OptiML with the BigML Dashboard

The BigML Team

Version 1.1

MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

About this Document

This document provides a comprehensive description of OptiML, an automated optimization process for supervised model selection and parametrization to solve classification and regression problems.

This document assumes that you are familiar with:

- Sources with the BigML Dashboard. The BigML Team. June 2016. [6]
- Datasets with the BigML Dashboard. The BigML Team. June 2016. [5]

To learn how to use the BigML Dashboard to build supervised predictive models read:

- Classification and Regression with the BigML Dashboard. [3]

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. [7]
- Cluster Analysis with the BigML Dashboard. The BigML Team. June 2016. [4]
- Anomaly Detection with the BigML Dashboard. The BigML Team. June 2016. [1]
- Association Discovery with the BigML Dashboard. The BigML Team. June 2016. [2]
- Topic Modeling with the BigML Dashboard. The BigML Team. November 2016. [8]

Contents

CHAPTER 1

Introduction

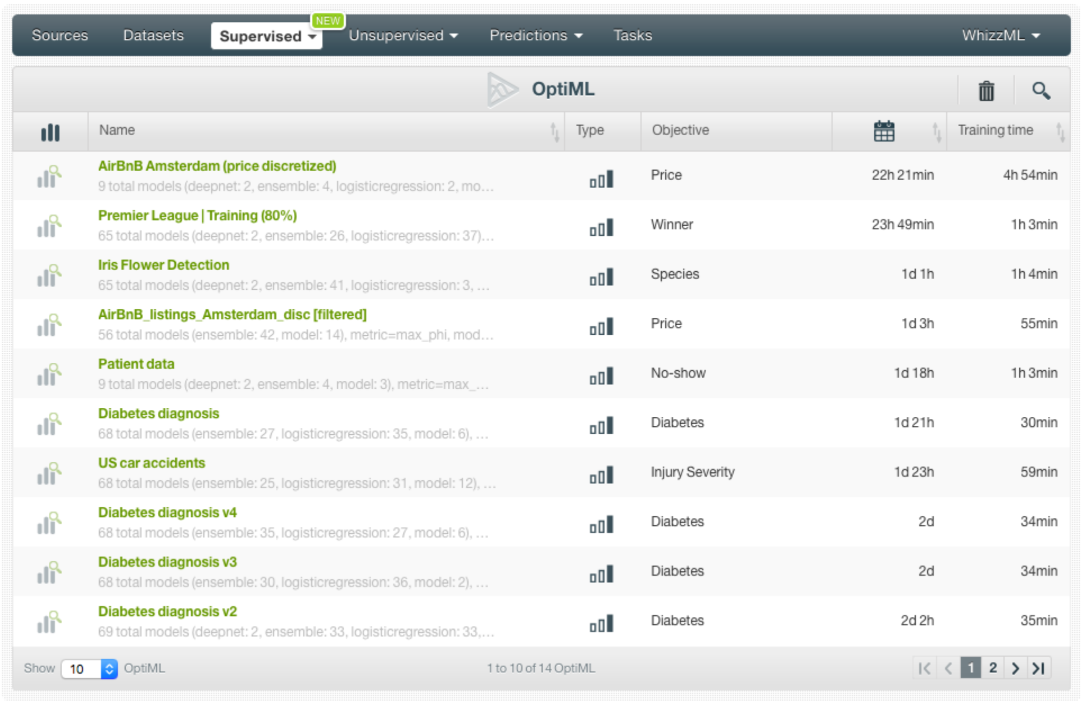
OptiML is an **automated optimization process** for **model selection and parametrization** (or hyper-parametrization) to solve classification and regression problems. Selecting the right algorithm and its optimum parameter values is a manual and time-consuming task for any Machine Learning practitioner. This iterative process is currently based on trial and error (creating and evaluating different models to find the best one) and it requires a high level of expertise and intuition. OptiML accelerates the process of model search and parameter tuning, allowing non-experts to build top-performing models.

OptiML uses Bayesian parameter optimization for the model selection and parameter tuning. It sequentially tries groups of parameters training and evaluating models using them, and based on the results, it tries a new group of parameters. The model search is guided by an optimization metric that can be configured by the user. When the process finishes, a list of the top performing models is returned in an interface to compare them and select the one that best fits your needs.

This document provides a comprehensive description of OptiML, including how it can be created with 1-click (Chapter 3), how to configure the different parameters (Chapter 4), and the visualization of the OptiML result (Chapter 5). You can also create, configure, retrieve, list, update, and delete your OptiML using the BigML API and bindings (Section 6.1 and Section 6.2).

The OPTIML section within the "Supervised" tab of the main menu of the BigML Dashboard allows you to list all your available OptiML resources. The OptiML list view (Figure 1.1), shows the dataset used to create each OptiML, the **Name** and the parameters configured to create the OptiML, the **Objective field** (explained in Section 4.1), the **Age** (time elapsed since it was created), and the **Training time** (time to create it). The SEARCH menu option in the top right menu of the OptiML list view allows you to **search** your OptiML by name or ID[1] (using the syntax "id:" followed by the OptiML ID). You can also search an OptiML by the parameters used to create it by typing in the search box the syntax "config:" followed by the parameters you are interested in.

[1] `https://support.bigml.com/hc/en-us/articles/360000029074-How-do-I-refer-to-my-resources-in-BigML-`



Name	Type	Objective		Training time
AirBnB Amsterdam (price discretized) 9 total models (deepnet: 2, ensemble: 4, logisticregression: 2, mo…		Price	22h 21min	4h 54min
Premier League | Training (80%) 65 total models (deepnet: 2, ensemble: 26, logisticregression: 37)…		Winner	23h 49min	1h 3min
Iris Flower Detection 65 total models (deepnet: 2, ensemble: 41, logisticregression: 3, …		Species	1d 1h	1h 4min
AirBnB_listings_Amsterdam_disc [filtered] 56 total models (ensemble: 42, model: 14), metric=max_phi, mod…		Price	1d 3h	55min
Patient data 9 total models (deepnet: 2, ensemble: 4, model: 3), metric=max_…		No-show	1d 18h	1h 3min
Diabetes diagnosis 68 total models (ensemble: 27, logisticregression: 35, model: 6), …		Diabetes	1d 21h	30min
US car accidents 68 total models (ensemble: 25, logisticregression: 31, model: 12), …		Injury Severity	1d 23h	59min
Diabetes diagnosis v4 68 total models (ensemble: 35, logisticregression: 27, model: 6), …		Diabetes	2d	34min
Diabetes diagnosis v3 68 total models (ensemble: 30, logisticregression: 36, model: 2), …		Diabetes	2d	34min
Diabetes diagnosis v2 69 total models (deepnet: 2, ensemble: 33, logisticregression: 33,…		Diabetes	2d 2h	35min

Show 10 OptiML 1 to 10 of 14 OptiML 1 2

Figure 1.1: OptiML list view

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

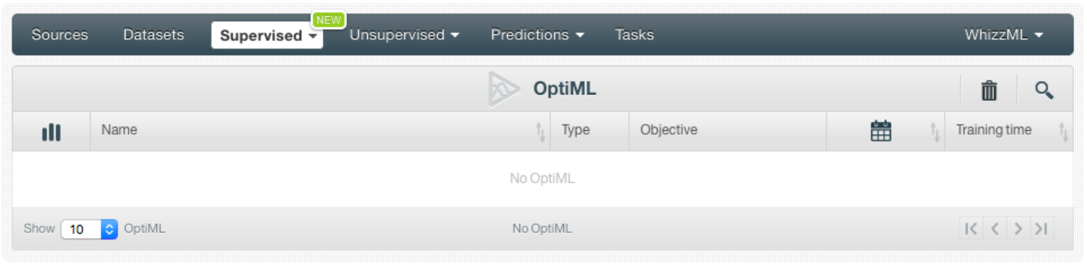


Figure 1.2: OptiML empty list view in the BigML Dashboard

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

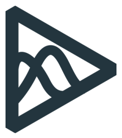


Figure 1.3: OptiML icon

CHAPTER 2

Understanding OptiML

OptiML is an **automated optimization process** for **model selection and parametrization** to solve classification and regression problems.

Selecting the right algorithm and its optimal parameter values is currently a manual and time-consuming process which requires a high level of expertise. This is because different supervised learning algorithms can be used to solve similar problems and it is very difficult to know in advance which one is the most suitable for your problem until you try them all. One reason for this is that the model performance highly depends on the data characteristics (e.g., whether the relationships between the predictors and the objective field are linear or not). Overall, when the most suitable algorithm for the data characteristics has been selected, the infinite combinations of its parameter values can yield very different results. In conclusion, this process requires a lot of manual experimentation from data scientists that need to leverage their knowledge, intuition and experience to iteratively build different models and evaluate them.

OptiML automates model selection and parametrization end-to-end. This automation accelerates the process of improving model performance, saving practitioners time and making sophisticated Machine Learning workflows accessible to non-experts.

2.1 OptiML Technicalities

In order to find the best performing model, OptiML sequentially trains and evaluates models using groups of parameters, and based on the results, it tries a new group of parameters. This process can be split in two main phases:

1. **Model and parameter search**: different sets of promising parameters are iteratively found via Bayesian parameter optimization. OptiML is based on an opimization technique called Sequential Model-based Algorithm Configuration (SMAC)[1]. OptiML tries sets of parameters for all BigML supervised models by default: decision trees, ensembles (up to 256 trees), logistic regressions (only for classification problems), and deepnets. You can also select a subset of those model types (see Subsection 4.4.1). The search is parameterized by an optimization metric that will guide the search (i.e., the best models according to that metric will be tried). By default, OptiML uses the R squared for regression and the maximum phi coefficient (the highest phi coefficient given all probability thresholds) for classification, but you can select another metric from the ones listed in Subsection 4.4.3. In the case of classification problems, you will also be able to select a given class to optimized; otherwise, the average for all classes will be optimized.
2. **Validation**: Monte Carlo cross-validation is iteratively performed on those models close to the optimum. During this phase, OptiML iteratively performs new train and test splits of the original dataset for the top half of the models because there is no point in performing cross-validation for the models with poor performance (that will be discarded afterward). For example, if there are 10 different models built using certain sets of parameters in the first iteration, the cross-validation

[1] http://www.cs.ubc.ca/labs/beta/Projects/SMAC/

will be performed over the top 5 performing models while the worst 5 models will be discarded. Thus, the best models will typically have more than one evaluation associated with them. When the process finishes, the OptiML view will display only one of the cross-validated models, the one associated with the median evaluation for the optimization metric. Along with the optimized metric value, you will also get its standard deviation calculated using all the cross-validated models (see Chapter 5).

Since the search and validation phases can take some time to finish, OptiML has two different parameters to **limit the runtime**. First, there is a limit for the number of models evaluated. OptiML evaluates 128 different **model candidates** (i.e., the models using a unique parametrization) by default, but this limit can be raised up to 200 models (see Section 4.3). You should take into account that due to cross-validation, BigML will build and evaluate a higher number of models in total. In the OptiML view, you will be able to inspect the top half of the model candidates (typically 64 models for a limit of 128 models).

Secondly, there is also a **training duration** to regulate the runtime. It is set to 5 (in a scale of 1 to 10) by default, but the optimum time needed for each OptiML will depend on the dataset size and characteristics (see Section 4.2).

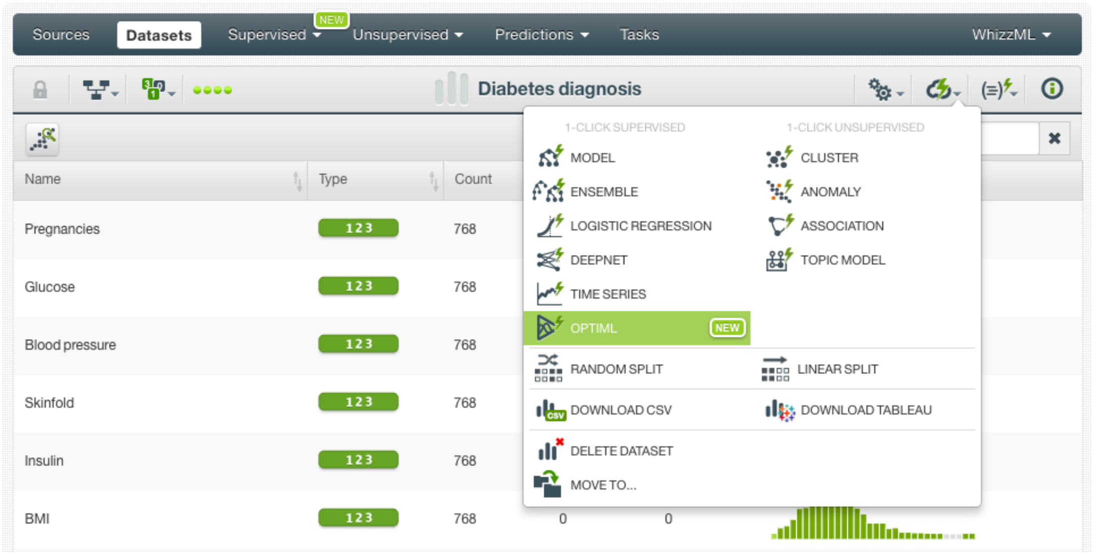
In general, the OptiML is not guaranteed to train the number of models requested. In some cases, the modeling process may run out of time, or certain sets of parameters may cause an error on a given dataset. In these cases, the number of models built will be less than the one requested.

CHAPTER 3

OptiML with 1-Click

To create an OptiML in BigML you have two options: you can use the **1-click option** which uses the default values for the available parameters, or you can tune the parameters in advance using the **configuration option** explained in Chapter 4.

You can find the 1-CLICK OPTIML option in the **1-click action menu** from the dataset view. (See Figure 3.1.)



Name	Type	Count
Pregnancies	123	768
Glucose	123	768
Blood pressure	123	768
Skinfold	123	768
Insulin	123	768
BMI	123	768

Figure 3.1: Create OptiML from 1-click action menu

Alternatively, you can use the 1-CLICK OPTIML option in the **pop up menu** from the dataset list view. (See Figure 3.2.)

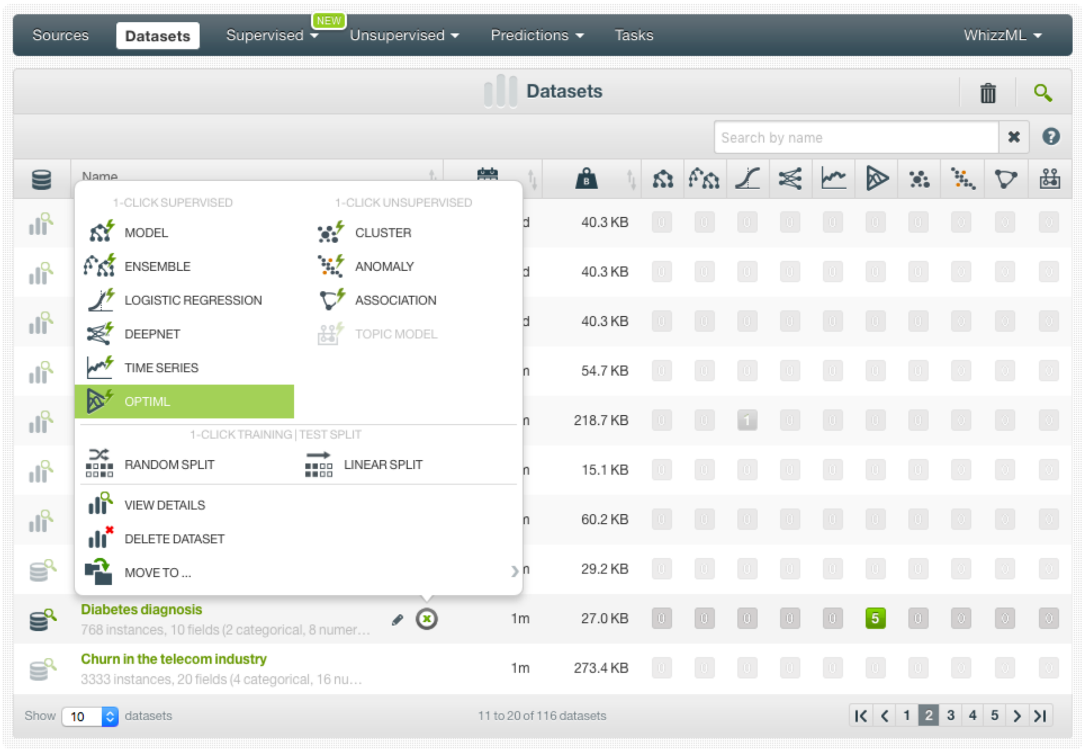


Figure 3.2: Create OptiML from pop up menu

Either option builds an OptiML using the default values for the available configuration parameters explained in the following section. (See Chapter 4.)

CHAPTER 4

OptiML Configuration Options

You can configure a number of parameters that affect the way BigML creates the OptiML. To display the configuration panel, click the CONFIGURE OPTIML menu option in the **configuration menu** from the dataset view. (See Figure 4.1.) See the following sections for a detailed explanation of each parameter.

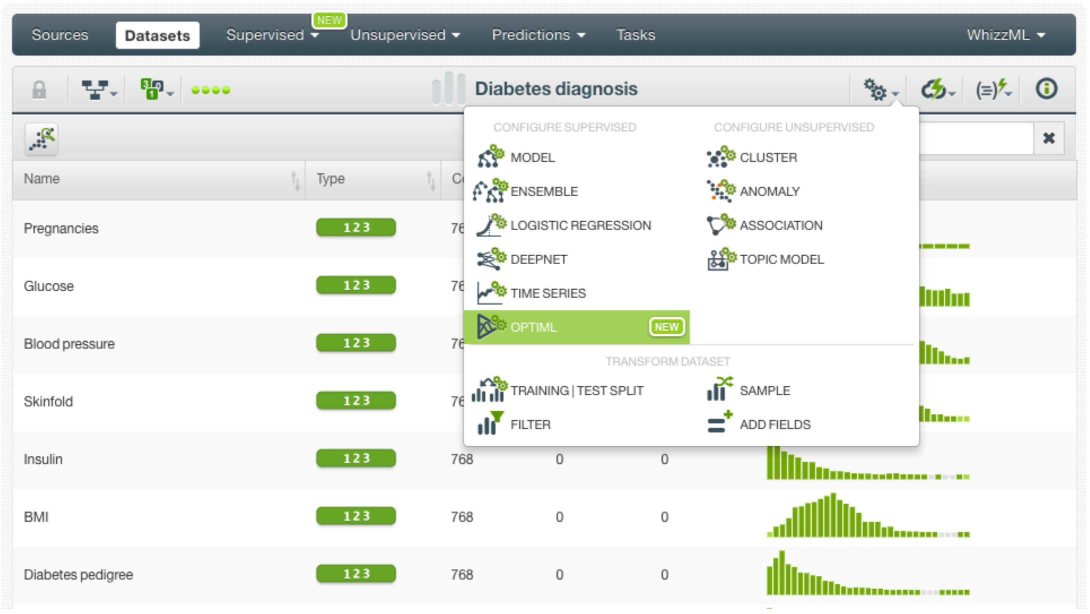
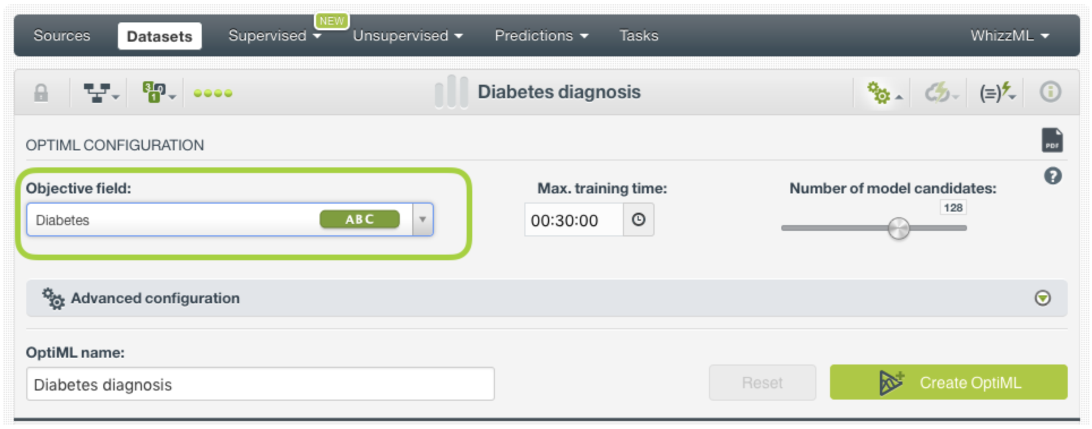


Figure 4.1: Configure OptiML

4.1 Objective Field

The objective field is the field you want to predict. You can select either a **numeric** or **categorical** field. BigML takes the **last field** in your dataset as the objective field by default.



