
Physical, Earth and Life Sciences
Professional Services
Public Sector and Nonprofit
Sports and Games
Technology and Communications
Transportation and Logistics
Travel and Leisure
Uncategorized
Utilities

Table 7.2: Categories used to classify evaluations by BigML

#### 7.9.4 Evaluation Tags

An evaluation can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide evaluations with some extra information. An evaluation inherits the tags from the model used to create it. Each tag is limited to a maximum of 128 characters. Each evaluation can have up to 32 different tags.

## 7.10 Evaluation Privacy

The link displayed in the **Privacy** panel is the private URL of your evaluation, so only a user logged into your account is able to see it.

BigML allows you to share your evaluations by enabling the **secret link**. Just click the switcher icon, copy the link and share it with others.

	Sources Datasets <b>Evaluations •</b> Clusters Anomalies Associations Predictions Tasks	Scripts 👻
	■ If #A •••• Evaluation_titanic_ensemble_balanc	🔋 🕉 🛈
	DETAILS	$\odot$
	INFO	۲
	PRIVACY	۲
	Private link   Only you can view this evaluation logged into your account.	
	https://bigml.com/dashboard/evaluation/5679c664200d5a7cc40019c1	
Share	Secret link   Only people with the secret link can view this evaluation.	
secret link	thtps://bigml.com/shared/evaluation/38RDHpEtkqpMGGRPpHbqCGT9dO1	Ø

Figure 7.107: Share evaluations

## 7.11 Moving Evaluations to Another Project

When you create an evaluation, it will be assigned to the same project where the original model is located. However, you can move evaluations between projects from two different places:

• Clicking the MOVE TO... option in the **1-click action menu** from the evaluation view:

Sources Datasets Evaluations  Clusters	Anomalies Associations	Predictions Tasks	Scripts 👻
	Evaluation of Diabetes d	liagnosis data	📋 🝊 🛈
Diabetes Diagnosis Dataset - Sam	ple (80 오	Diabete	NMODEL )
	NEW PROJECT	MOVE TO	
	PROJECTS PROJECT 1 PROJECT 2	v	
	ACCURACY		

Figure 7.108: Move evaluation from the 1-click action menu

• Clicking the MOVE TO... option in the **pop up menu** from the evaluation list view:

Sources	Data	sets <b>Evaluations</b> Clusters Anomalies	Ass	ociations	Predictions	Tasks		Scripts -
			Eval	uations				م <b>لاج</b>
វា	III	Name	\$	Туре	Performance	≎ ∰ ≎	ê \$	Instances
к <mark>9</mark>	ılî	Evaluation of Diabetes diagnosis dataset - sample	۲	00		1d21h	5.4 KB	154
ŕå	ılı	Evaluation of Fi		$\mathbf{\Lambda}$	_	2d 3h	10.1 KB	318
é 🖧	ılî	Evaluation of Fi		入		2d 3h	10.1 KB	318
5	ıl <mark>î</mark>	Evaluation of Fi 🎴 MOVE TO			EW PROJECT		10.1 KB	318
e 👌	ılî	Evaluation of Diabetes diagnosis dataset - balanc		PROJECT			5.4 KB	154
5 <b>2</b>	ılı	Evaluation of Diabetes diagnosis dataset - balanc			JECT 1		5.4 KB	154
£	ılı	Evaluation of Diabetes diagnosis dataset - sample		00		2a 3n	5.4 KB	154
к ^о	ılı	Evaluation of Diabetes diagnosis dataset - sample		00		2d 3h	5.4 KB	154
Show 10	ᅌ evalı	uations 1 to	8 of 8 ev	aluations			14	

Figure 7.109: Move evaluation from the pop up menu

## 7.12 Stopping Evaluation Creation

You can stop the creation of an evaluation before the task is finished by clicking the DELETE EVALUATION option from the **1-click action menu** from the evaluation view (Figure 7.110), or from the **pop up menu** on the evaluation list view (see Figure 7.111).



Figure 7.110: Stop evaluation creation

Sources	Data	sets <b>Evaluations</b> Clusters Anomalies	Asso	ociations	Predict	tions Ta	sks	\$	Cripts 🔻
			Evalu	uations	;			C	<b>(</b> ).
ŵ	III	Name	\$	Туре	Perfor	mance ≎	÷	<b>₿</b> ♦ II	nstances :
fr	ılî	Evaluation of Arrythmia diagnosis dataset's with	۲	00			0min	536.2 KB	452
é 😚	ılî	Evaluation of A		o0	_		8min	536.2 KB	452
×	ılı	Evaluation of Ir		00	_		1d 7h	921 bytes	30
×	ılı	Evaluation of Ir MOVE TO	)	00	_		1d 7h	921 bytes	30
é 😚	ılî	Evaluation of Iris dataset   Training (80%)'s with I		00	_		1d 7h	921 bytes	30
ŕå	ılı	Evaluation of Iris dataset v2's dataset - sample w		00			1d 7h	921 bytes	30
۶	ılî	Evaluation of Iris dataset   Training (80%)'s with I		00			1d 7h	921 bytes	30
×	ılı	Evaluation of Diabetis diagnosis dataset with Di		oD			2w 3d	8.1 KB	231
ŕå	ılî	Evaluation of Diabetis diagnosis dataset with Di		00			3w3d	5.4 KB	154
۶	ılı	Evaluation of Fictional Wine Sales dataset's with		00	-		1m	50.4 KB	1K+
Show 10	ᅌ evalı	uations 1 to 1	0 of 50 ev	aluations			K	< 1 2 3 4	5 > >

Figure 7.111: Stop evaluation from popu up menu

Note: if you stop the evaluation during its creation you will not be able to resume the same task. If you want to create the same evaluation you will have to start a new task.

## 7.13 Deleting Evaluation

You can delete your evaluations by clicking the DELETE EVALUATION option from the **1-click action menu** from the evaluation view, (Figure 7.112).



Figure 7.112: Delete evaluation from the 1-click action menu

Alternatively, you can click the DELETE EVALUATION option in the **pop up menu** on the evaluation list view (Figure 7.113).

Sources	Data	sets <b>Evaluations</b> Clusters Anomalies	Ass	ociations	Predictions	Tasks		Scripts -
			Eval	uations				۹ <b>(%</b> -
ŵ	ıll	Name	\$	Туре 🗘	Performance	≎ 🛗 ≎	<b>Å</b> \$	Instances 💠
к <mark>0</mark>	ılî	Evaluation of Diabetes diagnosis dataset - sample	۲	00		3min	5.4 KB	154
к ⁹	ılî	Evaluation of D			_	1d 21h	5.4 KB	154
£	ılı	Evaluation of Fi		木	_	2d 3h	10.1 KB	318
ŕå	ılı	Evaluation of Fi MOVE TO		> ∧	_	2d 3h	10.1 KB	318
K12	ılî	Evaluation of Fictional Wine Sales dataset with F		*	_	2d 3h	10.1 KB	318

Figure 7.113: Delete evaluation from popu up menu

A modal window will be displayed asking you for confirmation. Once an evaluation is deleted, it is permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it.

 ure you want to delete th	
 is evaluation, you will no longer have acces els and datasets.	s to it and you will need to recreate i

Figure 7.114: Delete evaluation confirmation

## 7.14 Takeaways

This chapter explains evaluations in detail. Here is a list of key points:

- An evaluation allows you to **measure** your model, ensemble, logistic regression, deepnet, and fusion **performance**.
- In BigML you can perform two types of evaluations: single **evaluations** and **cross-validation** evaluations.
- You need a model and a testing dataset to create a single evaluation. (See Figure 7.115.)
- You just need an existing dataset to create a cross-validation evaluation. (See Figure 7.116.)
- BigML provides you a range of **configuration** options before creating your evaluation.
- Performance measures are different for classification and regression models.
- The confusion matrix is a key element to evaluate the performance of classification models.
- You can compare your evaluations measures against models using the **mean**, the **mode**, and a **random** value to predict.
- BigML provides different visualizations for the **ROC curve**, the **Precision-Recall curve**, the **Gain curve**, and the **Lift curve** along with their AUC, K-S statistic and other metrics.
- You can **compare** two or more **evaluations** built with different configurations and algorithms to select the model with the best performance.