OPTIML CONFIGURATION

Objective field: Diabetes ABC

Max. training time: 00:30:00

Number of model candidates: 128

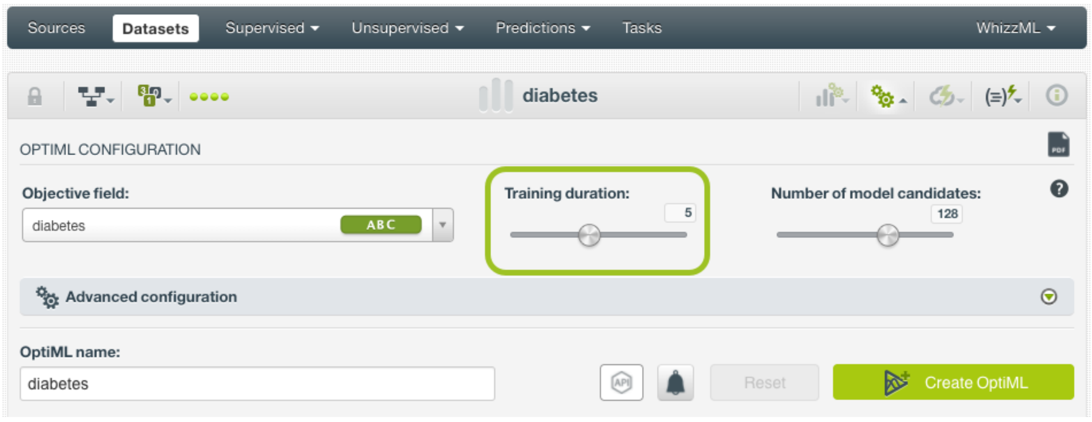
OptiML name: Diabetes diagnosis

Figure 4.2: Configure OptiML objective field

4.2 Training Duration

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



OPTIML CONFIGURATION

Objective field: diabetes ABC

Training duration: 5

Number of model candidates: 128

OptiML name: diabetes

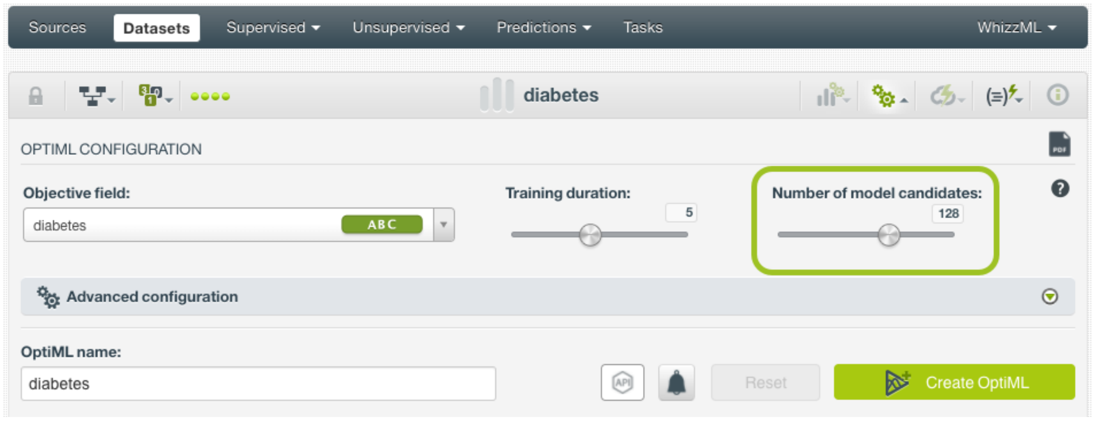
Figure 4.3: Configure OptiML training duration

4.3 Model Candidates

The maximum number of different models (i.e., models using a unique configuration) to be trained and evaluated during the OptiML process. The default is 128 which is usually enough to find the best model, but you can set it from 4 up to 200. The top-performing half of the model candidates will be returned in the final result. Therefore, if you select to evaluate 128 models, the top performing 64 models will be returned unless the training duration halts the process before (see Section 4.2).

The OptiML will usually create a higher number of models than the candidates requested due to cross-validation. These extra models have the same parametrization as the candidates but they use different dataset samples.

Deepnets are a special case. Deepnets have two specific optimization processes: the Automatic Network Search and the Structure Suggestion (see section 4.4.2 of the document Classification and Regression with the BigML Dashboard [3]). Therefore, BigML will perform these optimization processes in the background **returning only two deepnets**, one for each optimization option. The configurable parameters explained in Subsection 4.4.2 and Subsection 4.4.3 do not apply to deepnets.



Objective field: diabetes ABC

Training duration: 5

Number of model candidates: 128

OptiML name: diabetes

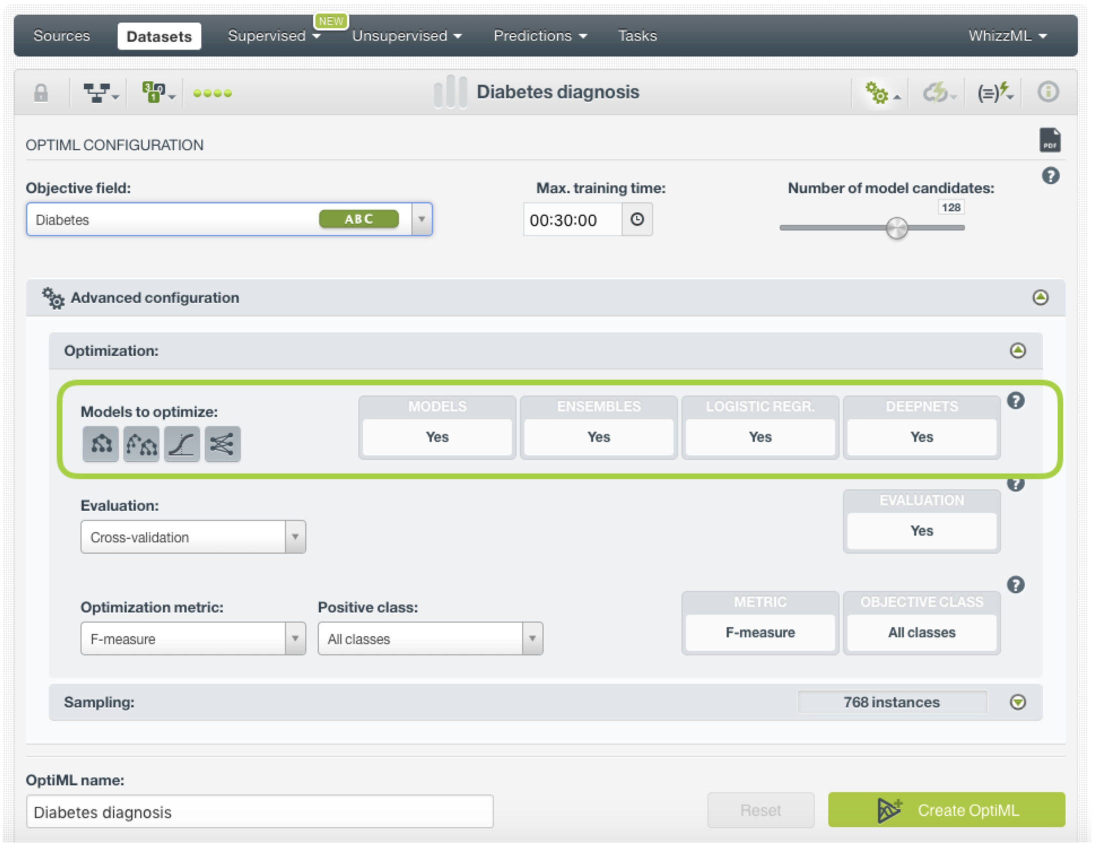
Figure 4.4: Configure the OptiML model candidates

4.4 Model Optimization

This section of the configuration panel allows you to specify a set of parameters that determine which models are optimized and how they are optimized.

4.4.1 Models to Optimize

Select the type of models (or algorithms) that you want to be optimized: decision trees, ensembles (up to 256 trees, including boosted trees and Random Decision Forests), logistic regressions (only for classification problems), and deepnets. By default all types of models are optimized, except in the case of logistic regression, which cannot be trained for regression problems. At least one model per type requested is guaranteed to be evaluated during the optimization process. However, it may happen that one or more of the selected algorithms are not located among the top half performing models, in which case they will not be included in the final list of models.



Objective field: Diabetes ABC

Max. training time: 00:30:00

Number of model candidates: 128

Optimization:

Models to optimize:

MODELS	ENSEMBLES	LOGISTIC REGR.	DEEPNETS
Yes	Yes	Yes	Yes

Evaluation: Cross-validation

EVALUATION
Yes

Optimization metric: F-measure

Positive class: All classes

METRIC	OBJECTIVE CLASS
F-measure	All classes

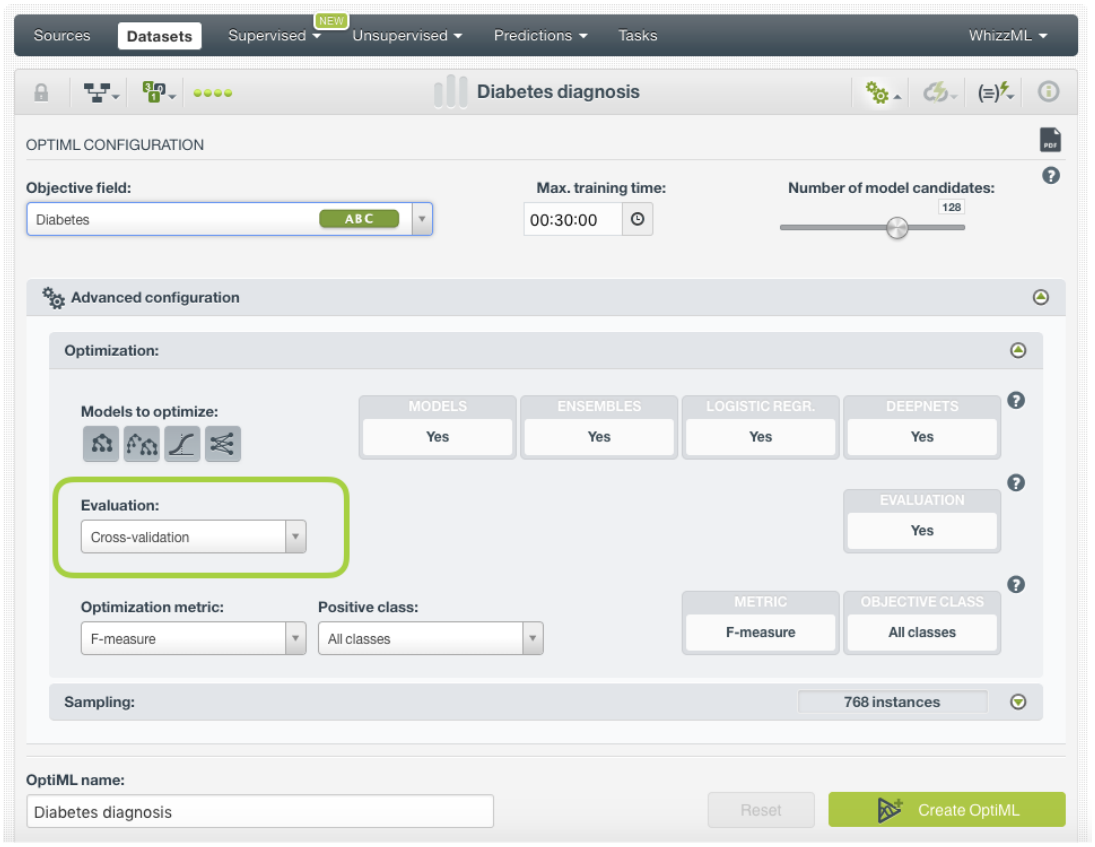
Sampling: 768 instances

OptiML name: Diabetes diagnosis

Figure 4.5: Select the type of models to be optimized

4.4.2 Evaluation

By default, BigML performs **Monte Carlo cross-validation** during the optimization process to evaluate the models and select the top performing ones. Cross-validation evaluations usually yield more accurate results than single evaluations since they avoid the potential error derived from randomly selecting an overly optimistic dataset. Usually smaller datasets, which are presumably less representative of the whole population, have a higher variation in their performance depending on the random split.



OPTIML CONFIGURATION

Objective field: Diabetes ABC

Max. training time: 00:30:00

Number of model candidates: 128

Advanced configuration

Optimization:

Models to optimize:

MODELS	ENSEMBLES	LOGISTIC REGR.	DEEPNETS
Yes	Yes	Yes	Yes

Evaluation: Cross-validation

EVALUATION
Yes

Optimization metric: F-measure

Positive class: All classes

METRIC	OBJECTIVE CLASS
F-measure	All classes

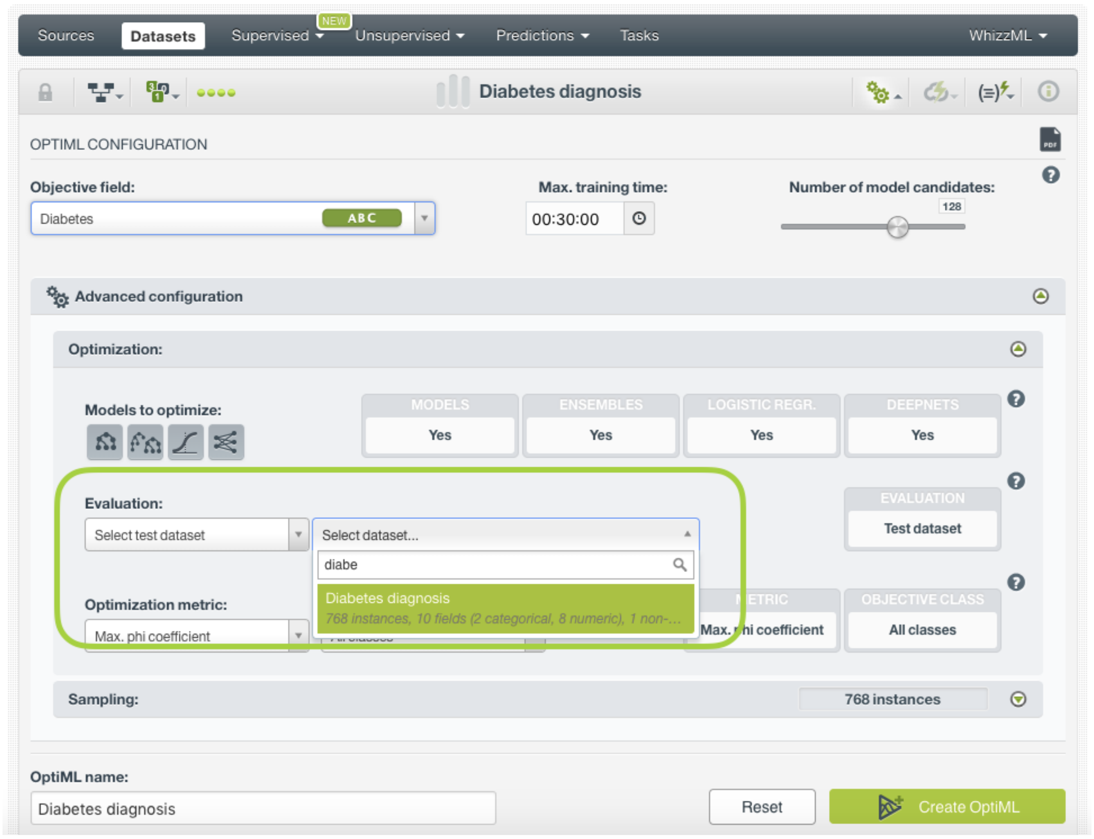
Sampling: 768 instances

OptiML name: Diabetes diagnosis

Figure 4.6: Select cross-validation for model evaluation

However, there are some problems that cannot be evaluated using a random split. For example, problems where you need to train the model using past data and evaluate it with the most recent data like sales forecasting or stock market predictions. For those cases, you can **select a test dataset** that OptiML will use during the optimization process to evaluate the models. To avoid unrealistically good evaluations due to the lack of cross-validation, BigML takes several subsets of the training data to create the models and evaluates them using the testing dataset.

Note: deepnets have their own optimization logic hence the test dataset selection does not apply to them. Cross-validation is always perform to optimize deepnets.



Objective field: Diabetes ABC

Max. training time: 00:30:00

Number of model candidates: 128

Models to optimize: MODELS Yes, ENSEMBLES Yes, LOGISTIC REGR. Yes, DEEPNETS Yes

Evaluation: Select test dataset — Select dataset... — diabe — Diabetes diagnosis, 768 instances, 10 fields (2 categorical, 8 numeric), 1 non-...

EVALUATION: Test dataset

Optimization metric: Max. phi coefficient

OBJECTIVE CLASS: All classes

Sampling: 768 instances

OptiML name: Diabetes diagnosis

Figure 4.7: Select a test dataset for model evaluation

4.4.3 Optimization Metric and Positive Class

The optimization metric is the one used for model selection during the optimization process. For classification problems, BigML uses the **maximum phi coefficient** by default, and for regression problems, the **R squared**, but you can select any other of the below metrics offered (listed below).

Note: deepnets have their own optimization logic hence the optimization metric and positive class configurations do not apply to them. Deepnets are always optimized using a combination of several metrics for all the classes in the objective field in the case of classification problems.

- **Classification metrics**:
  - **Accuracy**: the percentage of correctly classified instances over the total instances evaluated. It can take values between 0% and 100%.
  - **Precision**: the percentage of correctly predicted instances over the total instances predicted for the positive class. It can take values between 0% and 100%.
  - **Recall**: the percentage of correctly classified instances over the total actual instances for the positive class. It can take values between 0% and 100%.
  - **F-measure**: the balanced harmonic mean between precision and recall. It can take values between 0 and 1.
  - **Phi Coefficient**: the correlation coefficient between the predicted and actual values. It can take values between -1 and 1.
  - **Max. phi coefficient**: the maximum phi coefficient taking into account all the possible probability thresholds of the evaluation. It can take values between -1 and 1.
  - **ROC AUC**: the Area Under the ROC curve measures the trade-off between sensitivity and specificity. It can take values between 0 and 1.

  - **Precision-Recall AUC**: the Area Under the Precision-Recall Curve measures the trade-off between precision and recall. It can take values between 0 and 1.
  - **K-S Statistic**: it measures the maximum difference between the True Positive Rate (TPR) and the False Positve Rate (FPR) over all possible thresholds. It can take values between 0% and 100%.
  - **Kendall's Tau**: it 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. It can take values between -1 and 1.
  - **Spearman's Rho**: the degree of correlation between the model predictions and the actual values of the testing dataset. It can take values between -1 and 1.
- **Regression metrics**:
  - **R Squared**: it measures how much better the model is than always predicting the mean value of the target variable. It can take values from -1 to 1.

For an in-depth definition of each metric please refer to the **Evaluations** chapter in the document **Classification and Regression with the BigML Dashboard**. [3]

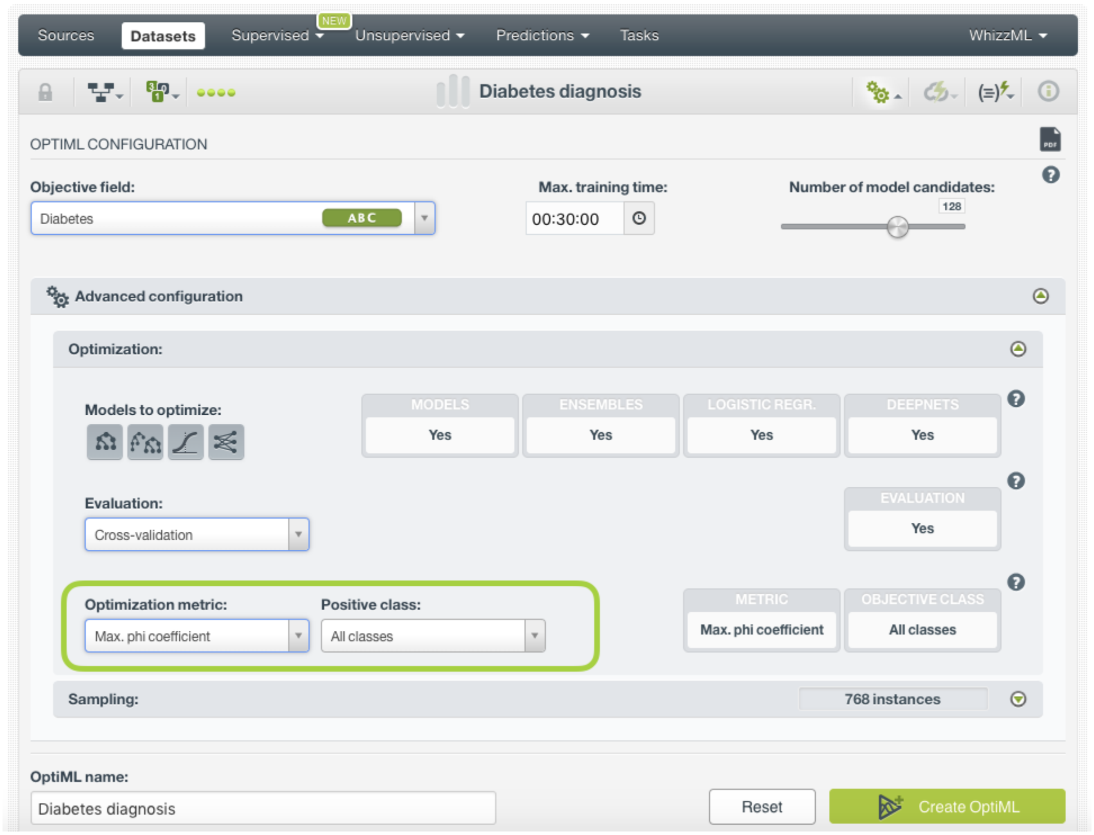
Choosing the **right metric** to be optimized by the OptiML depends on your data characteristics. This decision is mostly related to the need of setting a **threshold**. When evaluating or predicting with your model you can set a probability (or confidence) threshold for a selected class, known as the positive class. Therefore, 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 a more detailed explanation see section 5.2.1.4 of the document Classification and Regression with the BigML Dashboard [3]).

According to this decision of whether to set a threshold or not, you can follow the rules below:

- If you need to set a **threshold**, use the maximum phi, the K-S statistic, the ROC AUC, the precision-recall AUC,the Kendall's Tau or the Spearman's Rho.
- If you need to set a **threshold** and your problem is a **ranking problem** where you care more of your instances with higher probability (for example in the case of customer churn where you have a limited budget so you can only target the top *n* customers most likely to churn), then the decision is reduced to the maximum phi or the K-S statistic.
- If you do **not** need to set a **threshold** and your data is **balanced** (i.e., all classes have similar instances), then you can use the accuracy, the recall or the precision.
- If you do **not** need to set a **threshold** and your data is **imbalanced** (i.e., some classes have significantly more instances than others), then you can use the phi coefficient or the f-measure.

Apart from the optimization metric, for classification problems you can also select a **positive class** to be optimized; otherwise, the average metric for all classes will be optimized.

In general, the two options **metric** and **metric_class** combine to tell the OptiML process where to find the key in the created evaluations that should determine overall model quality. This value is used to drive the model search in OptiML, causing it to try more models that have a better value for this metric and class.



Objective field: Diabetes ABC
Max. training time: 00:30:00
Number of model candidates: 128
Models to optimize: MODELS Yes, ENSEMBLES Yes, LOGISTIC REGR. Yes, DEEPNETS Yes
Evaluation: Cross-validation — EVALUATION Yes
Optimization metric: Max. phi coefficient
Positive class: All classes
METRIC Max. phi coefficient — OBJECTIVE CLASS All classes
Sampling: 768 instances
OptiML name: Diabetes diagnosis

Figure 4.8: Select a metric and a positive class

4.5 Weights

When training an OptiML, there may be some instances in your dataset that are more important than others. BigML provides two different ways to assign specific **weights** to instances: **Objective weights** and the **Weight field** (see Subsection 4.5.1 and Subsection 4.5.2).

Both **Objective weights** and

4.5.1 Objective Weights

In classification problems, 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, supervised models tend to predict the most frequent value simply because the overall 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.

For classification with OptiML, you can set specific weights for each class in the objective field. For example, in Figure 4.9 below, the majority class, “False”, has a weight of 1, while the minority class, “True”, has a weight of 3. BigML oversamples your weighted instances replicating them as many times as the weight stablishes. If the “False” class had three times the number of instances of the “True” class in the original dataset, by setting a weight of 3 for the “True” class you are balancing the model, i.e., the model will be trained with the same number of instances for both classes. If a class is not listed in the objective weights, it is assumed to have a weight of 1. Weights equal to zero are valid as long as there are some positive valued weights. If every weight is equivalent to zero (this is possible with sampled datasets), then the resulting model will have a single node with a nil output.

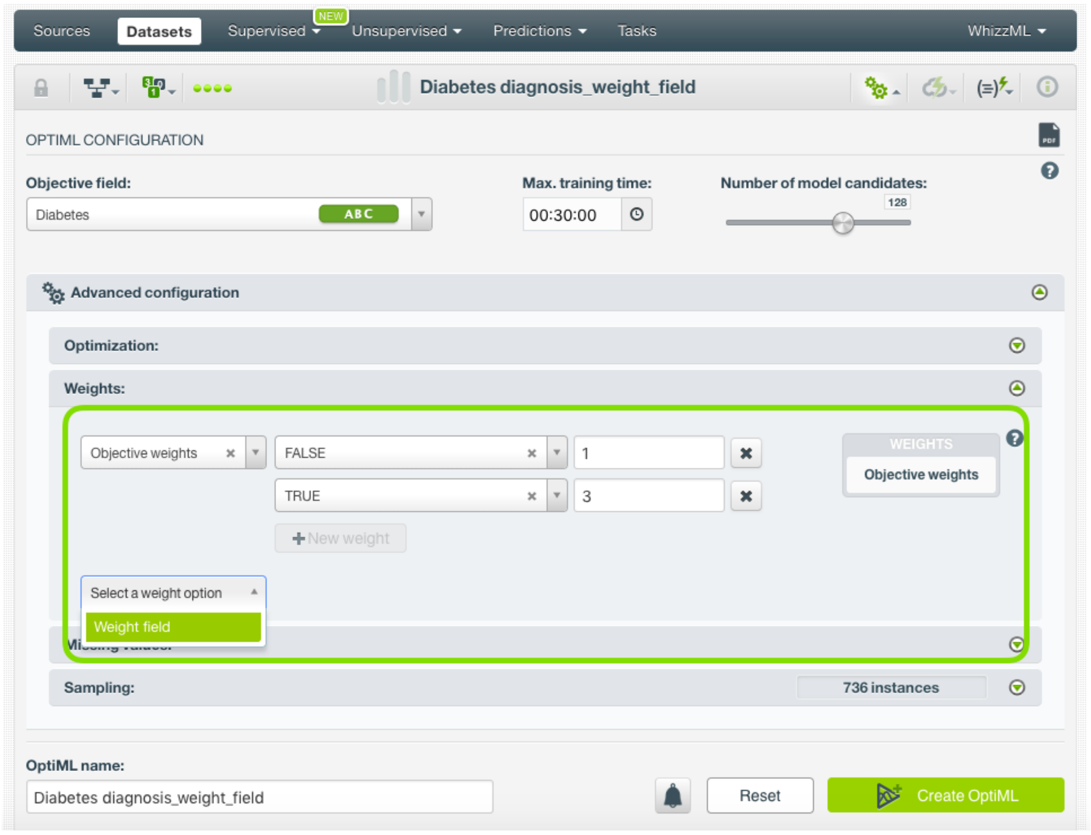


Figure 4.9: Configure the objective weights for the OptiML models

This option is valid only for classification models and you can combine it with the **Weight field** (see Subsection 4.5.2). 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.

4.5.2 Weight Field

The **Weight field** option allows you to assign individual weights to each instance by choosing a special weight field (see Figure 4.10). It can be used for both regression and classification models. 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 models. 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** explained in Subsection 4.5.1 for models, ensembles, and logistic regression. If an instance has a weight of 3 it will be replicated three times in the dataset to train the model. However, it works differently for deepnets. For deepnets, the weight field modifies the loss function to include the instance weight. The outcome is similar to the oversampling technique.

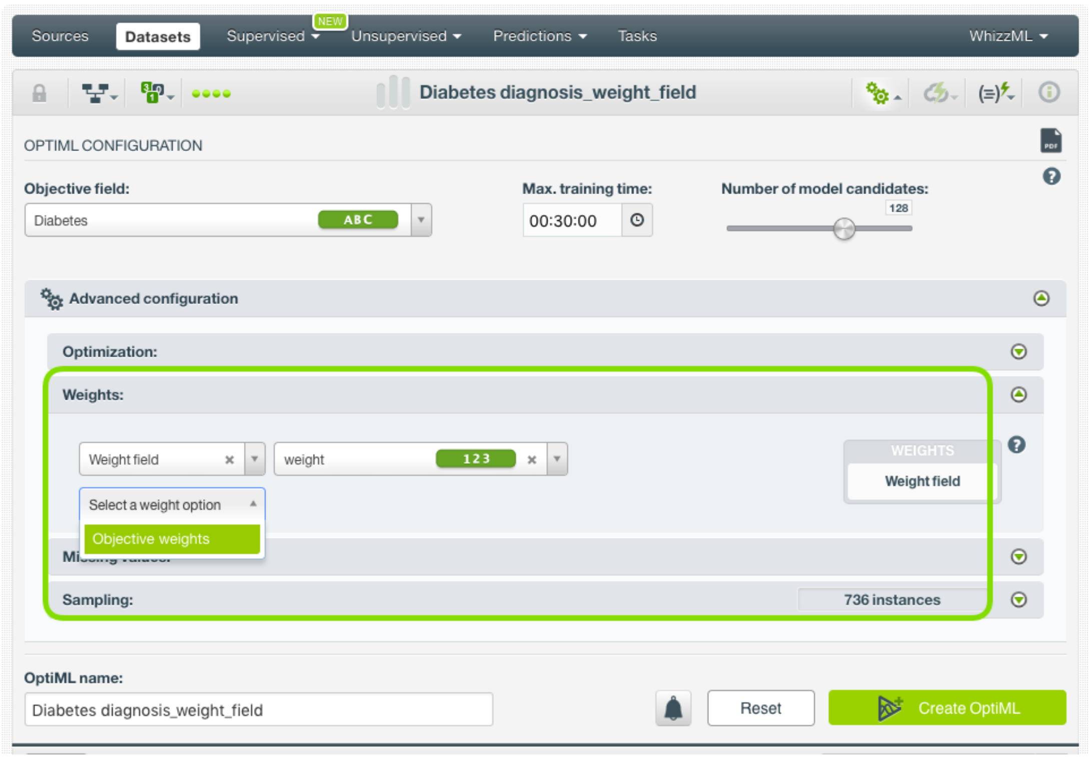


Figure 4.10: Configure the weight field for the OptiML models

4.6 Missing Values

When training the OptiML, BigML may encounter missing values in your dataset, which can be either considered or ignored. The way to include or exclude missing values from your models depends on the model type: for **decision trees and ensembles**, you can configure the **Missing splits** parameter (see Subsection 4.6.1), and for **logistic regressions and deepnets** you can configure the **Default numeric value** and the **Missing numerics** parameters (see Subsection 4.6.2).

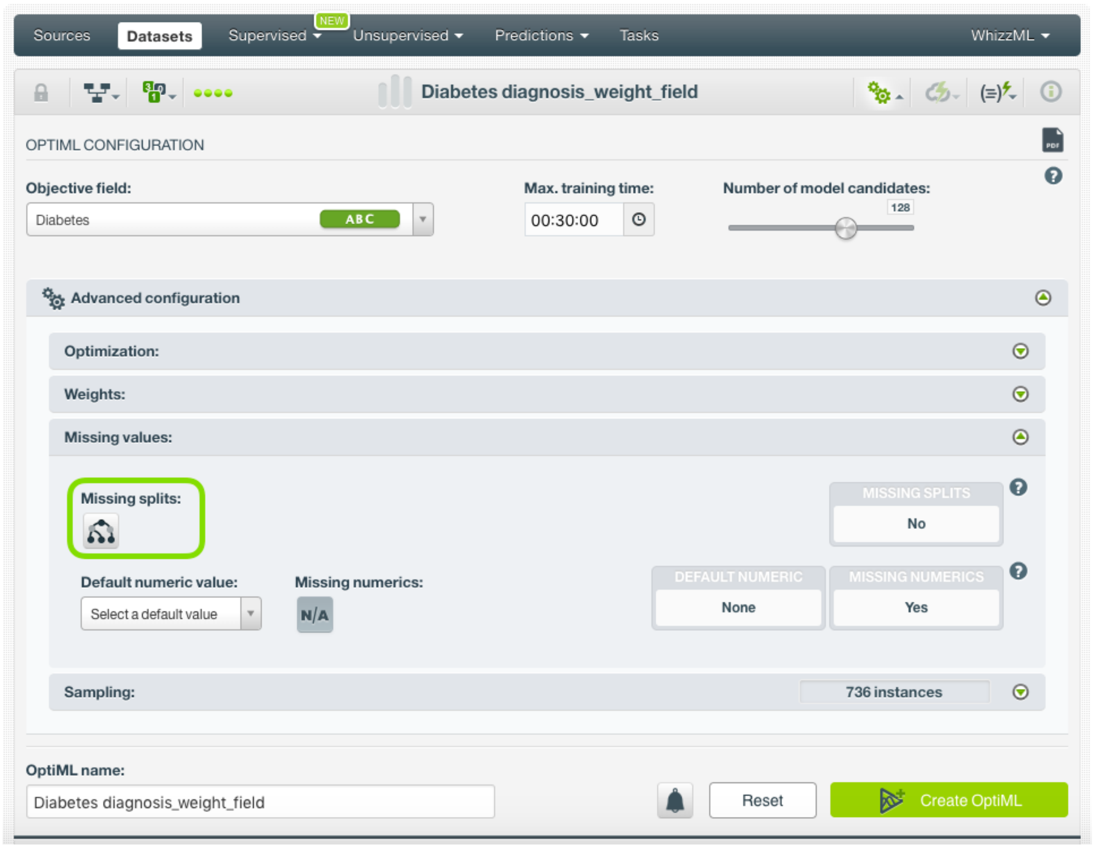
4.6.1 Missing Splits

By default, **decision trees and ensembles** ignore instances that contain missing values (numeric or categorical) for a given field. If the missing values in your data may be important to predict the objective field, then you can include them using the **Missing splits** option (see Figure 4.11).

If missing values are included, 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 approach[1] [9].

[1] `http://oro.open.ac.uk/22531/1/decision_trees.pdf`



Diabetes diagnosis_weight_field

OPTIML CONFIGURATION

Objective field: Diabetes ABC

Max. training time: 00:30:00

Number of model candidates: 128

Advanced configuration

Optimization:

Weights:

Missing values:

Missing splits:

MISSING SPLITS: No

Default numeric value: Select a default value

Missing numerics: N/A

DEFAULT NUMERIC: None

MISSING NUMERICS: Yes

Sampling: 736 instances

OptiML name: Diabetes diagnosis_weight_field

Figure 4.11: Include missing splits in your decision trees and ensembles

This option only includes missing values for **decision trees and ensembles**. To configure the missing values for logistic regressions and deepnets see Subsection 4.6.2.

4.6.2 Default Numeric Value and Missing Numerics

By default, **logistic regression and deepnets** always include missing values (categorical and numeric), but you can choose to exclude the missing numeric values. If the missing numeric values in your data are not important to predict the objective field, then you can exclude or replace them using the **Default numeric value** or the **Missing numerics** parameters (see Figure 4.12).

If the missing numeric values are not important for your objective field and you do not mind removing instances to train the models, then you may opt to disable the **Missing numerics** option. If you do not want to include the missing values but you care about removing the instances with missing numerics, you can use the **Default numeric value**. By using this option, you can replace the missing numeric values with the mean, median, zero, maximum or minimum.

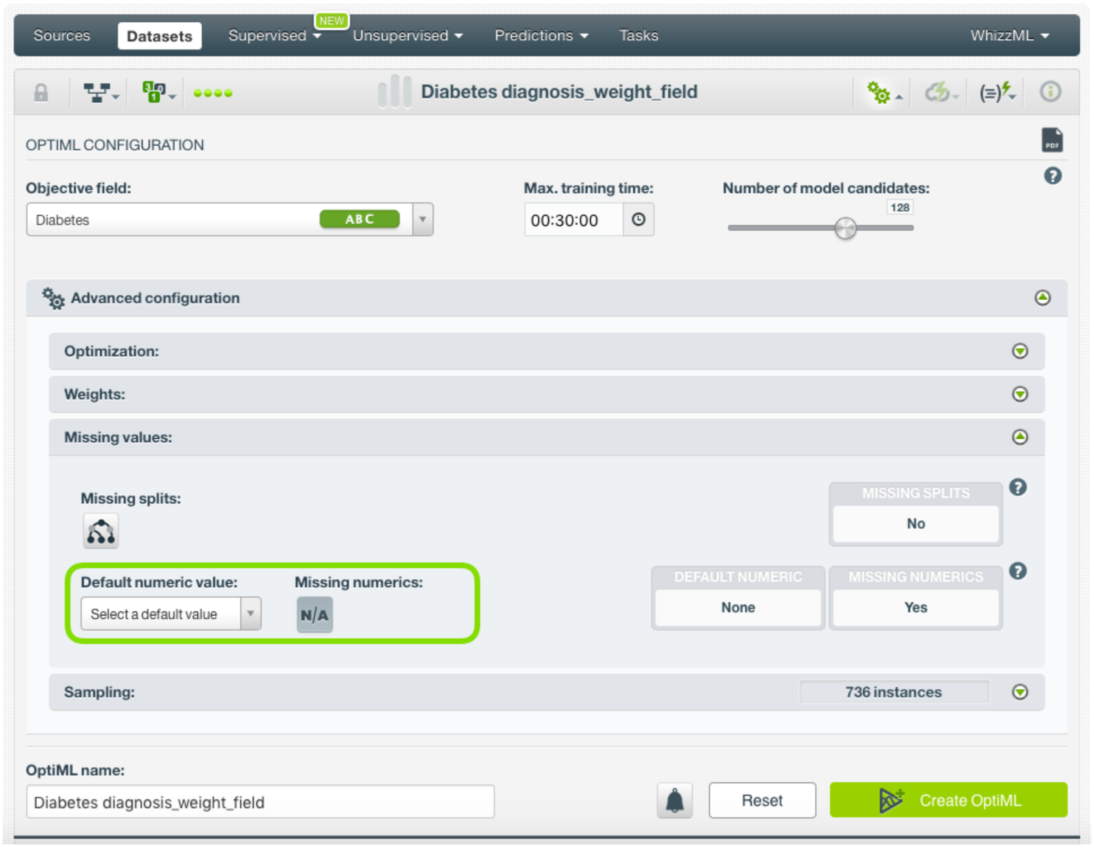


Figure 4.12: Include the missing splits in your decision trees and ensembles

Note: the missing categorical values are always included for logistic regressions and deepnets.