- You can download your confusion matrix in Excel format.
- You can create and use evaluations via the BigML API and bindings.
- You can add descriptive information to your evaluations.
- You can move your evaluations between projects.
- You can share your evaluations with other people using the secret link.
- You can **stop** your evaluations creation by deleting them.
- You can permanently **delete** an existing evaluation.



Figure 7.115: Single evaluations workflow



Figure 7.116: Cross-validation workflow

 $\ensuremath{\texttt{Please}}$  use the noidx option in the documentclass invocation.

# **List of Figures**

1.1		2
1.2		2
1.3	Models icon	2
1.4		6
1.5		8
1.6	A dataset with images and image features	9
1.7	A dataset with image feature fields shown	9
1.8	1-click model	0
1.9	1-click model from the dataset list view	0
1.10	Configure permanent parameter modal 1	1
1.11	Configure model	2
1.12	Automatic optimization	3
	Configure the missing splits and the weights for your model	3
1.14	Training duration	4
1.15	Model candidates	4
1.16	Model configuration options	5
	Weighting arguments for models 1	7
	Sampling arguments for models	8
	Ordering argument for models 19	9
	Create model after configuration	0
1.21	Model API request preview	0
1.22	Tree visualization	1
1.23	Node prediction	2
1.24	Prediction path	3
	Interactive filters	4
1.26	Create dataset from a branch	5
1.27	Model PDP view	6
1.28	Classification model PDP	7
	Regression model PDP	8
	Model PDP axis options	9
1.31	Prediction legend options	0
1.32	Input fields from in the model PDP	1
1.33	Sunburst visualization	2
	Change Sunburst colors	2
	Sunburst colors	3
1.36	Button to display a model's summary report	3
	Field importance	4
	Summary Report	5
	Predictions list empty view	
	Menu options of the predictions list view	6
	Single predictions icon	6
	Batch predictions icon	6
	Predictions list view	6

~ -
37
37
37
38
38
39
39
40
41
41
42
43
44
44
45
46
46
47
47
48
49
49
50
50
51
52
53
53
54
54 54
-
55
56
57
58
58
59
60
60
61
-
61
62
62
63
64
65
66
67
67
68
68
68
69
69
71
72
74
74
74
333334444444444444444444444444444444444

1.102Set a price for other users to make predictions with your model         1.103White Box models let other users to clone or purchase the full model         1.104Set a price for other users to clone your model and make predictions with it         1.105Public status changed to Black Box and the gallery link is available         1.105Public status changed to Black Box and the gallery link is available         1.105Public status changed to Black Box and the gallery link is available         1.105Public status changed to Black Box and the gallery link is available         1.105Public status changed to Black Box and the gallery link is available         1.105Public status changed to Black Box and the gallery link is available         1.105Public gallery         1.107Access to BigML Gallery .         1.107Access to BigML gallery         1.108BigML public gallery         1.109Modal window to confirm you want to buy the model's dataset         1.110Modal window to confirm you want to clone this model         1.111-click menu option to move models         1.112Pop up menu option to move models         1.113Pop up menu option to stop a model's creation         1.114Pop up menu option to stop a model's creation         1.115Menu option to delete a model         1.117Model deletion pop up menu option         1.118Model deletion modal window         1.119Model Workflows	75 76 76 77 77 77 78 78 79 79 80 80 81 81 82 82 82 82 84
<ul> <li>2.1 Ensembles list view</li> <li>2.2 Empty Dashboard ensembles view</li> </ul>	86 86
2.2         Empty Dashboard ensembles view           2.3         Ensemble icon	87
2.4 Field importance for ensembles	88
2.5 A dataset with images and image features	89
2.6 A dataset with image feature fields shown	90
2.7 1-click ensemble from dataset detail view	90
2.8 1-click ensemble from dataset list view	91
2.9 Configure permanent parameter modal	92
2.10 Configure ensemble	92
2.11 Ensemble objective field	93
2.12 Automatic optimization	94
2.13 Configure the missing splits and weights for your ensemble	95
2.14 Training duration	96
2.15 Ensemble candidates	96
2.16 Ensemble type: Decision Forests or Boosted Trees	97
2.17 Number of models	98
2.18 Number of iterations	98
2.19 Trees parameters configuration	100
2.20 Boosting parameters	
	103
2.22 Trees sampling for ensembles	104
· ····································	106
2.24 Ordering options for ensembles	107
· · · · · · · · · · · · · · · · · · ·	108 108
	108
<ul><li>2.27 Ensemble top menu</li><li>2.28 Switch from the chart view to the model list view</li></ul>	1109
	111
	111
2.31 Ensemble chart	112
	113
2.33 PREDICTION legend options	114
	115
	115
	116
2.37 One of the selected fields impact predictions	117
2.38 None of the selected fields impact predictions	
2.39 Ensemble model list view	119