4.7 Sampling

You can specify a subset of the instances of your dataset to create the OptiML. The **rate** option allows you to specify the percentage of instances you want to train your model.

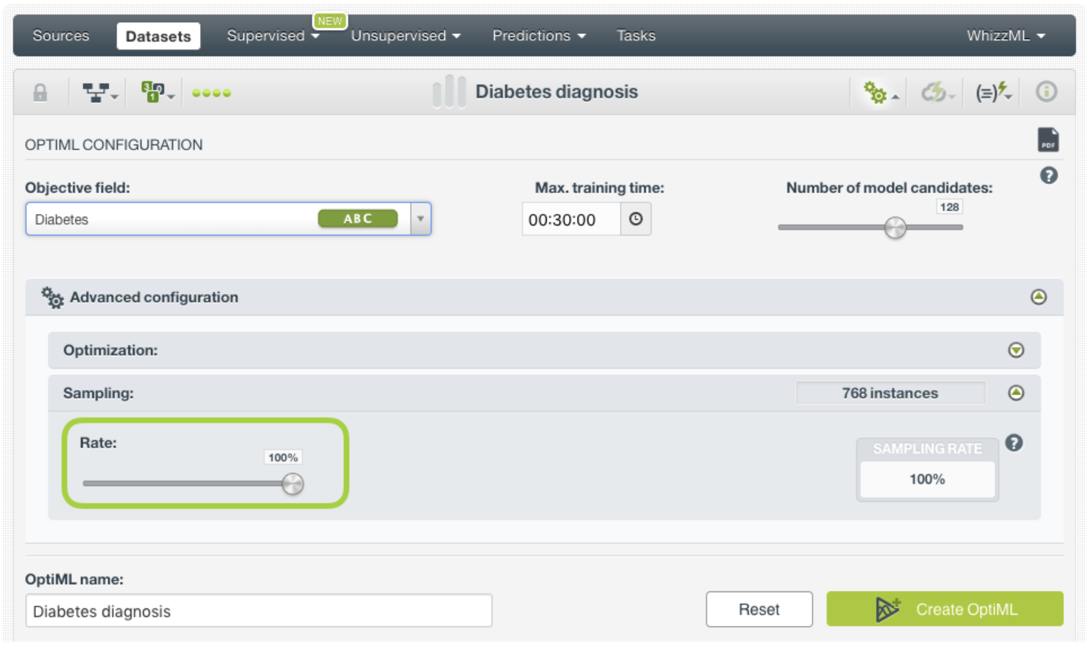


Figure 4.13: Set a rate for your OptiML

4.8 Notifications

If you enable the notifications before creating an OptiML (see Figure 4.14), you will receive an e-mail notifying you that the OptiML has been created.

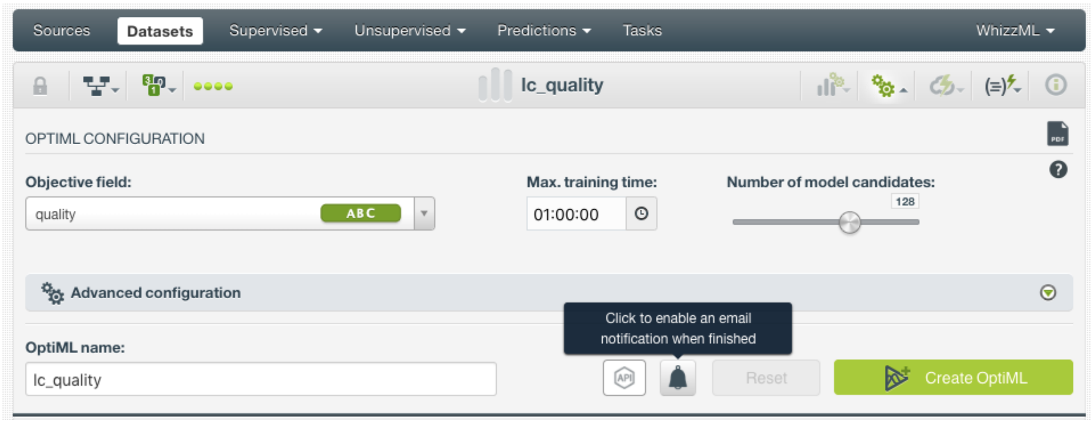
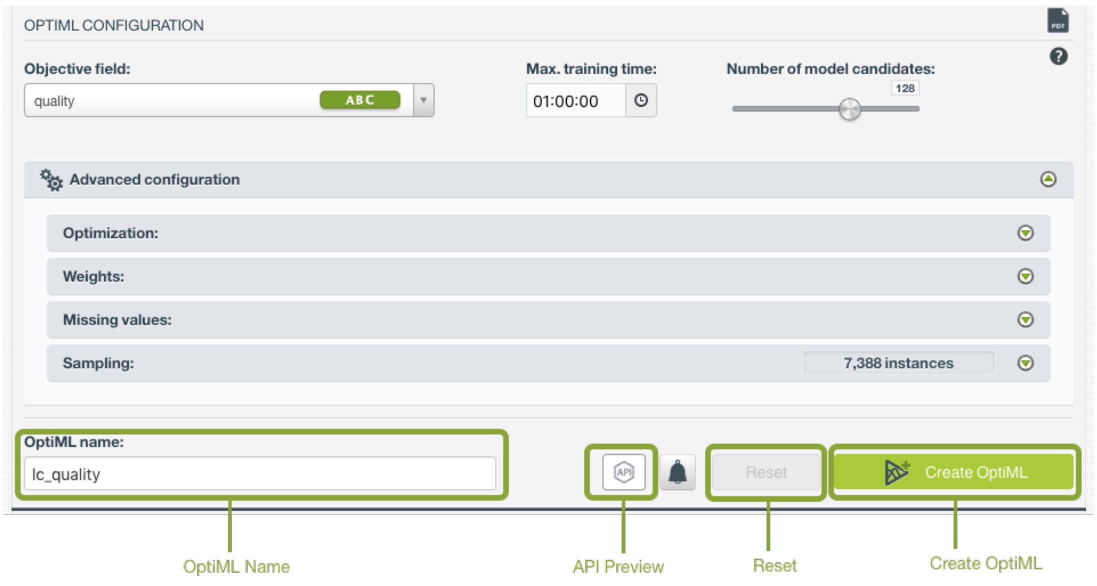


Figure 4.14: Receive an e-mail when the OptiML is created

4.9 Creating OptiMLs with Configured Options

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

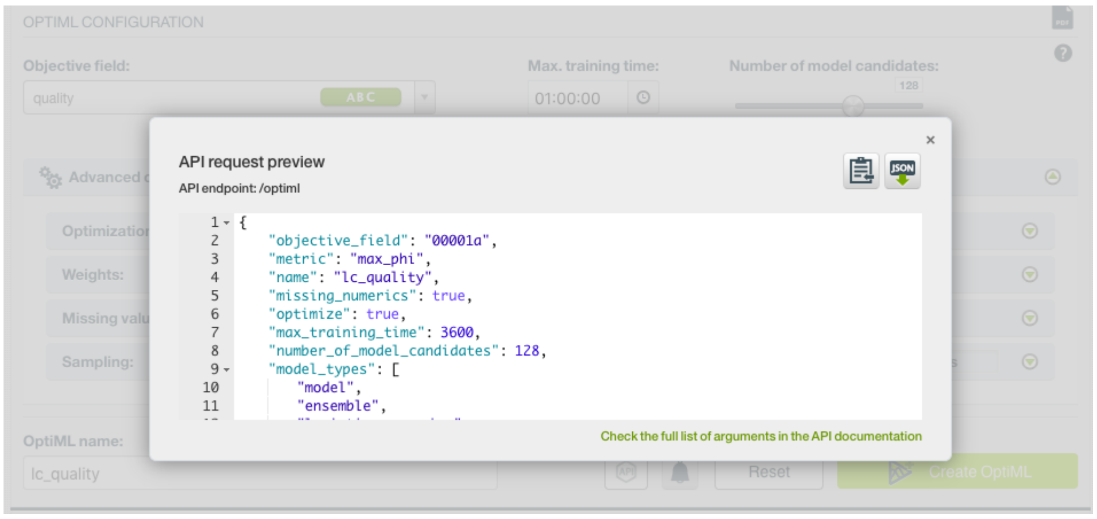


Objective field: quality ABC
Max. training time: 01:00:00
Number of model candidates: 128
Sampling: 7,388 instances
OptiML name: lc_quality

Figure 4.15: Create OptiML after configuration

4.10 API Request Preview

The API Request Preview button is in the middle on the bottom of the configuration panel, next to the Notification button (See (Figure 4.15)). This is to show how to create the OptiML programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 4.16) below:



API request preview

API endpoint: /optiml

```
{
    "objective_field": "00001a",
    "metric": "max_phi",
    "name": "lc_quality",
    "missing_numerics": true,
    "optimize": true,
    "max_training_time": 3600,
    "number_of_model_candidates": 128,
    "model_types": [
        "model",
        "ensemble",
```

Check the full list of arguments in the API documentation

Figure 4.16: OptiML 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 OptiML, 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.

CHAPTER 5

Visualizing OptiML

First of all, when you create an OptiML you will be redirected to the in-progress view (Subsection 5.0.1). When the process finishes, you will see the OptiML main view (Subsection 5.0.2). Finally, you will be able to find the single models composing the OptiML in your Dashboard (Subsection 5.0.3).

5.0.1 OptiML In-Progress View

When you create an OptiML, you will see an intermediate view (until the process finishes) containing a set of metrics that allow you to track the progress of the OptiML creation. Find below a detailed explanation of the elements composing the in-progress view.

- **Elapsed time**: the OptiML runtime. This is regulated by the training duration which can be configured (see Section 4.2).
- **Models evaluated**: the number of models trained by the OptiML using different sets of parameters. Although there is a default limit of 128 model candidates, the total models evaluated will usually be a higher number due to cross-validation. The total models evaluated can also be less than the model candidates configured (see Section 4.3).
- **Processed data**: the total dataset size processed by the OptiML multiplied by the number of resources created.
- **Resources created**: the total number of resources created by the optimization process. These resources include the models evaluated, the datasets and the evaluations (result of the cross-validation).
- **Resources flow-chart**: this is a simpliflied workflow of the OptiML process. The original dataset is split in two subsets (one for training and the other for testing) and the training set is used to build the models. Finally, the models are evaluated using the test dataset. The cross-validation phase is not represented in this flow.
- **Score bar chart**: all the values of the optimized metric are being plotted as they are generated by the model evaluations. Each bar in the chart is a different score from a given evaluation.

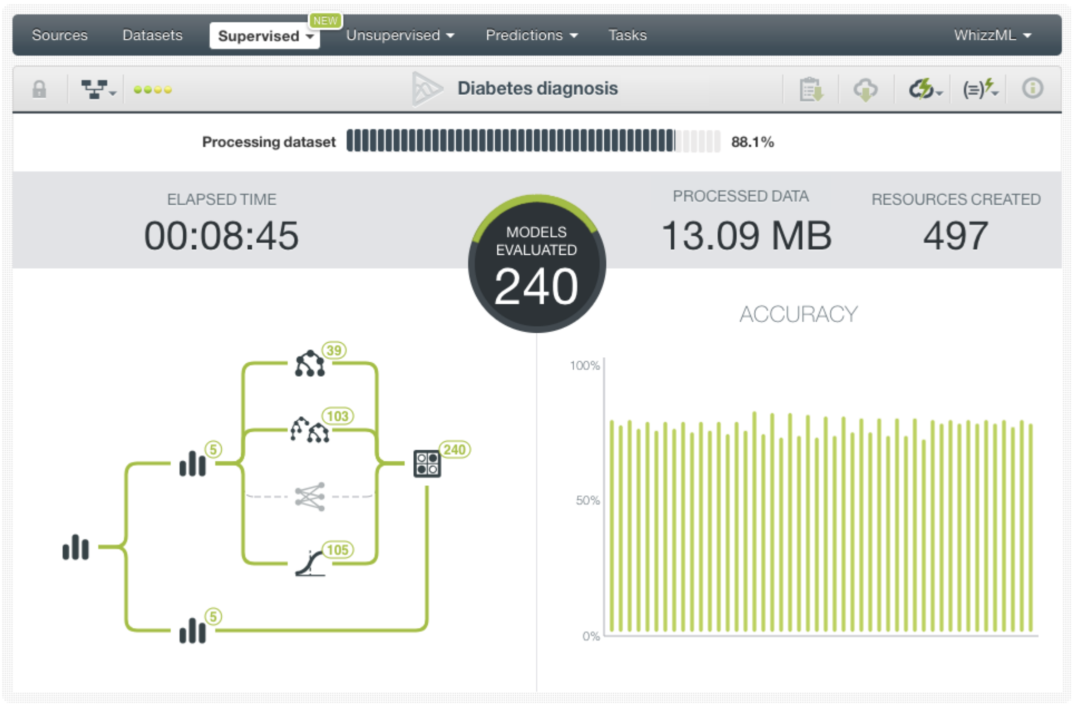


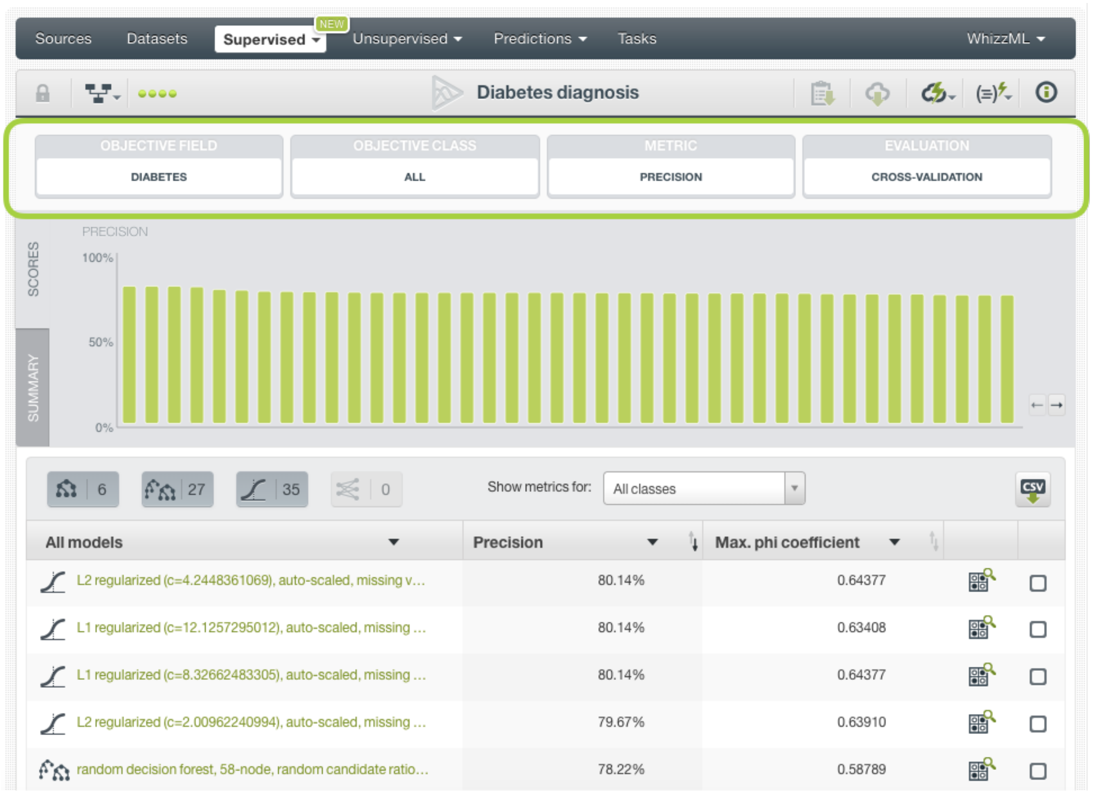
Figure 5.1: OptiML in-progress view

5.0.2 OptiML Main View

When the creation process finishes, you will see the results in the OptiML main view. This view is designed to **compare** the top resulting **optimized models** and **select the best one** in order to make **predictions** with it.

The main view is composed of the following elements:

- **Top menu**: displays some information about the OptiML configuration such as what is the objective field, the class to be optimizied (if it is a classification problem), the metric to be optimized, and the evaluation strategy (cross-validation or test dataset selection). You can read how to configure all these parameters in Chapter 4.



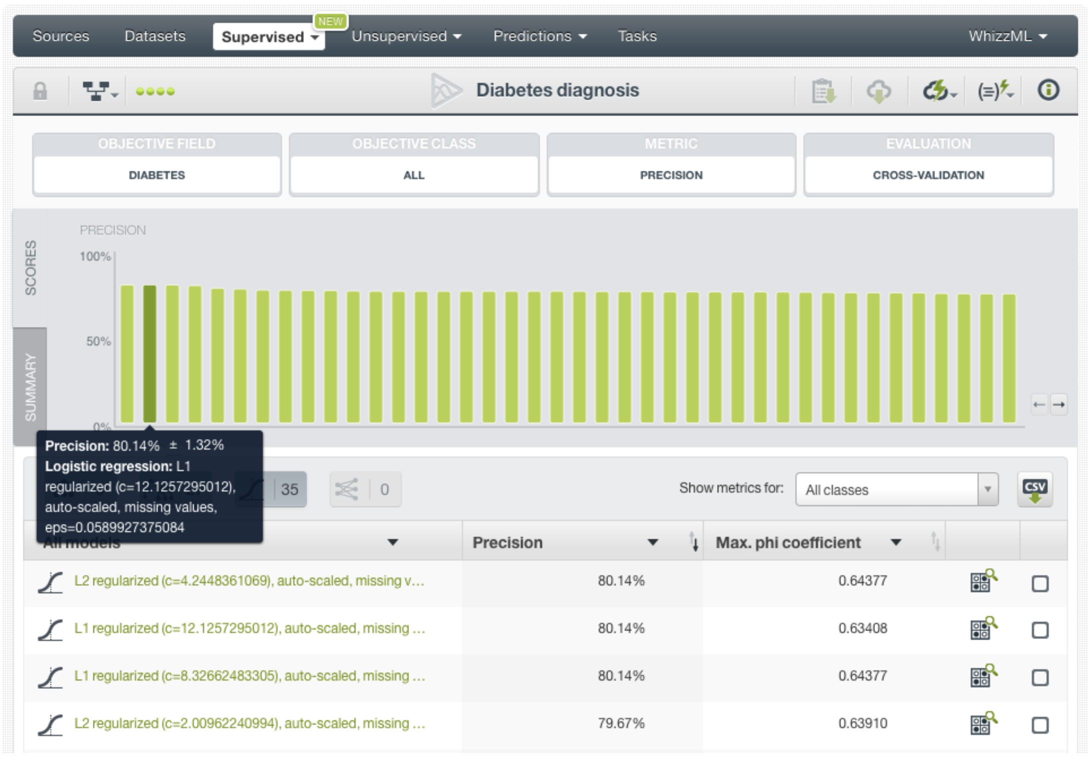
OBJECTIVE FIELD	OBJECTIVE CLASS	METRIC	EVALUATION
DIABETES	ALL	PRECISION	CROSS-VALIDATION

Show metrics for: All classes

All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v...	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing ...	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910
random decision forest, 58-node, random candidate ratio...	78.22%	0.58789

Figure 5.2: OptiML top menu information

- **Score bar chart**: this bar chart contains the final results per model. Each bar is the value of the optimized metric for each model. If you mouse over the bars you will see each metric value, the standard deviation, the type of model and its parametrization. If you click on the bar, you will be redirected to the corresponding model view.