	) Tree visualization					
	Predictions list empty view					
	2 Menu options of the predictions list view					
	3 Single predictions icon					
	Batch predictions icon					
	5 Predictions list view					
	Menu options to create predictions					
	7 Predict option from ensemble 1-click menu					
	Predict option from ensemble pop-up menu					
	Single predictions form					
	) Select fields in the prediction form					
	2 Single predictions list view					
	3 Select a single image source in the image input field					
	List the components of a composite source					
	5 Select a component of a composite source					
	Ensemble prediction form, image field and more					
	7 Ensemble single image prediction					
	Batch predictions option from ensemble 1-click menu					
	Batch predictions option from ensemble pop up menu					
	) Select dataset for batch predictions .					
	Configuration options displayed and output preview					
	2 Create dataset from batch predictions					
	Create batch predictions					
	Download batch prediction output CSV file					
	View batch predictions output dataset					
	Batch prediction using an image dataset					
	7 Missing strategies for single predictions					
2.68	3 Missing strategies for batch predictions					136
0.00	Design (1997) and the first second					400
	Probabilities, confidences or votes for Decision Forest single predictions					
2.70	Probabilities, confidences or votes for Decision Forest batch predictions	· ·	• •	· ·	· ·	139
2.70 2.71	Probabilities, confidences or votes for Decision Forest batch predictions Select probability, confidence or votes	  	•••	· · · ·	 	139 140
2.70 2.71 2.72	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> </ul>	  	  	  	· · · · · · · · · · · · · · · · · · ·	139 140 140
2.70 2.71 2.72 2.73	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> </ul>	· · · · · ·	  	· · · · · · · ·	· · · · ·	139 140 140 141
2.70 2.71 2.72 2.73 2.74	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	139 140 140 141 142
2.70 2.71 2.72 2.73 2.74 2.75	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	139 140 140 141 142 142
2.70 2.71 2.72 2.73 2.74 2.75 2.76	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·		139 140 140 141 142 142 142
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions .</li> <li>Select probability, confidence or votes .</li> <li>Select the positive class .</li> <li>Set a threshold .</li> <li>Configure a threshold for batchpredictions .</li> <li>Default numeric value for batch predictions .</li> <li>Fields Mapping for batch predictions .</li> <li>Output settings for batch predictions .</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · ·	139 140 140 141 142 142 142 143 145
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		139 140 140 141 142 142 143 145 146
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> </ul>	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	139 140 141 142 142 143 145 146 146
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> </ul>	· · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · ·	139 140 141 142 142 143 145 146 146 147
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> </ul>	· · · · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · ·	· · · · · · · · · · · · · · · · · · ·	139 140 140 141 142 142 143 145 146 146 147 148
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Input field importances</li> </ul>	· · · · · ·	· · · · · · · · · · · · · · · · · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	139 140 141 142 142 143 145 146 146 146 147 148 148
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions .</li> <li>Select probability, confidence or votes .</li> <li>Select the positive class .</li> <li>Set a threshold .</li> <li>Configure a threshold for batchpredictions .</li> <li>Default numeric value for batch predictions .</li> <li>Fields Mapping for batch predictions .</li> <li>Fields Mapping for batch predictions .</li> <li>Single predictions view for classification ensembles .</li> <li>Single predictions view for regression ensembles .</li> <li>Single predictions .</li> <li>Single prediction .</li> <li< td=""><td>· · · · · ·</td><td></td><td><ul> <li>.</li> <li>.&lt;</li></ul></td><td>· · · · · · · · · · · · · · · · · · ·</td><td>139 140 141 142 142 143 145 146 146 146 147 148 148 149</td></li<></ul>	· · · · · ·		<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	139 140 141 142 142 143 145 146 146 146 147 148 148 149
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Single prediction</li> <li>Production</li> <li>Set the detailed explanation</li> <li>Download batch prediction output file</li> </ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	139 140 141 142 143 145 146 146 146 147 148 148 149 150
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single prediction</li> <li>Single prediction</li> <li>Single prediction</li> <li>Single prediction</li> <li>Single prediction</li> <li>Source for the test of test of the test of test of the test of /li></ul>			<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	139 140 141 142 142 143 145 146 146 146 147 148 148 149 150 150
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>View batch predictions output dataset</li> </ul>			<ul> <li>.</li> <li>.&lt;</li></ul>		139 140 141 142 142 143 145 146 146 147 148 148 149 150 150 151
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Yiew batch predictions output dataset</li> </ul>					139 140 141 142 142 143 145 146 146 147 148 148 149 150 151 151
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions 1-click actions</li> </ul>					139 140 141 142 142 143 145 146 146 147 148 148 149 150 151 151 151
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.87 2.88 2.89	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Single prediction</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions</li> <li>Edit predictions</li> </ul>					139 140 141 142 142 143 145 146 146 147 148 148 149 150 151 151 151 152 153
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.89 2.90	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> </ul>					139 140 141 142 143 145 146 146 146 147 148 148 149 150 150 151 151 152 153 154
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.85 2.86 2.87 2.88 2.89 2.90 2.91	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> </ul>					139 140 141 142 143 145 146 146 146 147 148 148 149 150 150 151 151 152 153 154 155
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.89 2.90 2.91 2.92	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single prediction view for regression Boosted Trees</li> <li>Explain prediction</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> <li>Private link of a prediction</li> </ul>					$\begin{array}{c} 139\\ 140\\ 141\\ 142\\ 142\\ 143\\ 145\\ 146\\ 146\\ 146\\ 147\\ 148\\ 149\\ 150\\ 150\\ 151\\ 151\\ 152\\ 153\\ 154\\ 155\\ 156\end{array}$
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.85 2.86 2.87 2.88 2.89 2.90 2.91 2.92 2.93	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Single prediction</li> <li>Input field importances</li> <li>See the detailed explanation</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction output file</li> <li>View batch predictions output dataset</li> <li>Batch predictions</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> <li>Private link of a prediction</li> </ul>					$\begin{array}{c} 139\\ 140\\ 141\\ 142\\ 142\\ 143\\ 145\\ 146\\ 146\\ 147\\ 148\\ 149\\ 150\\ 151\\ 151\\ 151\\ 152\\ 153\\ 154\\ 155\\ 156\\ 156\end{array}$
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.85 2.86 2.87 2.88 2.89 2.90 2.91 2.92 2.93 2.94	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> <li>Bet the detailed explanation</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction supput dataset</li> <li>Batch predictions</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> <li>Private link of a prediction</li> <li>Stop prediction from the 1-click menu</li> <li>Delete prediction from the 1-click menu</li> </ul>					$\begin{array}{c} 139\\ 140\\ 141\\ 142\\ 142\\ 143\\ 145\\ 146\\ 146\\ 147\\ 148\\ 149\\ 150\\ 151\\ 151\\ 151\\ 152\\ 153\\ 154\\ 155\\ 156\\ 156\\ 157\\ \end{array}$
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.85 2.86 2.87 2.88 2.89 2.90 2.91 2.92 2.93 2.94 2.95	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions .</li> <li>Select probability, confidence or votes .</li> <li>Select the positive class .</li> <li>Set a threshold .</li> <li>Configure a threshold for batchpredictions .</li> <li>Default numeric value for batch predictions .</li> <li>Fields Mapping for batch predictions .</li> <li>Output settings for batch predictions .</li> <li>Single predictions view for classification ensembles .</li> <li>Single predictions view for regression ensembles .</li> <li>Single predictions view for regression Boosted Trees .</li> <li>Explain prediction .</li> <li>Download batch prediction output file .</li> <li>An example of a batch prediction output file .</li> <li>Wiew batch predictions output dataset .</li> <li>Batch predictions .</li> <li>Markdown editor for evaluations descriptions .</li> <li>Private link of a prediction .</li> <li>Stop prediction from the 1-click menu .</li> <li>Delete prediction from popup menu .</li> </ul>					$\begin{array}{c} 139\\ 140\\ 141\\ 142\\ 142\\ 143\\ 145\\ 146\\ 146\\ 147\\ 148\\ 149\\ 150\\ 151\\ 151\\ 152\\ 153\\ 156\\ 156\\ 157\\ 157\\ \end{array}$
2.70 2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.87 2.88 2.89 2.90 2.91 2.92 2.93 2.94 2.95 2.96	<ul> <li>Probabilities, confidences or votes for Decision Forest batch predictions</li> <li>Select probability, confidence or votes</li> <li>Select the positive class</li> <li>Set a threshold</li> <li>Configure a threshold for batchpredictions</li> <li>Default numeric value for batch predictions</li> <li>Fields Mapping for batch predictions</li> <li>Output settings for batch predictions</li> <li>Single predictions view for classification ensembles</li> <li>Single predictions view for regression ensembles</li> <li>Single predictions view for regression Boosted Trees</li> <li>Explain prediction</li> <li>Bet the detailed explanation</li> <li>Download batch prediction output file</li> <li>An example of a batch prediction supput dataset</li> <li>Batch predictions</li> <li>Batch predictions</li> <li>Markdown editor for evaluations descriptions</li> <li>Private link of a prediction</li> <li>Stop prediction from the 1-click menu</li> <li>Delete prediction from the 1-click menu</li> </ul>					$\begin{array}{c} 139\\ 140\\ 141\\ 142\\ 142\\ 143\\ 145\\ 146\\ 146\\ 147\\ 148\\ 149\\ 150\\ 151\\ 151\\ 152\\ 153\\ 156\\ 156\\ 157\\ 157\\ 157\\ 157\end{array}$