OBJECTIVE FIELD	OBJECTIVE CLASS	METRIC	EVALUATION
DIABETES	ALL	PRECISION	CROSS-VALIDATION

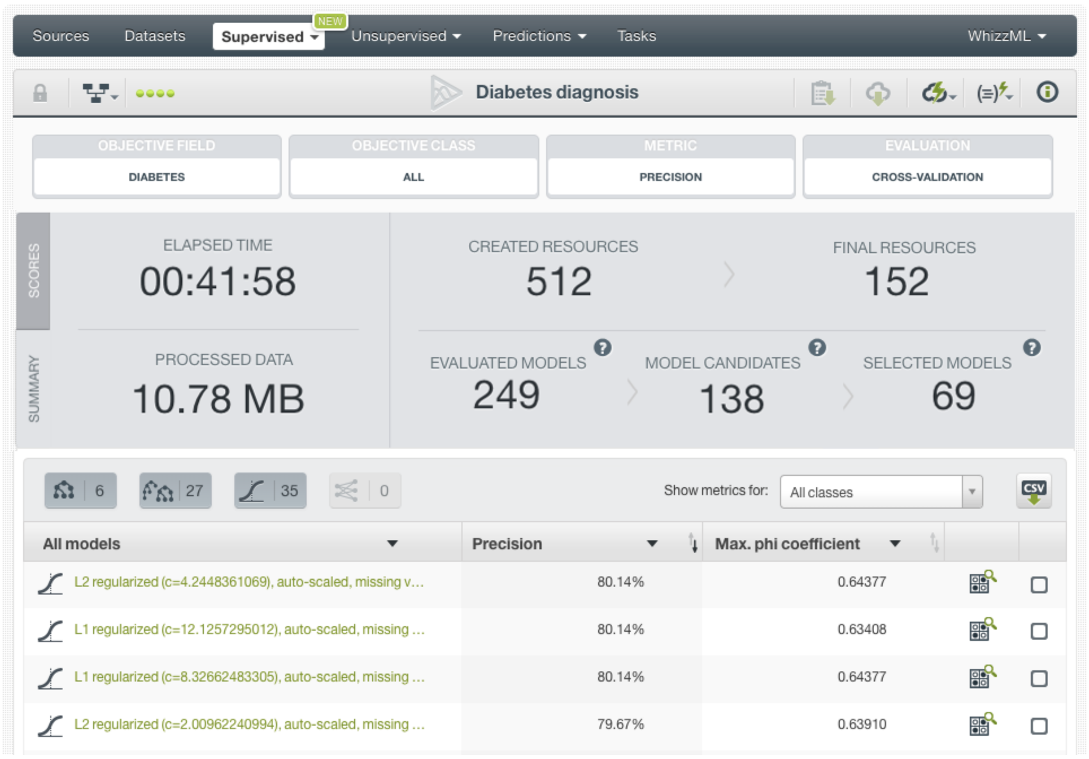
Precision: 80.14% ± 1.32%
Logistic regression: L1 regularized (c=12.1257295012), auto-scaled, missing values, eps=0.0589927375084

Show metrics for: All classes

All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v...	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing ...	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910

Figure 5.3: OptiML score bar chart

- **Summary**: this is the summary of the OptiML results where you can find the following information:
  - **Elapsed time**: the time it took to create the OptiML. This time is regulated by the training duration configured (see Section 4.2).
  - **Processed data**: the amount of MB or GB that the OptiML processed in order to create all the resources.
  - **Created resources**: include the models, datasets and evaluations created.
  - **Final resources**: the remaining resources after the selection of the top performing half of the model candidates.
  - **Evaluated models**: all the models that have been evaluated during the optimization process. This number is higher than the model candidates due to cross-validation.
  - **Model candidates**: the total number of distinct models (i.e., models using a unique parametrization) that have been trained during the optimization process. This number should be equal to the total model candidates configured (see Section 4.3). It may be lower in the case that the training duration stopped the process before all the model candidates were created.
  - **Selected models**: the top-performing half of the model candidates. These are the models included in the OptiML result.



OBJECTIVE FIELD	OBJECTIVE CLASS	METRIC	EVALUATION
DIABETES	ALL	PRECISION	CROSS-VALIDATION

ELAPSED TIME 00:41:58

PROCESSED DATA 10.78 MB

CREATED RESOURCES 512

FINAL RESOURCES 152

EVALUATED MODELS 249

MODEL CANDIDATES 138

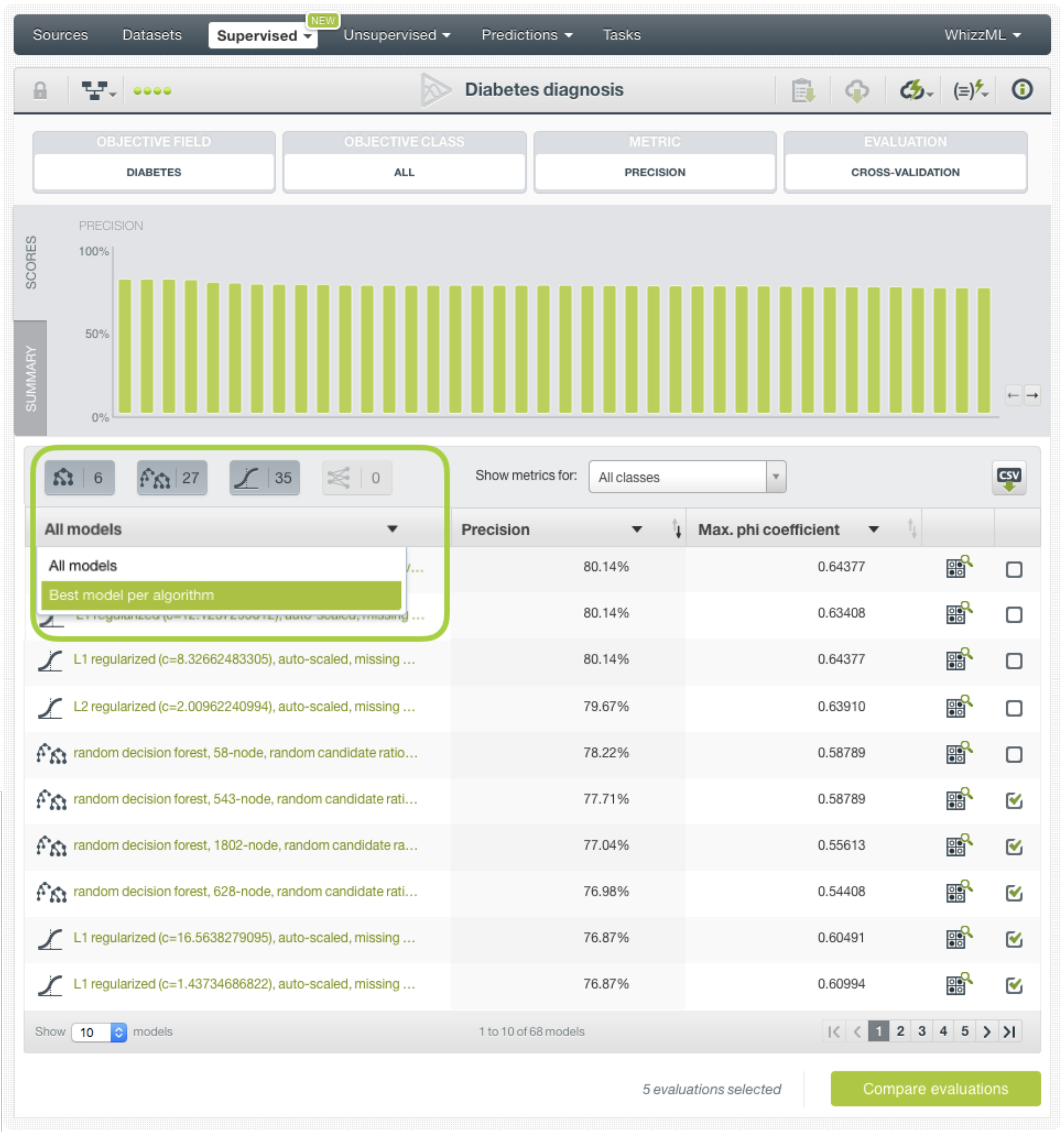
SELECTED MODELS 69

6 27 35 0

All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v...	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing ...	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910

Figure 5.4: OptiML summary

- **Model table**: all the selected models ranked by their performance, allowing you to compare them and select the model that best suits your use case.
  - **Filter models**: you can select the type of models you want to see in the table by clicking on the corresponding icons. You can also see the best model per model type by choosing the option shown in Figure 5.5.



OBJECTIVE FIELD: DIABETES
OBJECTIVE CLASS: ALL
METRIC: PRECISION
EVALUATION: CROSS-VALIDATION

Show metrics for: All classes

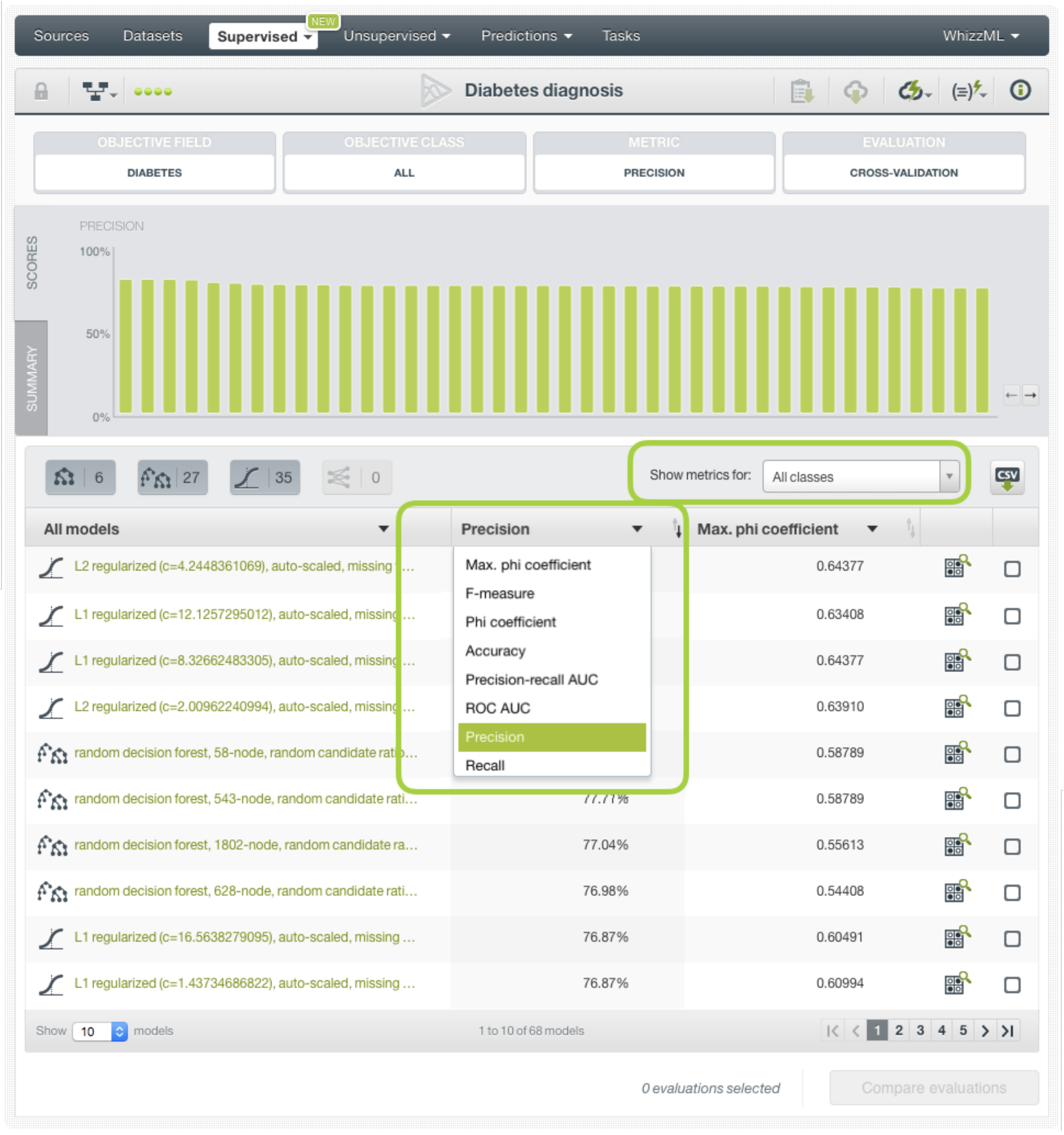
All models	Precision	Max. phi coefficient		
[illegible]	80.14%	0.64377		☐
[illegible]	80.14%	0.63408		☐
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377		☐
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910		☐
random decision forest, 58-node, random candidate ratio...	78.22%	0.58789		☐
random decision forest, 543-node, random candidate rati...	77.71%	0.58789		✓
random decision forest, 1802-node, random candidate ra...	77.04%	0.55613		✓
random decision forest, 628-node, random candidate rati...	76.98%	0.54408		✓
L1 regularized (c=16.5638279095), auto-scaled, missing ...	76.87%	0.60491		✓
L1 regularized (c=1.43734686822), auto-scaled, missing ...	76.87%	0.60994		✓

Show 10 models 1 to 10 of 68 models

5 evaluations selected

Figure 5.5: Filter OptiML models

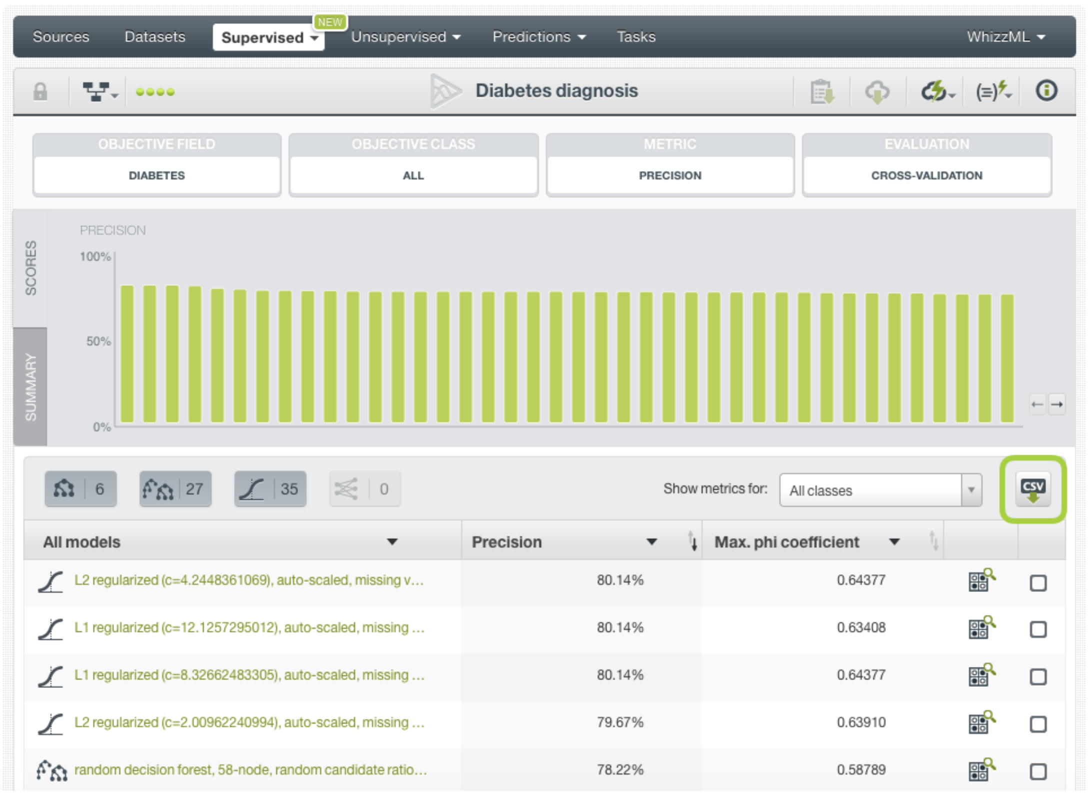
– **Select metrics**: you can compare the model performances by selecting the metrics you want in the table columns as shown in Figure 5.6. For classification problems, you can choose to show the average metrics for all the classes in the objective field or you can select a class to see its metrics. If the objective field has more than 16 classes, only the first 16 classes will be available.



All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing …		0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing …		0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing …		0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing …		0.63910
random decision forest, 58-node, random candidate ratio…		0.58789
random decision forest, 543-node, random candidate rati…	77.71%	0.58789
random decision forest, 1802-node, random candidate ra…	77.04%	0.55613
random decision forest, 628-node, random candidate rati…	76.98%	0.54408
L1 regularized (c=16.5638279095), auto-scaled, missing …	76.87%	0.60491
L1 regularized (c=1.43734686822), auto-scaled, missing …	76.87%	0.60994

Figure 5.6: Select metrics

– **Export results**: you can export the OptiML table in a CSV file format by clicking on the option highlighted in Figure 5.7. For classification problems, your CSV file will contain the metrics of the class that you have selected in the OptiML view.

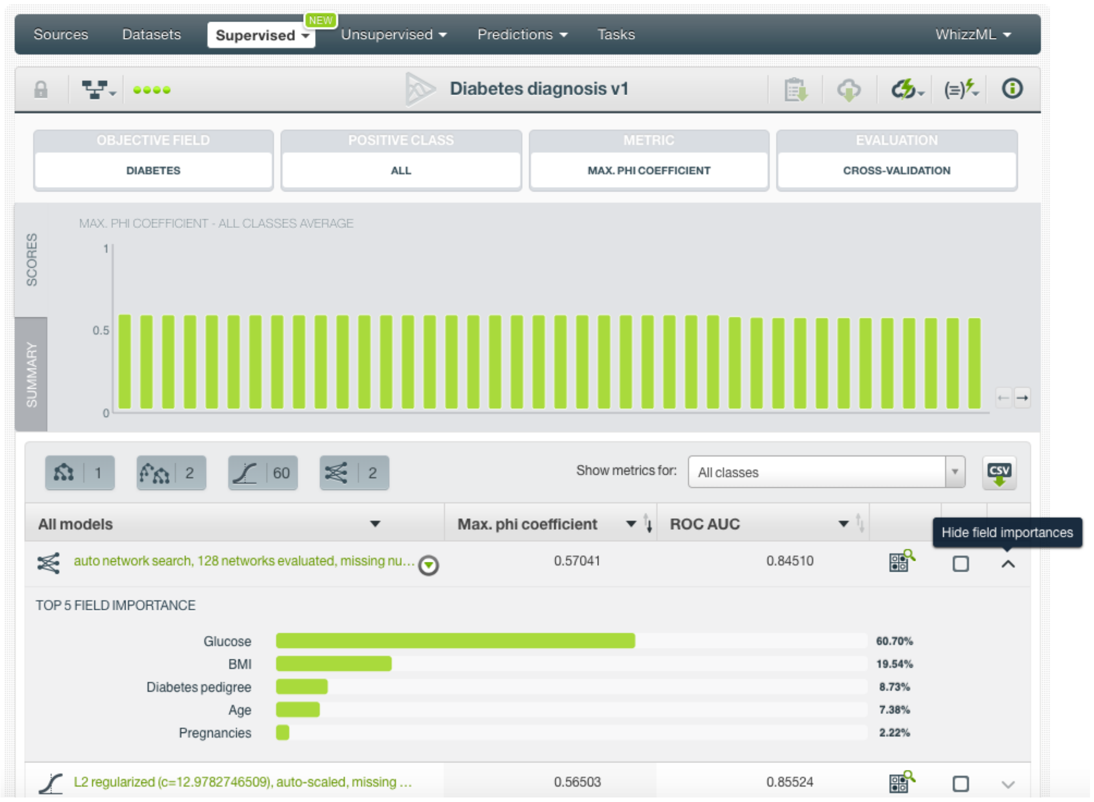


Show metrics for: All classes

All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v…	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing …	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing …	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing …	79.67%	0.63910
random decision forest, 58-node, random candidate ratio…	78.22%	0.58789

Figure 5.7: Export OptiML in a CSV file format

– **Top five important fields**: you can see the five most important fields for models, ensembles and deepnets by clicking in the icon shown in Figure 5.8. The field importance measures the impact of each field int he model predictions. If you want to see all field importances you can do it from the model view as usual.



All models	Max. phi coefficient	ROC AUC
auto network search, 128 networks evaluated, missing nu...	0.57041	0.84510
L2 regularized (c=12.9782746509), auto-scaled, missing ...	0.56503	0.85524

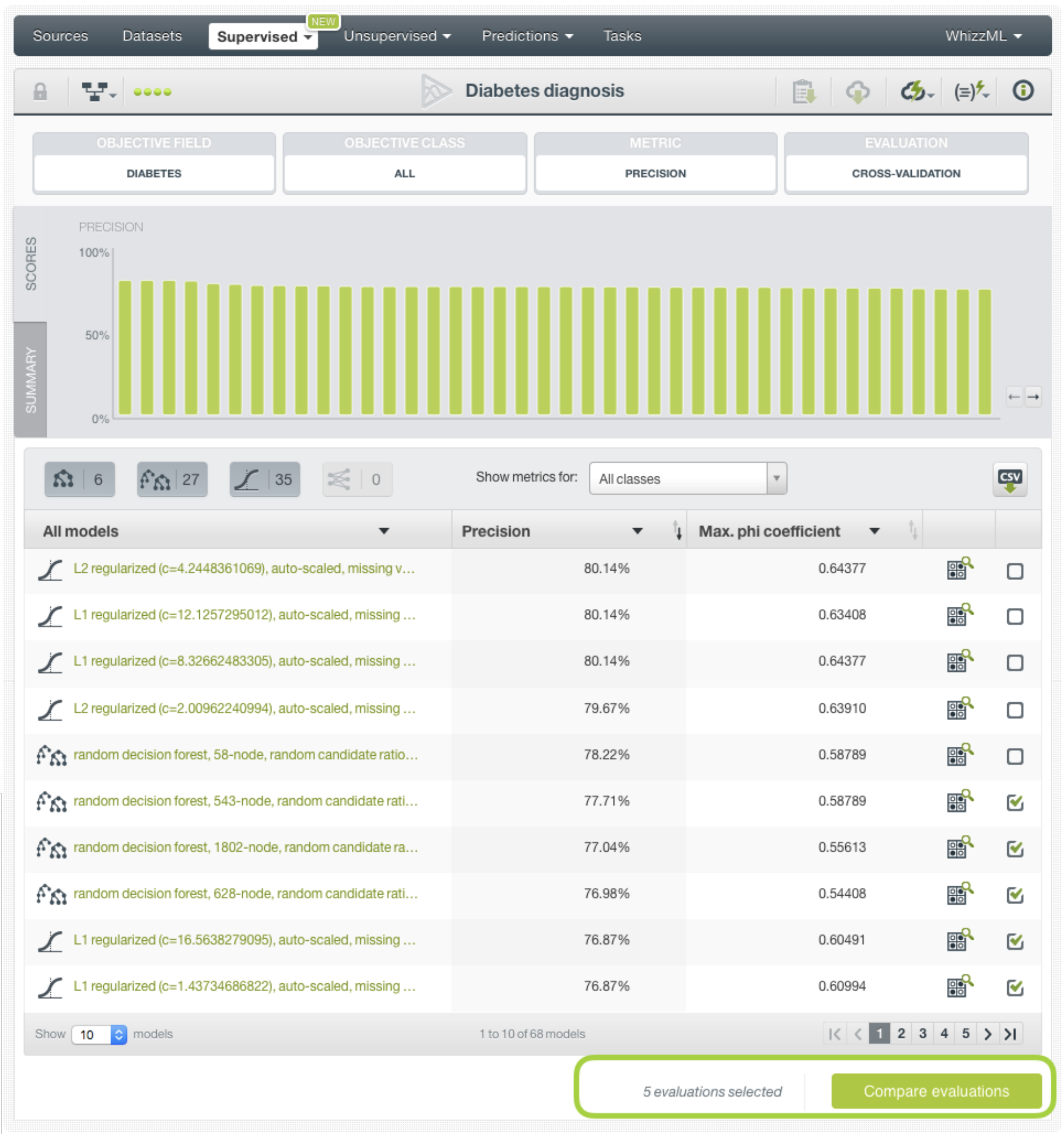
TOP 5 FIELD IMPORTANCE

Glucose	60.70%
BMI	19.54%
Diabetes pedigree	8.73%
Age	7.38%
Pregnancies	2.22%

Figure 5.8: Top five field importances

Note: remember that field importances are not calculated for logistic regressions.

– **Select models**: you can compare multiple model evaluations at the same time using the BigML evaluation comparison chart. For classification problems, you can select up to 20 evaluations and for regression problems you can select two evaluations to compare them side-by-side.



OBJECTIVE FIELD: DIABETES
OBJECTIVE CLASS: ALL
METRIC: PRECISION
EVALUATION: CROSS-VALIDATION

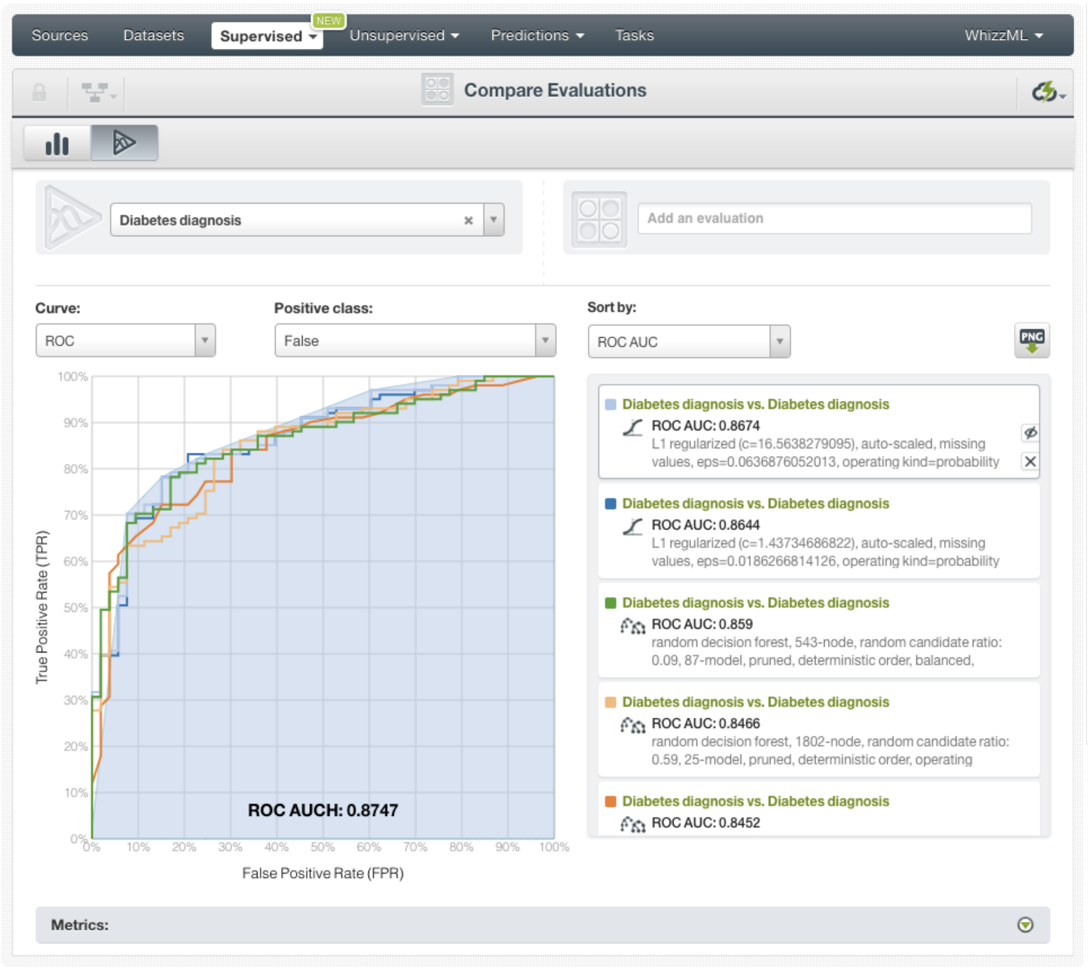
Show metrics for: All classes

All models	Precision	Max. phi coefficient		
L2 regularized (c=4.2448361069), auto-scaled, missing v…	80.14%	0.64377		☐
L1 regularized (c=12.1257295012), auto-scaled, missing …	80.14%	0.63408		☐
L1 regularized (c=8.32662483305), auto-scaled, missing …	80.14%	0.64377		☐
L2 regularized (c=2.00962240994), auto-scaled, missing …	79.67%	0.63910		☐
random decision forest, 58-node, random candidate ratio…	78.22%	0.58789		☐
random decision forest, 543-node, random candidate rati…	77.71%	0.58789		✓
random decision forest, 1802-node, random candidate ra…	77.04%	0.55613		✓
random decision forest, 628-node, random candidate rati…	76.98%	0.54408		✓
L1 regularized (c=16.5638279095), auto-scaled, missing …	76.87%	0.60491		✓
L1 regularized (c=1.43734686822), auto-scaled, missing …	76.87%	0.60994		✓

Show 10 models 1 to 10 of 68 models 1 2 3 4 5

5 evaluations selected Compare evaluations

Figure 5.9: Select models



Diabetes diagnosis

Curve: ROC

Positive class: False

Sort by: ROC AUC

🟦 Diabetes diagnosis vs. Diabetes diagnosis
ROC AUC: 0.8674
L1 regularized (c=16.5638279095), auto-scaled, missing values, eps=0.0636876052013, operating kind=probability

🟦 Diabetes diagnosis vs. Diabetes diagnosis
ROC AUC: 0.8644
L1 regularized (c=1.43734686822), auto-scaled, missing values, eps=0.0186266814126, operating kind=probability

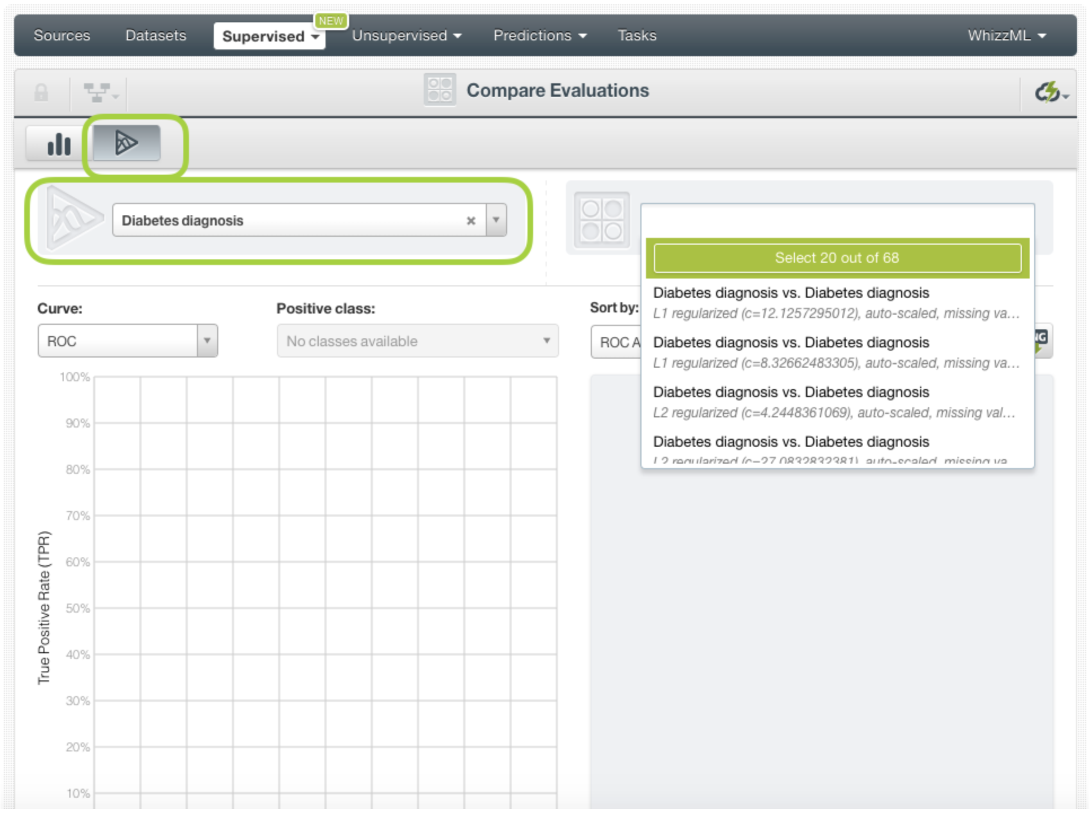
🟩 Diabetes diagnosis vs. Diabetes diagnosis
ROC AUC: 0.859
random decision forest, 543-node, random candidate ratio: 0.09, 87-model, pruned, deterministic order, balanced,

🟧 Diabetes diagnosis vs. Diabetes diagnosis
ROC AUC: 0.8466
random decision forest, 1802-node, random candidate ratio: 0.59, 25-model, pruned, deterministic order, operating

🟧 Diabetes diagnosis vs. Diabetes diagnosis
ROC AUC: 0.8452

Figure 5.10: Compare evaluations

Alternatively, from the multiple evaluation comparison chart view, you can select an OptiML, and then select the evaluations you want to compare as shown in Figure 5.10.

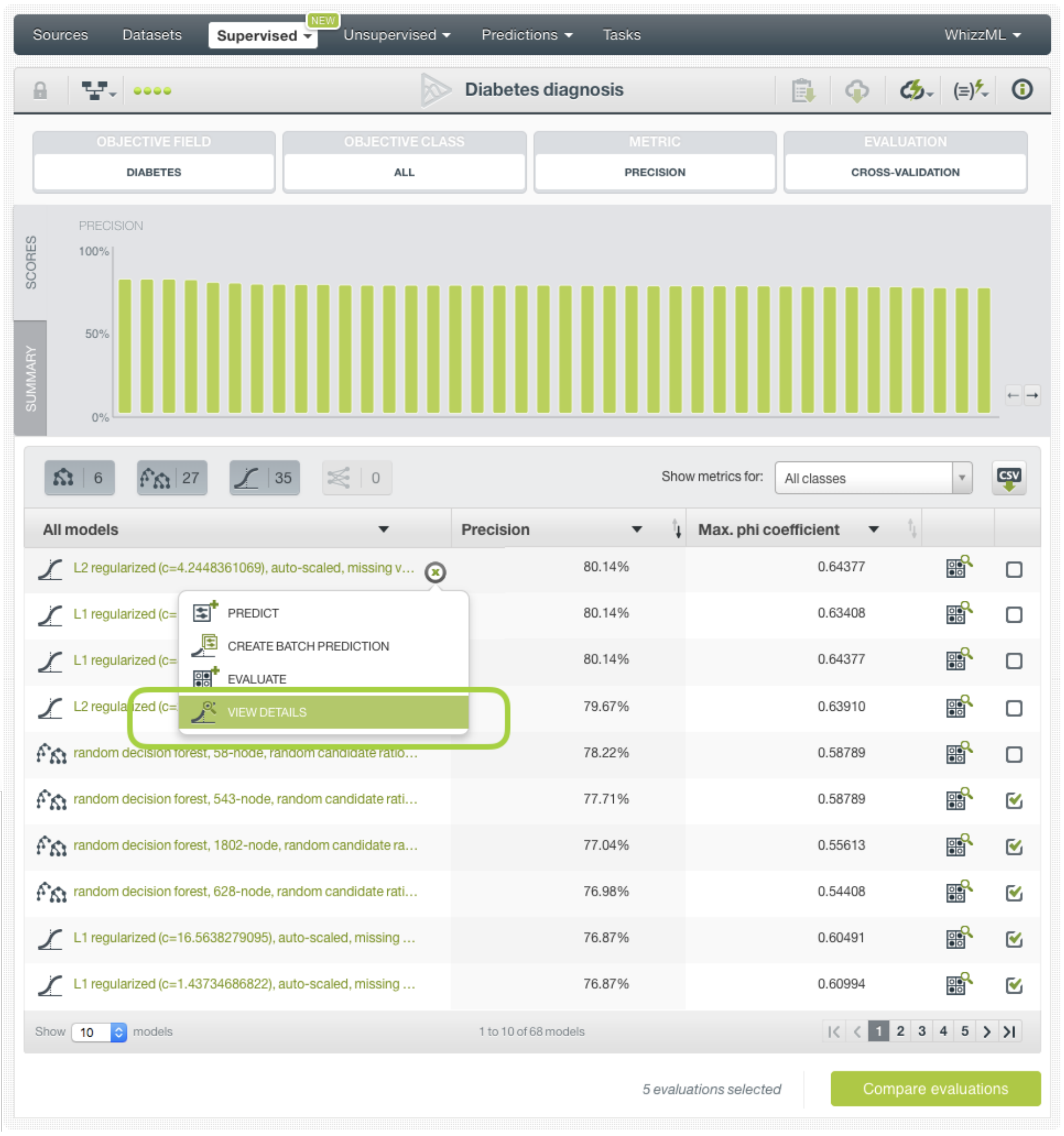


Diabetes diagnosis vs. Diabetes diagnosis
L1 regularized (c=12.1257295012), auto-scaled, missing va…
Diabetes diagnosis vs. Diabetes diagnosis
L1 regularized (c=8.32662483305), auto-scaled, missing va…
Diabetes diagnosis vs. Diabetes diagnosis
L2 regularized (c=4.2448361069), auto-scaled, missing val…
Diabetes diagnosis vs. Diabetes diagnosis

Figure 5.11: Compare evaluations from an OptiML

5.0.3 OptiML Selected Models

You can further **inspect the models** that compose an OptiML by visualizing the model in the BigML Dashboard. You can access the model view taking different paths: clicking on the model bar in the score chart, clicking on the model name from the table, or selecting the option VIEW DETAILS highlighted in Figure 5.12.

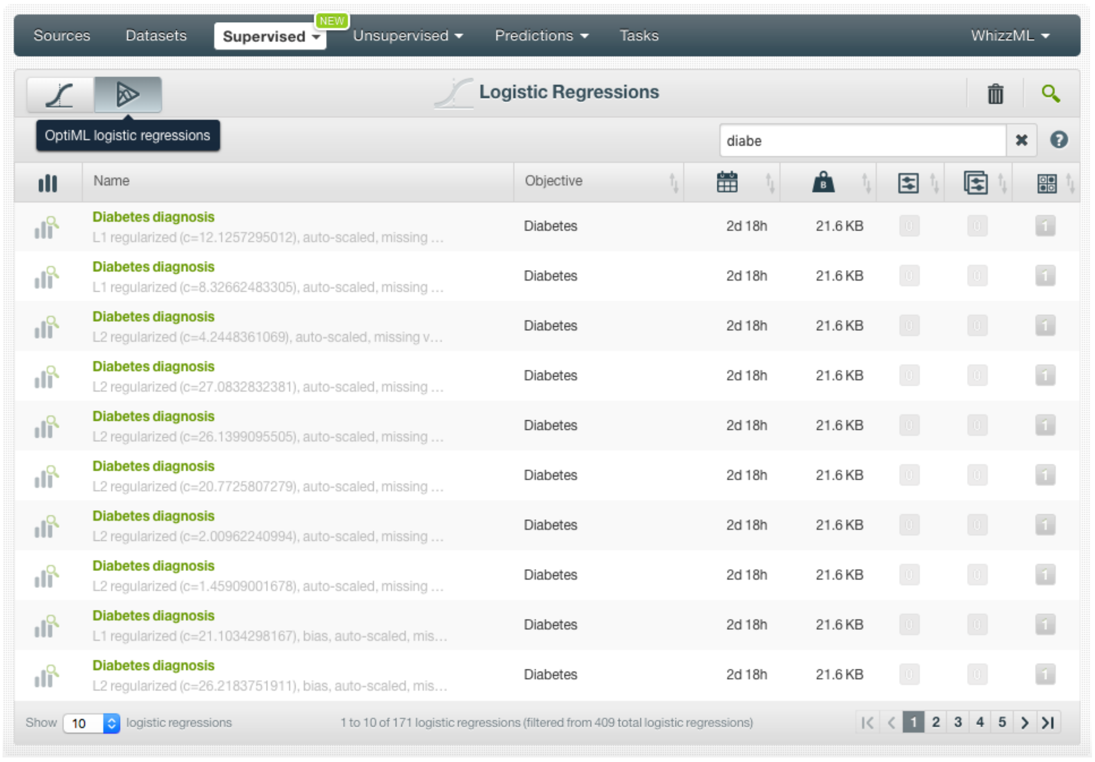


All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v…	80.14%	0.64377
L1 regularized (c=	80.14%	0.63408
L1 regularized (c=	80.14%	0.64377
L2 regularized (c=	79.67%	0.63910
random decision forest, 58-node, random candidate ratio…	78.22%	0.58789
random decision forest, 543-node, random candidate rati…	77.71%	0.58789
random decision forest, 1802-node, random candidate ra…	77.04%	0.55613
random decision forest, 628-node, random candidate rati…	76.98%	0.54408
L1 regularized (c=16.5638279095), auto-scaled, missing …	76.87%	0.60491
L1 regularized (c=1.43734686822), auto-scaled, missing …	76.87%	0.60994

Show 10 models   1 to 10 of 68 models   1 2 3 4 5

Figure 5.12: View the models

You can also find the models in the **OptiML tab** for each resource listing in the Dashboard (see Figure 5.13). This is to keep your Dashboard organized and avoid mixing the dozens of models automatically created with your standard models manually configured. The evaluations and the datasets created during the cross-validation phase are not listed in the Dashboard; you can only access them from the OptiML view.