2.99 Marl 2.100Men 2.101Ense 2.102Men 2.103Men 2.105Men 2.106Men 2.107Ense 2.108Ense	el to edit an ensemble's name, category, description and tags kdown editor for ensemble descriptions	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · ·	160 161 162 162 163 163 164 164 164
3.1 Line	ar regression list view					169
	oty Dashboard linear regression view					
	ar Regression icon					
	sing numeric coefficients at the end of linear regression table					
	ate 1-click linear regression from dataset 1-click action menu					
	ate 1-click linear regression from dataset popup menu					
	figure linear regression					
	figure the objective field to create the linear regression					
	nge the default objective field					
	matic optimization					
3.11 Con	figure the default numeric value	• •		• •	• •	1/6
	ning duration					
	ar regression candidates					
	ght options for linear regression					
	parameter					
	d coding configuration					
	ble field coding configuration					
	ect the dummy class					
	d codings: dummy					
	e modal window					
	d codings configured					
	ble field coding configuration					
3.24 Dum	nmy class in table view					185
	d coding configuration					186
	ble field coding configuration					
	d codings: contrast coding					
	the contrast coding values for each class					
	trast coding saved					
	e modal window					
	d codings configured					
	ble field coding configuration					
	trast modal window in table view					
	ble field coding configuration					
	d codings: other coding					
	the other coding values for each class					
	er coding saved					
	e modal window					
	d codings configured					
	ble field coding configuration					
	er coding in coefficients table					197
	er coding modal window					
3.45 Sam	pling parameters for linear regression	• •				199

3.46 Ordering argument for linear regression	. 200
3.47 Create linear regression after configuration	. 200
3.48 Linear regression API request preview	
3.49 Switch chart, PDP and tableviews	. 202
3.50 Linear regression chart parts	. 202
3.51 1D chart	. 203
3.52 Partial Dependence Plot	. 204
3.53 Prediction legend	. 205
3.54 Configure the values for other input fields	. 205
3.55 Reset the values for the input fields	
3.56 Export chart as image with or without legends	. 207
3.57 Table view for linear regression	. 208
3.58 Multiple field variables for categorical dummy encoded fields	
3.59 Missing numeric coefficients at the end of linear regression table	
3.60 Significant icon	. 210
3.61 Non-significant icon	
3.62 Select significance level	
3.63 Significance icons for coefficient estimates	
3.64 Summary of stats per coefficient	
3.65 Search and filter linear regression table	. 213
3.66 Export table in CSV file	
3.67 Predictions list view	
3.68 Empty predictions list view	
3.69 Menu options of the predictions list view	
3.70 Single predictions icon	
3.71 Batch predictions icon	
3.72 Predict using the 1-click action menu	
3.73 Predict using the pop up menu	
3.74 Linear regression prediction form	
3.75 Get the linear regression prediction	
3.76 Save your linear regression predictions	
3.77 Create batch prediction using 1-click action menu	
3.78 Create batch prediction using pop up menu	
3.79 Select dataset for batch prediction	
3.80 Configuration options for linear regression batch prediction	
3.81 Create a dataset from batch prediction	
3.82 Create batch prediction         3.83 Download batch prediction CSV file	
3.84 Batch prediction output dataset	
3.85 Configure Default numeric value for batch prediction	
3.86 Configure Default numeric value for batch prediction	
3.87 Configure the fields mapping for batch prediction	
3.88 linear regression output settings for batch predictions	
3.89 Prediction explanation	
3.90 Input field importances	
3.91 linear regression prediction descriptive information	
3.92 Markdown editor for linear regression prediction descriptions	
3.93 linear regression predictions privacy	
3.94 Stop linear regression batch prediction from 1-click action menu	
3.95 linear regression delete prediction confirmation	
3.96 Linear regression delete prediction from 1-click menu	. 234
3.97 Linear regression delete prediction from pop up menu	. 234
3.98 Linear regression delete prediction confirmation	
3.99 Click download icon	
3.100Select language to download linear regression	
3.101Warning message when the linear regression does not have any numeric input field	
3.102Warning message when the linear regression has more than 800 predictors	
3.103Warning message when the coefficient table limits are reached	. 238

3.104Edit linear regression descriptive information         3.105Markdown editor for linear regression descriptions         3.106Counters for linear regressions         3.107linear regression privacy         3.108Change project from 1-click action menu         3.109Change project from pop up menu         3.1010Stop linear regression creation from 1-click action menu         3.110Stop linear regression creation from pop up menu         3.112Confirmation message to delete a linear regression         3.113Delete linear regression from 1-click action menu         3.114Delete linear regression from pop up menu         3.115Confirmation message to delete a linear regression	239 240 241 242 242 242 243 243 243 244 244 244
4.1 Logistic regressions under Supervised tab	246
4.2 Logistic regression list view	
<ul><li>4.3 Empty Dashboard logistic regressions view</li><li>4.4 Logistic regression icon</li></ul>	
<ul><li>4.4 Logistic regression icon</li><li>4.5 Logistic regression formula</li></ul>	
4.6 Logistic regression formulas for two classes	
4.7 Missing numeric coefficients at the end of logistic regression table	
4.8 A dataset with images and image features	
4.9 A dataset with image feature fields shown	
4.10 Create 1-click logistic regression from dataset 1-click action menu	
4.11 Create 1-click logistic regression from dataset popup menu	
4.12 Configure logistic regression	253
4.13 Configure the objective field to create the logistic regression	
4.14 Change the default objective field	
4.15 Automatic optimization	
4.16 Configure the default numeric value and the missing numerics	
4.17 Training duration	
4.18 Logistic regression candidates         4.19 Weight options for logistic regression	
4.19 Weight options for logistic regression	
4.21 Include missing numeric values in your logistic regression	
4.22 Missing numeric coefficients at the end of logistic regression table	
4.23 Missing numerics is disabled when there is a Default numeric value selected	
4.24 Eps parameter for logistic regression	
4.25 Include statistics in logisticregression	
4.26 Check your categorical fields encoding by clicking in the green wheel icon	262
4.27 Bias parameter and auto-scaling parameter for numeric fields	
4.28 Regularization parameters	
4.29 Field coding configuration	
4.30 Enable field coding configuration	
4.31 Select the dummy class	
4.32 Field codings: dummy       4.33 Close modal window	
4.35 Close modal window	
4.35 Disable field coding configuration	
4.36 Dummy class in table view	
4.37 Field coding configuration	
4.38 Enable field coding configuration	
4.39 Field codings: contrast coding	272
4.40 Set the contrast coding values for each class	
4.41 Contrast coding saved	
4.42 Close modal window	
4.43 Field codings configured	
4.44 Disable field coding configuration         4.45 Contrast icon in table view	
	210