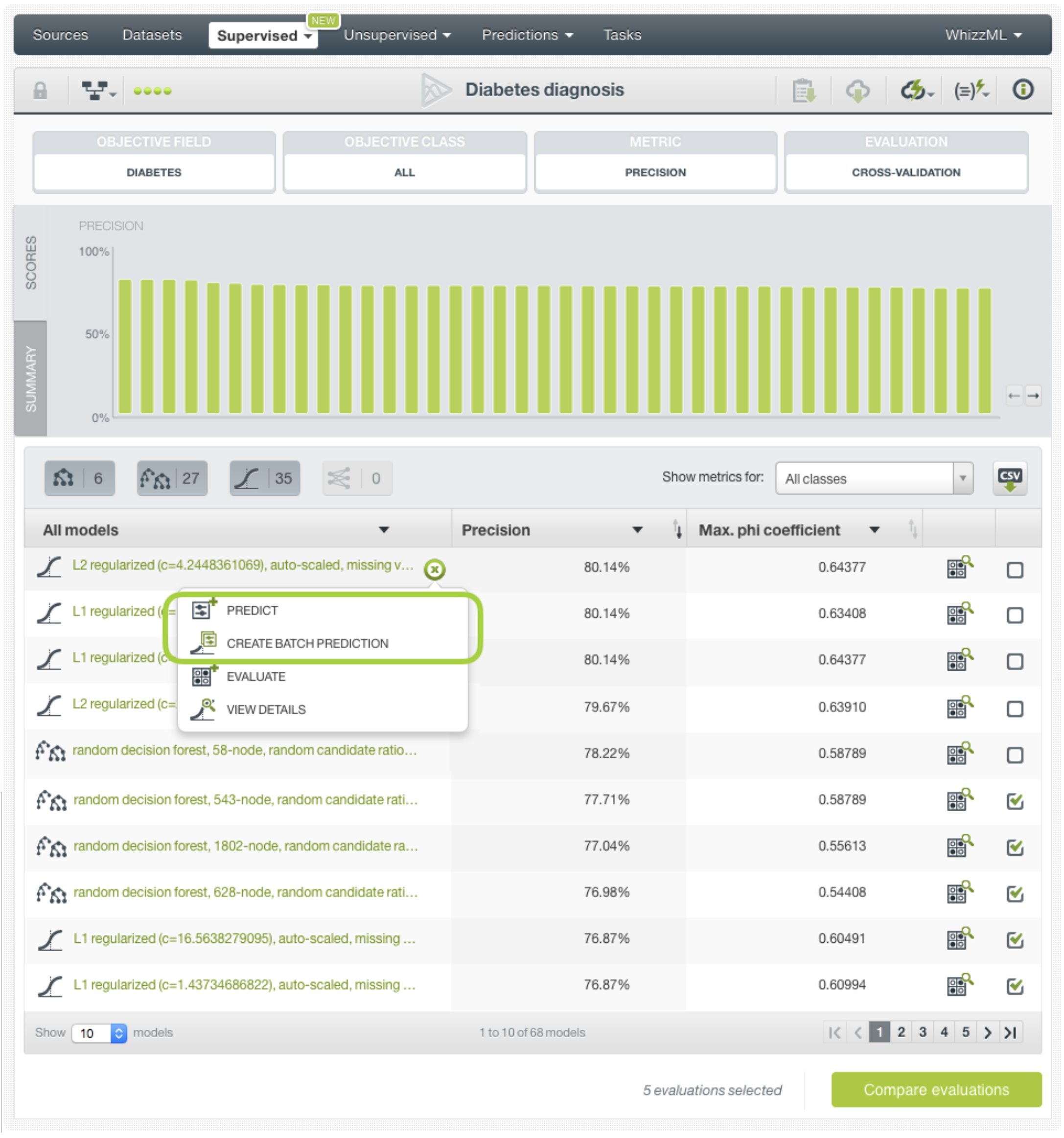


	Name	Objective					
	Diabetes diagnosis L1 regularized (c=12.1257295012), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L1 regularized (c=8.32662483305), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=4.2448361069), auto-scaled, missing v...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=27.0832832381), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=26.1399095505), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=20.7725807279), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=2.00962240994), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=1.45909001678), auto-scaled, missing ...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L1 regularized (c=21.1034298167), bias, auto-scaled, mis...	Diabetes	2d 18h	21.6 KB	0	0	1
	Diabetes diagnosis L2 regularized (c=26.2183751911), bias, auto-scaled, mis...	Diabetes	2d 18h	21.6 KB	0	0	1

Show 10 logistic regressions   1 to 10 of 171 logistic regressions (filtered from 409 total logistic regressions)   1 2 3 4 5

Figure 5.13: Find the models in the Dashboard listings

Finally, the OptiML view offers you a shortcut to start **predicting** with your model. When you have selected the model that best suits your needs, you can make single or batch predictions using the options from the pop-up menu as shown in Figure 5.14. Read more about each model predictions in the document Classification and Regression with the BigML Dashboard [3].



OBJECTIVE FIELD	OBJECTIVE CLASS	METRIC	EVALUATION
DIABETES	ALL	PRECISION	CROSS-VALIDATION

Show metrics for: All classes

All models	Precision	Max. phi coefficient	
L2 regularized (c=4.2448361069), auto-scaled, missing v…	80.14%	0.64377	☐
L1 regularized (c=	80.14%	0.63408	☐
L1 regularized (c	80.14%	0.64377	☐
L2 regularized (c=	79.67%	0.63910	☐
random decision forest, 58-node, random candidate ratio…	78.22%	0.58789	☐
random decision forest, 543-node, random candidate rati…	77.71%	0.58789	✓
random decision forest, 1802-node, random candidate ra…	77.04%	0.55613	✓
random decision forest, 628-node, random candidate rati…	76.98%	0.54408	✓
L1 regularized (c=16.5638279095), auto-scaled, missing …	76.87%	0.60491	✓
L1 regularized (c=1.43734686822), auto-scaled, missing …	76.87%	0.60994	✓

PREDICT
CREATE BATCH PREDICTION
EVALUATE
VIEW DETAILS

Show 10 models 1 to 10 of 68 models 1 2 3 4 5

5 evaluations selected Compare evaluations

Figure 5.14: Make predictions

CHAPTER 6

Consuming OptiML

You can also create and consume your OptiML programmatically via the **BigML API and bindings**. The following subsections explain these options.

6.1 Using OptiML Via the BigML API

OptiML has full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, and delete OptiML.

See in the example below how to create an OptiML using an existing dataset after you have properly set the `BIGML_AUTH` environment variable to contain your authentication credentials:

```
curl "https://bigml.io/optiml?$BIGML_AUTH" \
     -X POST \
     -H 'content-type: application/json' \
     -d '{"dataset": "dataset/50650bdf3c19201b64000322"}'
```

For more information on using OptiML through the BigML API, please refer to the OptiML REST API documentation[1].

6.2 Using OptiML Via the BigML Bindings

You can also create and use OptiML via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your programming language of choice. BigML offers bindings in multiple languages including Python, Node.js, Java, Swift, and Objective-C. See below an example to create an OptiML with the BigML Python bindings.

```
from bigml.api import BigML
api = BigML()
optiml = api.create_optiml('dataset/57506c472275c1666b004b10')
```

For more information on BigML bindings, please refer to the dedicated bindings page[2].

[1] https://bigml.com/api/optiml
[2] https://bigml.com/tools/bindings

CHAPTER 7

OptiML Limits

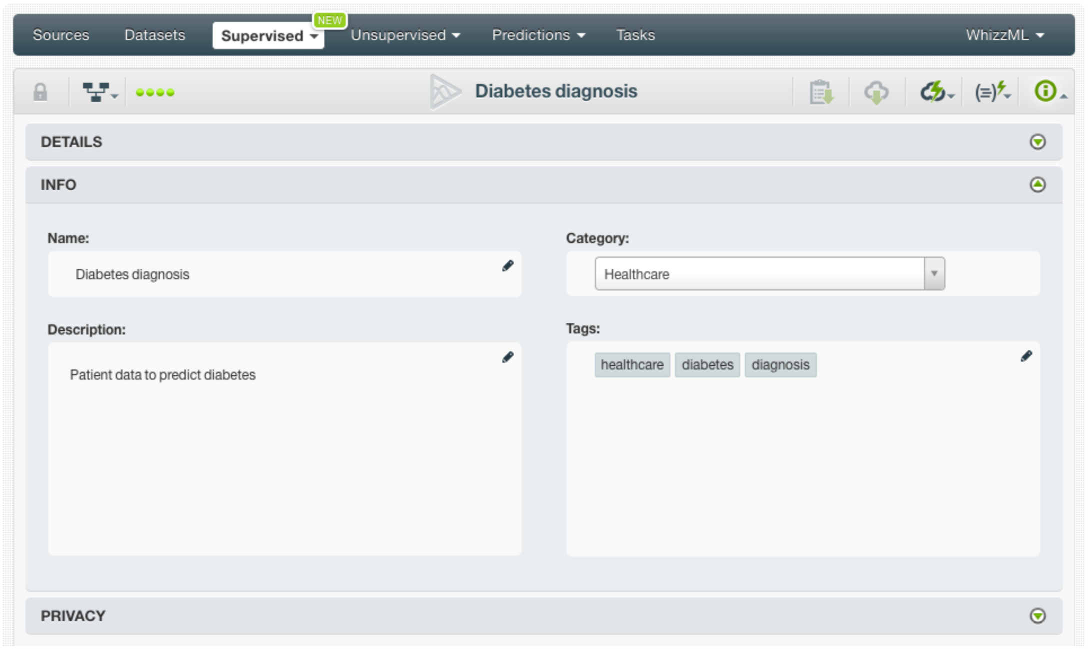
OptiML has a few limitations regarding some of the configurable parameters to optimize models. See the corresponding limits listed below:

- **Model candidates**: a minimum of 4 and a maximum of 200 models can be evaluated during the optimization process.
- **Selected models**: the top performing half of the model candidates will be returned in the OptiML view for you to inspect them. Since there is a maximum of 200 models to evaluate, there is also a maximum of 100 models to be selected for the final output.
- **Selected deepnets**: since deepnets have two specific optimization options, BigML will run those in the background and only return a maximum of two deepnets, one per option, along with the other selected models.
- **Number of classes**: BigML will return per-class metrics in the OptiML main view up to a maximum of 16 classes in the objective field.
- **Moving OptiML**: you cannot move an OptiML to an organization project because all the models composing the OptiML are also tied to the BigML account under which the OptiML was originally created.

CHAPTER 8

OptiML Descriptive Information

Each OptiML has an associated **name**, **description**, **category**, and **tags**. The following subsections briefly describe each concept. See in Figure 8.1 the options under the MORE INFO menu to edit OptiML.



Name: Diabetes diagnosis

Category: Healthcare

Description: Patient data to predict diabetes

Tags: healthcare diabetes diagnosis

Figure 8.1: Editing OptiML

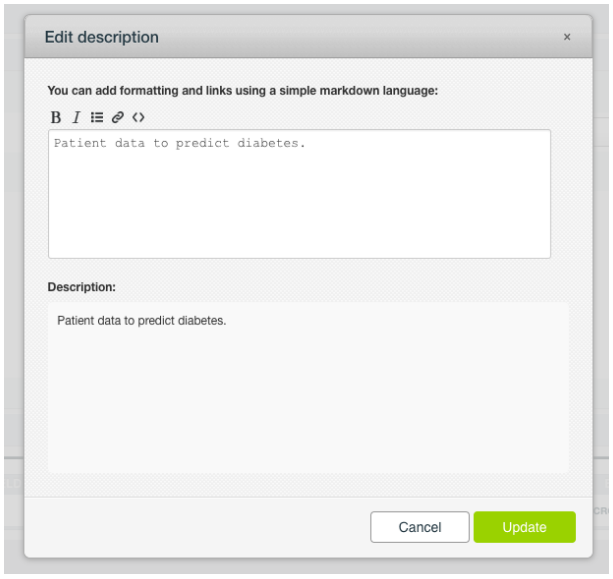
8.1 OptiML Name

Each OptiML has a name that is displayed in the OptiML list view and also on the top bar of the OptiML view. OptiML names are indexed to be used in searches. When you create an OptiML, it gets a default name (the dataset name). You can change it using the MORE INFO menu option in the right corner of the OptiML view. The name of an OptiML cannot be longer than **256** characters. More than one OptiML can have the same name even within the same project, but they will always have different identifiers.

8.2 Description

Each OptiML has a **description** that is very useful for documenting your Machine Learning projects. OptiML takes the description of the datasets used to create them by default.

Descriptions can be written using plain text and markdown[1]. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See Figure 8.2.)



You can add formatting and links using a simple markdown language:

```
Patient data to predict diabetes.
```

Description:

Patient data to predict diabetes.

Figure 8.2: Markdown editor for OptiML descriptions

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

8.3 Category

A **category**, taken from the dataset used to create it, is associated with each OptiML. Categories help to classify an OptiML according to the domain that your data comes from, which is useful when you use BigML to solve problems across industries or for multiple customers.

An otpimization category must be one of the **24** categories listed in Table 8.1.

[1] https://en.wikipedia.org/wiki/Markdown

Table 8.1: Categories used to classify OptiML by BigML

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

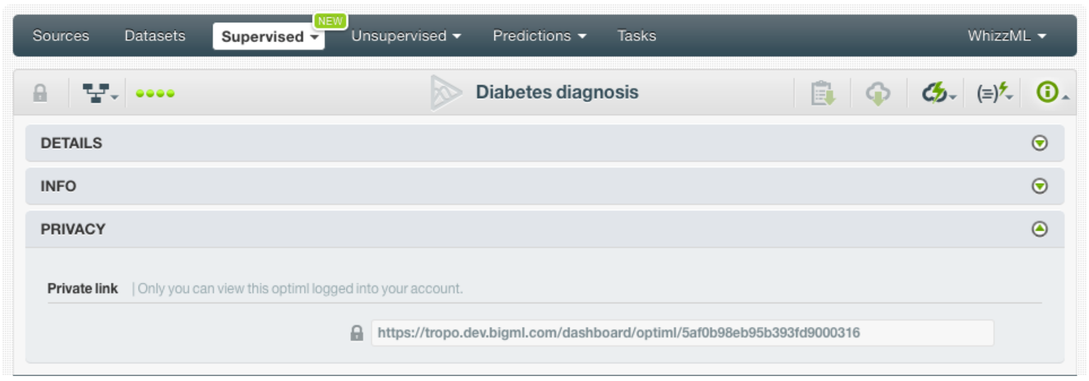
8.4 Tags

An OptiML can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or the Dashboard. OptiML inherits the tags from the dataset used to create them. Each tag is limited to a maximum of 128 characters. Each OptiML can have up to **32** different tags.

CHAPTER 9

OptiML Privacy

Privacy options for OptiML can be defined in the MORE INFO menu option. (See Figure 9.1.) For now, all OptiML are only **private**, i.e., accessible by authorized users (the owner and those who have been granted access).



Only you can view this optiml logged into your account.

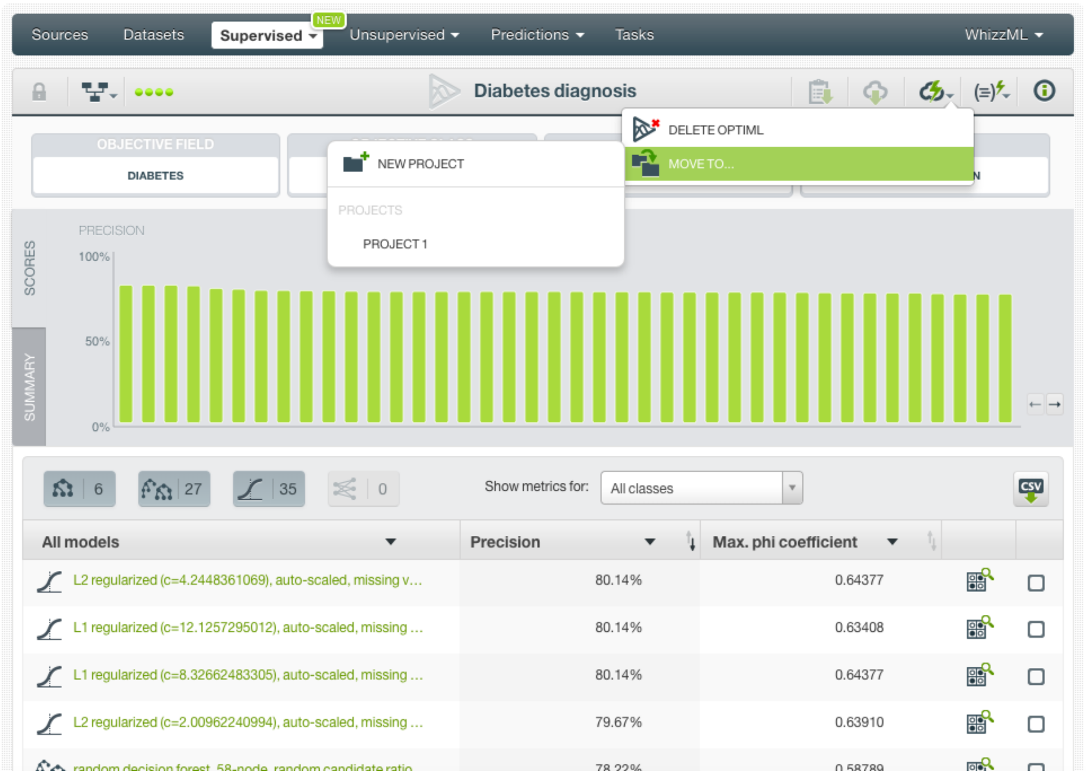
Figure 9.1: OptiML privacy

CHAPTER 10

Moving OptiML

When you create an OptiML, it will be assigned to the same project where the original dataset is located. OptiML can only be assigned to a single project. However, you can move OptiML between projects within your Dashboard. The menu option for this can be found in two places:

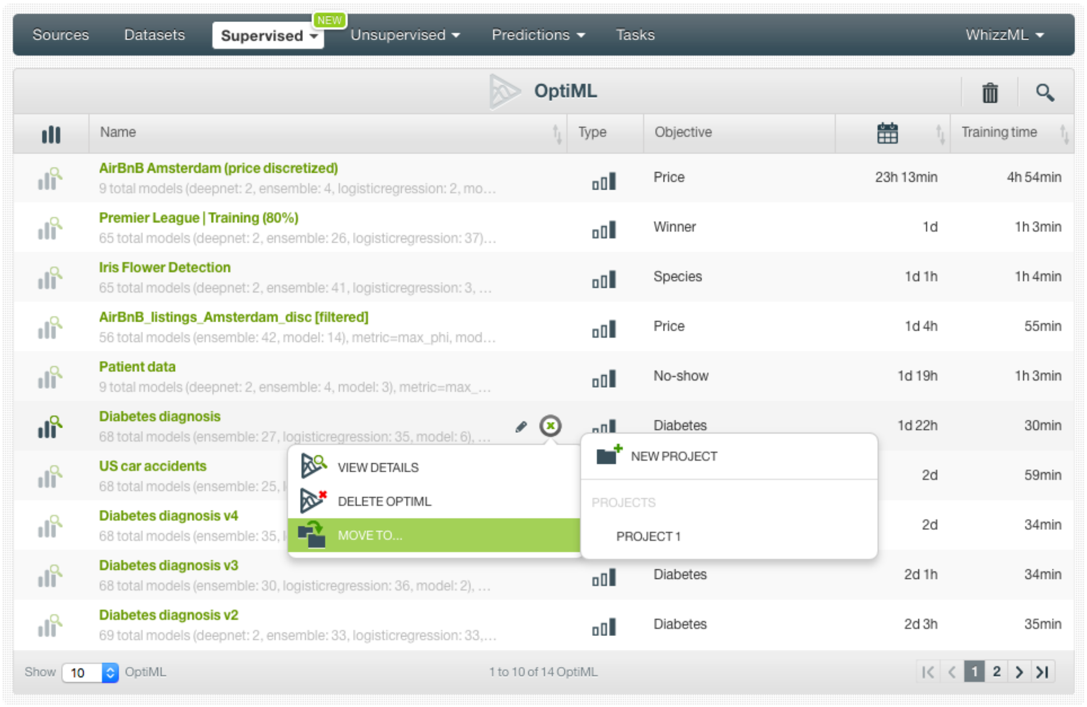
1. Click MOVE TO... within the **1-click action menu** from the optimization view. (See Figure 10.1.)