4.46 Contrast modal window in table view	
4.47 Field coding configuration	
4.48 Enable field coding configuration	
4.49 Field codings: other coding	
4.50 Set the other coding values for each class	
4.51 Other coding saved	
4.52 Close modal window	
4.53 Field codings configured	
4.54 Disable field coding configuration	
4.55 Other coding in coefficients table	
4.56 Other coding modal window	
4.57 Sampling parameters for logistic regression	
4.58 Create logistic regression after configuration	
4.59 Logistic regression API request preview	
4.60 Switch chart and table views	
4.61 Logistic regression chart parts	
4.62 1D chart	
4.63 2D chart	
4.64 Prediction legend	
<ul><li>4.65 Configure the values for other input fields</li><li>4.66 Reset the values for the input fields</li></ul>	
4.67 Export chart as image with or without legends	
4.68 Table view for logistic regression	
4.69 Multiple field variables for categorical one-hot encoded fields	
4.70 Missing numeric coefficients at the end of logistic regression table	
4.71 Search and filter logistic regression table	
4.72 Export table in CSV file	
4.73 Likelihood ratio	
4.74 Significant likelihood ratio	
4.75 Select significance level	
4.76 Likelihood ratio <i>p</i> -value	
4.77 Significant icon	
4.78 Non-significant icon	
4.79 Significance icons for coefficient estimates	
4.80 Summary of stats per coefficient	
4.81 Predictions list view	
4.82 Empty predictions list view .	
4.83 Menu options of the predictions list view	
4.84 Single predictions icon	
<ul><li>4.85 Batch predictions icon</li><li>4.86 Predict using the 1-click action menu</li></ul>	
<ul><li>4.87 Predict using the pop up menu</li><li>4.88 Logistic regression prediction form</li></ul>	
4.89 Logistic regression predictions form	
4.09 Get the logistic regression prediction	
4.90 Get the logistic regression prediction	
4.92 Save your logistic regression predictions	
4.93 Select a single image source in the image input field	
4.95 Select a single image source in the image input held	
4.95 Select a component of a composite source	
4.96 Logistic regression image prediction form, more fields	
4.97 Logistic regression image single prediction	
<ul><li>4.97 Logistic regression image single prediction</li><li>4.98 Create batch prediction using 1-click action menu</li></ul>	
<ul> <li>4.97 Logistic regression image single prediction</li> <li>4.98 Create batch prediction using 1-click action menu</li> <li>4.99 Create batch prediction using pop up menu</li> </ul>	311
<ul> <li>4.97 Logistic regression image single prediction</li> <li>4.98 Create batch prediction using 1-click action menu</li> <li>4.99 Create batch prediction using pop up menu</li> <li>4.100Select dataset for batch prediction</li> </ul>	311 311
<ul> <li>4.97 Logistic regression image single prediction</li> <li>4.98 Create batch prediction using 1-click action menu</li> <li>4.99 Create batch prediction using pop up menu</li> <li>4.100Select dataset for batch prediction</li> <li>4.101Configuration options for logistic regression batch prediction</li> </ul>	311 311 312
<ul> <li>4.97 Logistic regression image single prediction</li> <li>4.98 Create batch prediction using 1-click action menu</li> <li>4.99 Create batch prediction using pop up menu</li> <li>4.100Select dataset for batch prediction</li> </ul>	311 311 312 313

4.104Download batch prediction CSV file	. 314
4.10 Batch prediction output dataset	
4.10@Batch prediction using an image dataset	. 315
4.107Select the positive class for single predictions	
4.108Set the probability threshold for single predictions	
4.109Set the probability threshold for batch predictions	
4.110Configure Default numeric value for batch prediction	. 318
4.111Configure the fields mapping for batch prediction	. 319
4.112 ogistic regression output settings for batch predictions	
4.113Logistic regression single prediction	
4.114Logistic regression all class probabilities	
4.115Logistic regression export all class probabilities histogram	
4.116Logistic regression probability threshold	
4.117 Explain prediction	
4.118nput field importances	
4.119See the detailed explanation	
4.120Download batch prediction CSV file	
4.121An example of a logistic regression batch prediction CSV file	
4.122Batch prediction output dataset	
4.123Logistic regression batch prediction output dataset	
4.124Logistic regression prediction descriptive information	
4.125Markdown editor for logistic regression descriptions	
<ul><li>4.126Logistic regression predictions privacy</li><li>4.127Stop logistic regression batch prediction from 1-click action menu</li></ul>	
4.128 Logistic regression delete prediction confirmation	
4.129_ogistic regression delete prediction from 1-click menu	
4.130Logistic regression delete prediction from pop up menu	
4.131Logistic regression delete prediction confirmation	
4.132Click download icon	
4.133Select language to download logistic regression	
4.134Warning message when the logistic regression does not have any numeric field .	
4.135Warning message when the logistic regression has more than 100 fields	
4.136Warning message when the objective field has more than 200 classes	
4.137Warning message when the table limits are reached	. 337
4.138 Edit logistic regression descriptive information	
4.139Markdown editor for logistic regression descriptions	
4.140Counters for logistic regressions	
4.141Logistic regression privacy	. 340
4.142Change project from 1-click action menu	. 340
4.143Change project from pop up menu	
4.144Stop logistic regression creation from 1-click action menu	
4.145Stop logistic regression creation from pop up menu	
4.146Confirmation message to delete a logistic regression	
4.147Delete logistic regression from 1-click action menu	
4.148Delete logistic regression from pop up menu	
4.149Confirmation message to delete a logistic regression	
4.150Logistic Regression Workflow	. 345
5.1 Deepnet list view	347
5.2 Empty Dashboard deepnet view	
5.3 Deepnet icon	
5.4 Create 1-click deepnet from dataset 1-click action menu	
5.5 Create 1-click deepnet from dataset popup menu	
5.6 Configure deepnet	
5.7 Configure the objective field to create the deepnet	
5.8 Change the default objective field	
5.9 Select an automatic optimization option	

5.10	Disable automatic optimization options to manually configure the rest of the network pa-	
	rameters	354
	Select a default numeric value to replace missing numeric values	
	Include missing numeric values in your deepnet	
	Set the training duration for a deepnet	
5.14	Set the maximum number of iterations for a deepnet	356
5.15	Configure the hidden layers of the network	357
5.16	Enable or disable the residuals learning	358
5.17	Enable or disable the batch normalization	359
5.18	Enable or disable the tree embedding	360
5.19	Momentum algorithm	361
5.20	Adagrad algorithm	362
5.21	RMSProp algorithm	363
5.22	Adam algorithm	364
	FTRL algorithm	
	Configure the learning rate	
	Configure the dropout rate	
	Set a seed for the deepnet	
	Weight options for deepnets	
	Sampling parameters for deepnet	
	Create deepnet after configuration	
	Deepnet API request preview	
	Deepnet chart parts	
	Deepnet chart	
	Prediction legend	
	Configure the values for other input fields	
	Reset the values for the input fields	
	Export chart as image with or without legends	
	Image Deepnet Page Classification layout	
	Image Deepnet Page Classification Performance Panel	
	Image Deepnet Page Classification Image Results	
	Image Deepnet Page Classification Class List	
	Image Deepnet Page Classification filtering by probability	
	Image Deepnet Page Classification showing one class	
	Image Deepnet Page Regression layout	
	Image Deepnet Page Regression Performance Panel	
	Image Deepnet Page Regression Display Error Slider	
	Image Deepnet Page Regression Split Error Slider	
	Image Deepnet Page Regression Image Results	
	Field importance for deepnets	
	View networks' configuration	
		392
		392
		392
	- <b>3</b> - 1	393
	Batch predictions icon	
	Predict using the 1-click action menu	
	Predict using the pop up menu	
	Deepnet prediction form	
	Deepnet predictions form	
	Get the deepnet prediction	
	Display all class probabilities	
		397
	Predict using the pop-up menu	
	Select a single image source in the image input field	
	List the components of a composite source	
	Select a component of a composite source	
5.66	Deepnet image single prediction	400

<ul> <li>5.67 Deepnet image prediction with more than seven classes</li></ul>	402
5.70 Create batch prediction using pop up menu	403
5.71 Select dataset for batch prediction	
5.72 Configuration options for deepnet batch prediction	
5.74 Create batch prediction	405
5.75 Download batch prediction CSV file	
5.76 Batch prediction output dataset	
5.77 Batch prediction using an image dataset	
5.79 Set a probability threshold	
5.80 Save the prediction	
5.81 Configure a probability threhsold for batch predictions	
5.82 Configure Default numeric value for batch prediction	
<ul><li>5.83 Configure the fields mapping for batch prediction</li></ul>	
5.85 Deepnet single prediction	
5.86 Deepnet all class probabilities	
5.87 Explain prediction	
5.88 Input field importances	
5.89 See the detailed explanation	
5.90 Download batch prediction CSV file	
<ul><li>5.91 An example of a deepnet batch prediction CSV file</li></ul>	
5.92 Deepnet batch prediction output dataset	
5.94 Deepnet prediction descriptive information	
5.95 Markdown editor for deepnet descriptions	
5.96 Deepnet predictions privacy	
5.97 Stop deepnet batch prediction from 1-click action menu	
5.98 Deepnet delete prediction confirmation	
5.99 Deepnet delete prediction from 1-click menu	
5.101Deepnet delete prediction confirmation	
5.102Click download icon	
5.103Select language to download deepnet	
5.104Edit deepnet descriptive information	
5.105Markdown editor for deepnet descriptions	
5.106Counters for deepnet	
5.108Change project from 1-click action menu	
5.109Change project from pop up menu	
5.110Stop deepnet creation from 1-click action menu	
5.111Stop deepnet creation from pop up menu	
5.112Confirmation message to delete a deepnet	
5.113Delete deepnet from 1-click action menu	
5.114Delete deepnet from pop up menu	
5.11@Deepnet Workflow	
6.1 Fusion list view	
6.2 Empty Dashboard fusion view	
<ul><li>6.3 Fusion icon</li></ul>	
6.5 Create fusion from pop up menu	
6.6 Create fusion from model list views	
6.7 Select models with the same objective field	438

	Select models from the OptiML table	
	Create fusion using another fusion	
	Fusion creation view	
	Search for models	
	Selected models	
	The model search will be filtered by the objective field name	
	Remove the objective field filter to select models with different objective field names	
	Select which objective field you want to use to select more models	
	Assign weights to the models	
	Create fusion after configuration	
	Fusion API request preview	
	Switch from PDP to model list views	
	Fusions chart view parts	
	Select fields for your fusion PDP axis	
	Prediction legend	
	Configure the values for other input fields	
	Reset the values for the input fields	
	Export chart as image with or without legends	
	Fusion model list view	
	Predictions list view	
	Empty predictions list view	
	Menu options of the predictions list view	
	Single predictions icon	
	Batch predictions icon	
	Predict using the 1-click action menu	
	Predict using the pop up menu	
	Fusion prediction form	
	Fusion prediction	
	Create batch prediction using 1-click action menu	
	Create batch prediction using pop up menu	
	Select dataset for batch prediction	
	Configuration options for fusion batch prediction	
	Create a dataset from batch prediction	
	Download batch prediction CSV file and output dataset	
	Missing strategies for fusion single predictions	
	Missing strategies for fusion batch predictions	
	Select the positive class	
	Set a probability threshold	
	Get the prediction	
	Configure a probability threhsold for batch predictions	
	Configure Default numeric value for batch prediction	
	Exclude fields from the fusion prediction	
	Configure the fields mapping for batch prediction	
	Fusion output settings for batch predictions	
	Fusion single prediction	
	Fusion all class probabilities	
	Explain prediction	
	Input field importances	
	See the detailed explanation	
	Download batch prediction CSV file	
	An example of a fusion batch prediction CSV file	
	Batch prediction output dataset	
	Fusion batch prediction output dataset	
	Fusion prediction descriptive information	
	Markdown editor for fusion descriptions	
	Fusion predictions privacy	
	Stop fusion batch prediction from 1-click action menu	
6.65	Fusion delete prediction confirmation	478

6.67 6.68 6.69 6.70	Delete fusion prediction from 1-click menuDelete fusion prediction from pop up menuDelete fusion prediction confirmationClick download iconSelect language to download your fusionsThe PDP cannot be displayed	· · · · · ·	  	. 479 . 480 . 480 . 481
	The PDP cannot be displayed because the predictions take too much time to be of			
	Edit fusion descriptive information			
	Markdown editor for fusion descriptions			
	Counters for fusions			
	Fusions privacy			
	Change project from 1-click action menu			
	Change project from pop up menu			
	Stop fusion creation from 1-click action menu			
	Stop fusion creation from pop up menu			
	Confirmation message to delete a fusion			
	Delete fusions from 1-click action menu			
	Delete fusion from pop up menu			
	Confirmation message to delete a fusion			
0.85	Fusion Workflow	• •	• •	. 493
7.1	Evaluations section in the BigML Dashboard			. 495
7.2	Evaluation list view			
7.3	Evaluation list view 1-click action menu			
7.4	Evaluations icon			
7.5	Cross-validation icon			
7.6	Classification evaluations icon			. 497
7.7	Regression evaluations icon			. 497
7.8	Classification cross-validation icon			
7.9	Regression cross-validation icon			
	Confusion Matrix example			
	Cell colors indicate the class selected as positive			
	Accuracy example			
	Precision example			
	Recall example			
	F-measure example			
	Phi Coefficient example			. 501
	Macro-averaged measures			
	Kendall's Tau-b coefficient			
	Spearman's Rho coefficient			
	Select the probability threshold and the positive class			
	Probability threshold of 100%			
	Probability threshold of 0%			
	Area Under the Curve for the Precision-Recall curve			
7.25	Area Under the Convex Hull for the Precision-Recall curve			. 510
	The ROC curve			
7.27	The ROC AUC			. 512
	The ROC AUCH			
	The Gain curve			
	The Lift curve			
	Evaluate model from evaluation list view			
	Select model and dataset			
	Evaluate model from 1-click action menu.			
	Evaluate model from pop up menu			
	Evaluate ensemble from evaluation list view			
	Select ensemble and dataset			
1.01		•••	• •	. 520

7 38	Evaluate ensemble from 1-click action menu	520
	Evaluate ensemble from pop up menu	
	Evaluation with pre-filled ensemble and dataset information	
7.41	Evaluate logistic regression from evaluation list view	. 521
7.42	Select logistic regression and dataset	. 522
	Evaluate logistic regression from 1-click action menu	
	Evaluate logistic regression from pop up menu	
	Evaluation with pre-filled logistic regression and dataset information	
	Evaluate deepnet from evaluation list view	
	Select deepnet and dataset	
	Evaluate deepnet from 1-click action menu	
7.49	Evaluate deepnet from pop up menu	. 525
7.50	Evaluation with pre-filled deepnet and dataset information	. 525
	Evaluate fusion from evaluation list view	
	Select fusion and dataset	
	Evaluate fusion from 1-click action menu	
	Evaluate fusion from pop up menu	
1.55	Evaluate a fusion using a testing dataset	. 528
	Basic 5-fold cross-validation	
7.57	Model's k-fold cross-validation	. 529
7.58	Ensemble's k-fold cross-validation	. 530
	Logistic regression's k-fold cross-validation	
	Deepnet's k-fold cross-validation	
	Cross-validation script view	
	Clone script from preview	
	Clone script from script view	
	Confirmation message to clone script	
7.65	Configure cross-validation inputs	. 534
	Execute cross-validation script	
	Cross-validation execution progress	
	Cross-validation output and resources	
	Cross-validation script in scripts list view	
	Evaluation configuration panel	
	Last prediction icon	
	Proportional missing strategy icon	
	Missing strategies for evaluations	
7.74	Select the voting strategy to calculate the evaluation	. 539
	Fields Mapping for evaluations	
	Sampling options for evaluations	
	Select a dataset to execute a basic 5-fold cross-validation	
	Model's k-fold cross-validation configuration	
	Ensemble's k-fold cross-validation configuration	
	Example of an ensemble evaluation top view information	
7.81	Confusion matrix view	. 547
7.82	Select a positive class	. 548
7.83	Performance metrics and benchmark models view	. 549
	Download confusion matrix	
	Select the evaluation curve, the positive class and the threshold	
	Export options for the threshold confusion matrices and the evaluation curves	
	Performance measures for regression models, ensembles, deepnets, and fusions	
	Compare regression models against random and mode values	
	Cross-validation view for classification measures	
	Cross-validation single evaluations	
7.91	Compare evaluations using 1-click action menu	. 554
	Compare evaluations using pop up menu	
	Compare evaluations using 1-click action menu from list view	
	Select the other evaluation	
	Compare evaluations side by side	
1.30		. 556

7.96 Compare multiple evaluations using 1-click action menu	557
7.97 Compare multiple evaluations using 1-click action menu from the list view	557
7.98 Select an evaluation curve	558
7.99 Select evaluations to compare	559
7.100Manage evaluations from the legend	560
7.101See the PR AUCH and ROC AUCH by hovering over the evaluation name in the table	
7.102See the K-S statistic by hovering over the evaluation name in the table	
7.103Table containing all the evaluations and the model parameters	
7.104 Export multiple comparison chart in PNG format	
7.105Panel to edit an evaluation name, description, category, and tags	
7.106Markdown editor for evaluations descriptions	
7.107Share evaluations	
7.108Move evaluation from the 1-click action menu	
7.109Move evaluation from the pop up menu	
7.110Stop evaluation creation	
7.111Stop evaluation from popu up menu	
7.112Delete evaluation from the 1-click action menu	
7.113Delete evaluation from popu up menu	
7.114Delete evaluation confirmation	
7.115Single evaluations workflow	
7.116Cross-validation workflow	572

# **List of Tables**

1.1 1.2 1.3	Weight Field example for transactional dataset       17         Categories used to classify predictions by BigML       66         Categories used to classify models by BigML       73
2.1 2.2 2.3 2.4 2.5 2.6	Weight Field example for transactional dataset       102         Example of classification ensemble using probability       136         Example of regression ensemble using probability       137         Example of classification ensemble using votes       137         Categories used to classify predictions by BigML       155         Categories used to classify ensembles by BigML       161
3.1 3.2 3.3 3.4 3.5	Number of predictors per input field172Dummy coding example for 3 classes180Contrast coding example for 3 classes185Other coding192Categories used to classify linear regression by BigML240
4.1 4.2 4.3 4.4 4.5	One-hot coding265Dummy coding example for 3 classes265Contrast coding example for 3 classes270Other coding276Categories used to classify logistic regression by BigML339
5.1	Categories used to classify deepnet by BigML
6.1	Categories used to classify fusions on BigML
7.1 7.2	Example of diabetes prediction probabilities for three different patients

# Glossary

- **Bagging** an ensemble based algorithm that uses a random subset of instances to generate each single tree. 96
- **BigML Gallery** a section of BigML to share, buy or sell datasets, models, and scripts. Go to Gallery. 74, 77, 84, 528
- **Classification** a modeling task whose objective field (i.e., the field being predicted) is categorical and predicts classes. ii, 1, 3, 23, 25, 83, 85, 98, 165, 168, 246, 346, 347, 357, 391, 434, 436, 452, 494, 498, 556
- **Clustering** an unsupervised Machine Learning task in which dataset instances are grouped into geometrically related subsets. 10
- **Confidence** an indicator of the prediction's certainty for classification models and ensembles. It takes into account the class distribution and the number of instances at a certain node. It is a value between 0% and 100%. 35, 83, 119, 166
- **Dashboard** The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. ii, 1, 35, 77, 86, 120, 168, 214, 219, 246, 298, 305, 346, 391, 397, 434, 452, 456, 494
- **Dataset** the structured version of a BigML source. It is used as input to build your predictive models. For each field in your dataset a number of basic statistics (min, max, mean, etc.) are parsed and produced as output. 1, 48, 132, 221, 312, 404, 458, 494
- **Decision Trees** a class of Machine Learning algorithms used to solve regression and classification problems. Decision trees are composed of nodes and branches that create a model of decisions with a tree graph. Nodes represent the predictors or labels that have an influence in the predictive path, and the branches represent the rules followed by the algorithm to make a given prediction. 1
- **Deepnets** an optimized implementation of deep neural networks, a class of supervised learning algorithms, that can be used to solve regression and classification problems. The input features are fed to one or several groups "nodes", each group of nodes form a "layer". Each node is essentially a function on the input that transforms the input features into another value or collection of values. This process continues layer by layer, until we reach the final output (prediction), an array of perclass probabilities forclassification problems or a single, real value for regression problems. ii, 168, 434, 435, 440, 494
- **Early split** when the dataset is bigger than 34 GB, BigML automatically takes a sample of your data and performs an early split so the model creation becomes significantly faster. It detects when an early split is safe by calculating the summary statistics collected at each node. Early splitting requires that the training data is shuffled beforehand to avoid generating inaccurate models caused by ordered fields in the input rows (as it will process the first x instances, then the next x ones and so on). 3, 19, 106, 199

- **Ensembles** a class of Machine Learning algorithms in which multiple independent classifiers or regressors are trained, and the combination of these classifiers is used to predict an objective field. An ensemble of models built on samples of the data can become a powerful predictor by averaging away the errors of each individual model. ii, 119, 165, 168, 246, 346, 434, 435, 440, 494
- **Expected error** an indicator of the prediction's certainty for regression models and ensembles. It is the average of the predictions errors at a certain node. 7, 35, 83, 119, 166
- Field an attribute of each instance in your data. Also called "feature", "covariate", or "predictor". Each field is associated with a type (numeric, categorical, text, items, or date-time). 1, 25, 85, 110, 246, 346, 434
- **Field importance** a measure of each field importance for predicting the objective field relative to the other fields. It is computed averaging the error each field helps to reduce at every tree split. 39, 122
- **Fusions** supervised model that solves classification and regression problems by averaging the predictions of multiple models, ensembles, logistic regressions, and/or deepnets. Fusions are based on the same "wisdom of the crowds" principle than ensembles under which the combination of multiple models is often more performant than any of its individual members.. ii, 434, 440, 494
- **Histogram** a bar chart-style visualization of a collection of values, in which the range of the values is broken up into a collection of ranges, and the height of a given bar increases as more points fall into the range associated with that bar. 4
- **Instances** the data points that represent the entity you want to model, also known as observations or examples. They are usually the rows in your data with a value (potentially missing) for each field that describes the entity. 1
- Leaf a terminal node in a tree, i.e., a node that has no children. 3, 21, 593
- Linear regression a popular technique from the fields of statistics that has been borrowed by Machine Learning to solve regression problems. Linear regression assumes the output variable, or the objective field, is a function of linear combination of the inputs. ii
- Local predictions the predictions made in your local environment, faster, at no cost, by downloading your model. 41, 69, 126, 158, 217, 230, 302, 324
- **Logistic regression** another technique from the fields of statistics that has been borrowed by Machine Learning to solve classification problems. For each class of the objective field, logistic regression fits a logistic function to the training data. Logistic regression is a linear model, in the sense that it assumes the probability of a given class is a function of a weighted combination of the inputs. ii, 346, 434, 435, 440, 494
- **Missing value** the data points that represent the entity you want to model may present missing value, i.e., not provide a value for all fields that compose the entity. 15, 99
- **Model** a single decision tree-like model when we refer to it in particular, and a predictive model when we refer to it in general. ii, 35, 83, 168, 246, 346, 434, 435, 440, 494

Node threshold the maximum number of nodes that a BigML model is allowed to grow. 15, 99

- **Non-preferred fields** fields that, for a number of possible reasons, are by default not included in the modeling process. One example of this is fields that contain the same value for every instance; in general, constant fields add no information to the modeling process. 11, 92
- **Objective Field** the field that a regression or classification model will predict (also known as target). 1, 12, 15, 83, 85, 86, 93, 101, 109, 165, 168, 173, 174, 201, 245, 246, 247, 252, 253, 284, 326, 344, 346, 351, 373, 416, 432, 434, 436, 441, 445, 473, 492, 494, 498, 542, 543

- **OptiML** an automated optimization process for model selection and parametrization (or hyperparametrization) to solve classification and regression problems. 13, 94, 176, 255, 438, 440
- **Organization** a collaborative workspace where all the users in the organization can access, work on, and visualize the same projects and resources in the BigML Dashboard. Furthermore, organizations enable you to define different roles and permissions for each user involved on your Machine Learning projects. 488
- **Orthogonal** the default "one-hot" coding is orthogonal since a single instance can't be two categories at the same time, so after recoding we also need to ensure there are not co-dependent coefficients. Orthogonality is met when the dot product for the codings equals 0, e.g., the following codings are orthogonal [1, 1, -1, -1], [-1, 1, 0, 0] since (1) * (-1) + (1) * (1) + (-1) * (0) + (-1) * (0) = 0. 187, 272
- **Overfitting** the process of tailoring the model to fit the training data at the expense of generalization. 5, 14, 15, 85, 97, 99, 264, 357, 364, 365, 366, 511, 516, 542, 543
- Predicate a predicate is a statement that can be either true or false depending on the values of its component variables. BigML predicates may use the boolean operators =, <=, >=, <, >, in. Example of predicates are balance < 1,000 and field x = "category". 3</p>
- **Predicting** the result of obtaining the objective field value for your new data using an existing model. The model returns the predicted value along with a performance measure (confidence for classification or expected error for regression). 35, 119, 214, 245, 298, 344, 391, 432, 452, 492, 494
- Prediction Path the series of rules that lead to a certain node in a decision tree. 22
- Predictors the fields your model uses as inputs to generate the set of rules to make predictions. 109
- **Project** an abstract resource that helps you group related BigML resources together. 2, 79, 86, 162, 214, 241, 247, 299, 340, 347, 392, 428, 435, 453, 487, 496
- Random Decision Forests an ensemble based algorithm which uses a random subset of features to generate anomaly scores. 5, 96
- **Regression** a modeling task whose objective field (i.e., the field being predicted) is numeric. ii, 1, 3, 23, 25, 83, 85, 98, 165, 168, 346, 347, 357, 391, 434, 436, 452, 494, 498
- **Resource** any of the Machine Learning objects provided by BigML that can be used as a building block in the workflows needed to solve Machine Learning problems. 494
- Root the node from which a tree originates. 21, 593

Sampling the process of partitioning your dataset to consider just a subset of your instances. 17, 104

- **Supervised learning** a type of Machine Learning problem in which each instance of the data has a label. The label for each instance is provided in the training data, and a supervised Machine Learning algorithm learns a function or model that will predict the label given all other features in the data. The function can then be applied to data unseen during training to predict the label for unlabeled instances. ii, 1, 85, 168, 169, 246, 247, 346, 347, 434, 436, 494
- **Support** the proportion of instances in the dataset which contain an itemset. The support of an association is the portion of instances in the dataset which contain the rule's antecedent and rule's consequent together over the total number of instances (N) in the dataset. 23
- **Task** the process of creating a BigML resource, such as creating a dataset, or training a model. A given task can also create subtasks, as, in the case of a WhizzML script that contains calls to create other resources. 79
- **Time series** a sequentially indexed representation of your historical data that can be used to forecasting future values of numerical properties. BigML implements exponential smoothing where the smoothing parameters assign exponentially increasing weights to most recent instances. Exponential smoothing methods allow the modelization of data with trend and seasonal patterns. ii

- **Tree** a data structure that can be described as a collection of nodes, starting at a root node, where each node may recursively have a number of child nodes. Nodes that have no childrens are called leaves. 3, 21, 591, 592
- **Unsupervised learning** a type of Machine Learning problem in which the objective is not to learn a predictor, and thus does not require each instance to be labeled. Typically, unsupervised learning algorithms infer some summarizing structure over the dataset, such as a clustering or a set of association rules. ii
- **WhizzML** BigML domain-specific language for automating complex Machine Learning workflows and implementing high-level algorithms. <u>528</u>

## References

- [1] Leo Breiman and Adele Cutler. *Random Forests*. June 2004. URL: http://www.stat.berkeley. edu/~breiman/RandomForests/cc_home.htm#giniimp.
- [2] Thomas G. Dietterich. Ensemble Methods in Machine Learning. Tech. rep. Oregon State University, Dec. 2000. URL: http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf.
- [3] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. Dec. 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.
- [4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. "Evaluation of text classification". In: Introduction to Information Retrieval. Cambridge University Press, 2008. Chap. 13.6, pp. 258–263. URL: http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-oftext-classification-1.html.
- [5] Charles Parker. An Analysis of Performance Measures For Binary Classifiers. Tech. rep. BigML, Inc., 2011. URL: http://www.clparker.org/parker-measure.pdf.
- [6] The BigML Team. Anomaly Detection with the BigML Dashboard. Tech. rep. BigML, Inc., Jan. 2016.
- [7] The BigML Team. Association Discovery with the BigML Dashboard. Tech. rep. BigML, Inc., Dec. 2015.
- [8] The BigML Team. *Cluster Analysis with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.
- [9] The BigML Team. *Datasets with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.
- [10] The BigML Team. OptiML with the BigML Dashboard. Tech. rep. BigML, Inc., May 2018.
- [11] The BigML Team. Sources with the BigML Dashboard. Tech. rep. BigML, Inc., Jan. 2016.
- [12] The BigML Team. *Time Series with the BigML Dashboard*. Tech. rep. BigML, Inc., July 2017.
- [13] The BigML Team. Topic Models with the BigML Dashboard. Tech. rep. BigML, Inc., Nov. 2016.
- [14] B. E. T. H. Twala, M. C. Jones, and D. J Hand. "Good methods for coping with missing data in decision trees." In: *Pattern Recognition Letters* 29.7 (2008), pp. 950–956. DOI: http://oro.open. ac.uk/22531/1/decision_trees.pdf.