All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v...	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing ...	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910
random decision forest, 58-node, random candidate ratio	78.22%	0.58789

Figure 10.1: Change project from 1-click menu

2. Click MOVE TO... within the **pop up menu** from the optimization list view. (See Figure 10.2)



Name	Type	Objective		Training time
AirBnB Amsterdam (price discretized) 9 total models (deepnet: 2, ensemble: 4, logisticregression: 2, mo…		Price	23h 13min	4h 54min
Premier League | Training (80%) 65 total models (deepnet: 2, ensemble: 26, logisticregression: 37)…		Winner	1d	1h 3min
Iris Flower Detection 65 total models (deepnet: 2, ensemble: 41, logisticregression: 3, …		Species	1d 1h	1h 4min
AirBnB_listings_Amsterdam_disc [filtered] 56 total models (ensemble: 42, model: 14), metric=max_phi, mod…		Price	1d 4h	55min
Patient data 9 total models (deepnet: 2, ensemble: 4, model: 3), metric=max_…		No-show	1d 19h	1h 3min
Diabetes diagnosis 68 total models (ensemble: 27, logisticregression: 35, model: 6), …		Diabetes	1d 22h	30min
US car accidents 68 total models (ensemble: 25, l			2d	59min
Diabetes diagnosis v4 68 total models (ensemble: 35, l			2d	34min
Diabetes diagnosis v3 68 total models (ensemble: 30, logisticregression: 36, model: 2), …		Diabetes	2d 1h	34min
Diabetes diagnosis v2 69 total models (deepnet: 2, ensemble: 33, logisticregression: 33,…		Diabetes	2d 3h	35min

Figure 10.2: Change project from pop up menu

You can also move the single models composing the OptiML across projects within your Dashboard.

Note: you cannot move an OptiML or the single models composing it to an organization project.

CHAPTER 11

Stopping OptiML

You can stop the creation of an OptiML before the task is finished by clicking the DELETE OPTIML option in the **1-click action menu** from the OptiML view. (See Figure 11.1.)

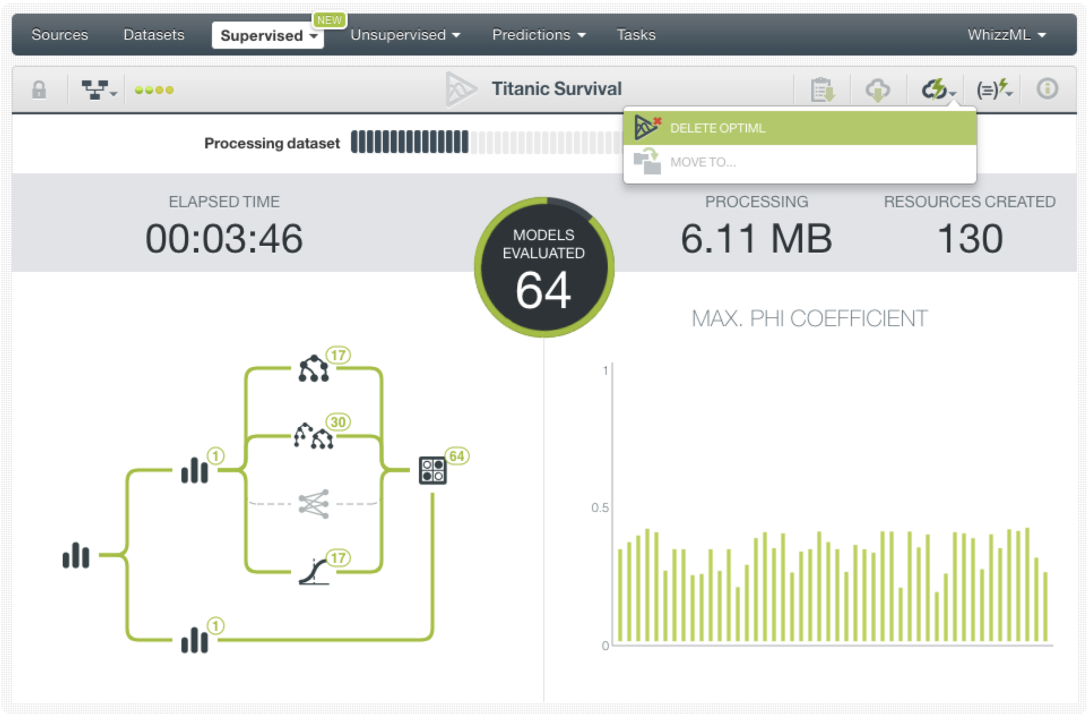
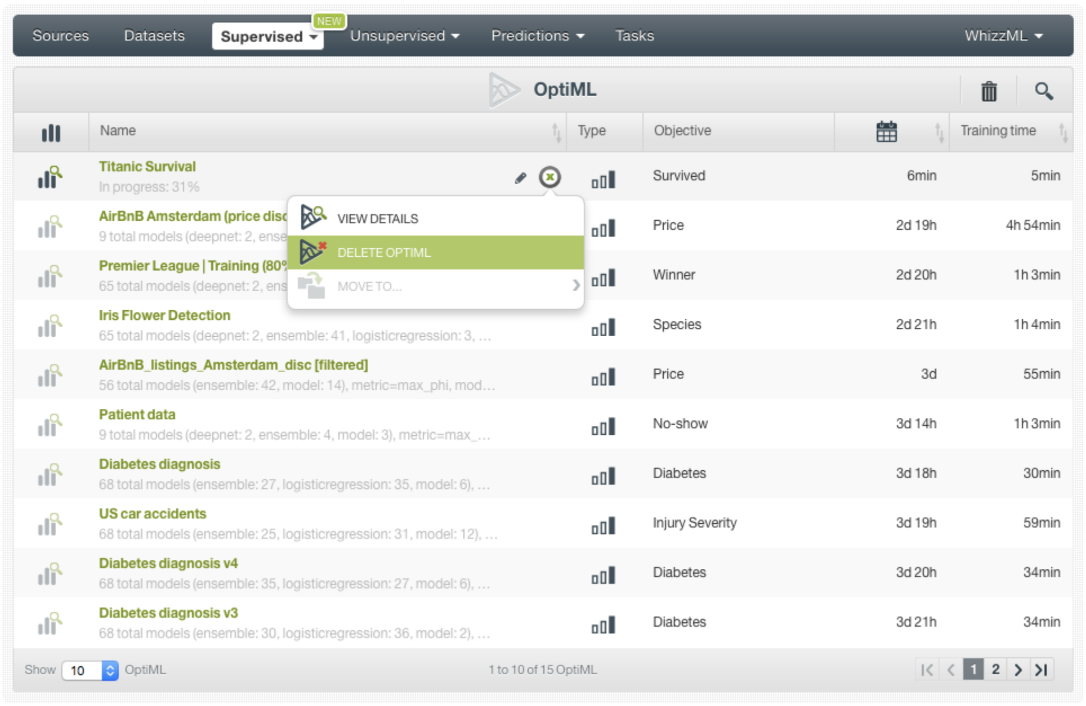


Figure 11.1: Stop OptiML creation from 1-click menu

Alternatively, click the DELETE OPTIML in the **pop up menu** from the OptiML list view. (See Figure 11.2.)



Name	Type	Objective		Training time
Titanic Survival In progress: 31%		Survived	6min	5min
AirBnB Amsterdam (price disc 9 total models (deepnet: 2, ense		Price	2d 19h	4h 54min
Premier League | Training (80° 65 total models (deepnet: 2, ens		Winner	2d 20h	1h 3min
Iris Flower Detection 65 total models (deepnet: 2, ensemble: 41, logisticregression: 3, ...		Species	2d 21h	1h 4min
AirBnB_listings_Amsterdam_disc [filtered] 56 total models (ensemble: 42, model: 14), metric=max_phi, mod...		Price	3d	55min
Patient data 9 total models (deepnet: 2, ensemble: 4, model: 3), metric=max_...		No-show	3d 14h	1h 3min
Diabetes diagnosis 68 total models (ensemble: 27, logisticregression: 35, model: 6), ...		Diabetes	3d 18h	30min
US car accidents 68 total models (ensemble: 25, logisticregression: 31, model: 12), ...		Injury Severity	3d 19h	59min
Diabetes diagnosis v4 68 total models (ensemble: 35, logisticregression: 27, model: 6), ...		Diabetes	3d 20h	34min
Diabetes diagnosis v3 68 total models (ensemble: 30, logisticregression: 36, model: 2), ...		Diabetes	3d 21h	34min

Show 10 OptiML 1 to 10 of 15 OptiML 1 2

Figure 11.2: Stop OptiML creation from pop up menu

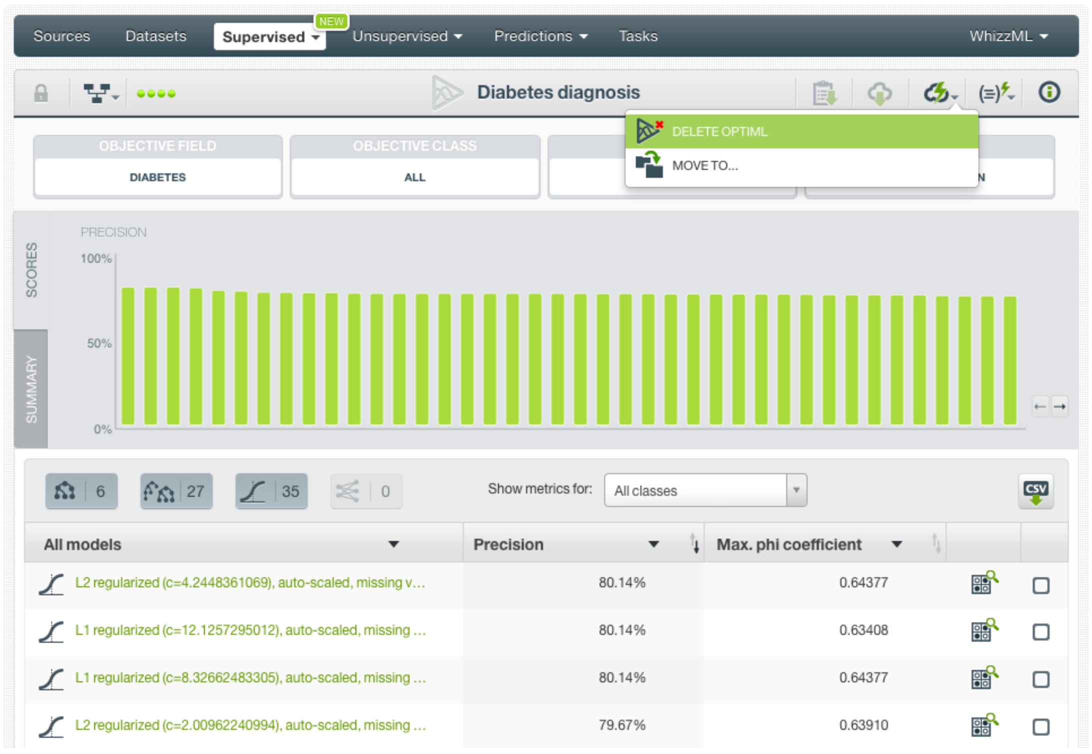
A modal window will be displayed asking you if you want to delete also the resoruces that the OptiML has created so far as shown in Figure 12.3.

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

CHAPTER 12

Deleting OptiML

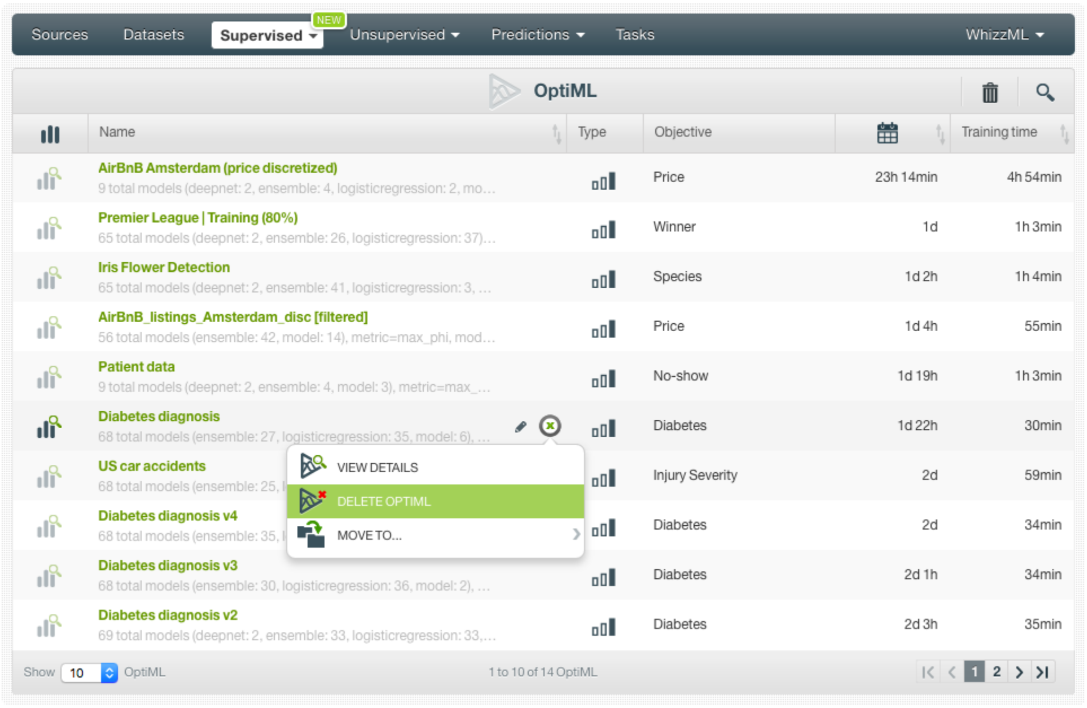
You can delete your OptiML by clicking the DELETE OPTIML option in the **1-click action menu** from the OptiML view. (See Figure 12.1.)



All models	Precision	Max. phi coefficient
L2 regularized (c=4.2448361069), auto-scaled, missing v...	80.14%	0.64377
L1 regularized (c=12.1257295012), auto-scaled, missing ...	80.14%	0.63408
L1 regularized (c=8.32662483305), auto-scaled, missing ...	80.14%	0.64377
L2 regularized (c=2.00962240994), auto-scaled, missing ...	79.67%	0.63910

Figure 12.1: Delete OptiML from 1-click menu

Alternatively, click the DELETE OPTIML in the **pop up menu** from the OptiML list view. (See Figure 12.2.)

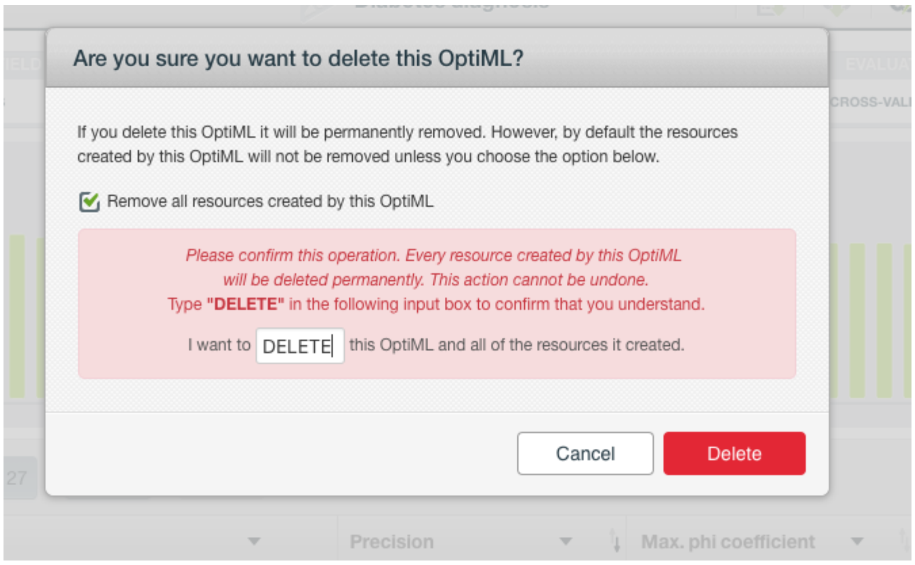


Name	Type	Objective		Training time
AirBnB Amsterdam (price discretized) 9 total models (deepnet: 2, ensemble: 4, logisticregression: 2, mo...		Price	23h 14min	4h 54min
Premier League | Training (80%) 65 total models (deepnet: 2, ensemble: 26, logisticregression: 37)...		Winner	1d	1h 3min
Iris Flower Detection 65 total models (deepnet: 2, ensemble: 41, logisticregression: 3, ...		Species	1d 2h	1h 4min
AirBnB_listings_Amsterdam_disc [filtered] 56 total models (ensemble: 42, model: 14), metric=max_phi, mod...		Price	1d 4h	55min
Patient data 9 total models (deepnet: 2, ensemble: 4, model: 3), metric=max_...		No-show	1d 19h	1h 3min
Diabetes diagnosis 68 total models (ensemble: 27, logisticregression: 35, model: 6), ...		Diabetes	1d 22h	30min
US car accidents 68 total models (ensemble: 25,		Injury Severity	2d	59min
Diabetes diagnosis v4 68 total models (ensemble: 35,		Diabetes	2d	34min
Diabetes diagnosis v3 68 total models (ensemble: 30, logisticregression: 36, model: 2), ...		Diabetes	2d 1h	34min
Diabetes diagnosis v2 69 total models (deepnet: 2, ensemble: 33, logisticregression: 33,...		Diabetes	2d 3h	35min

Show 10 OptiML 1 to 10 of 14 OptiML 1 2

Figure 12.2: Delete OptiML from pop up menu

A modal window will be displayed asking if you also want to delete the single models composing the OptiML. If you choose to delete the models, they will be permanently deleted, and there is no way you (or even the IT folks at BigML) can retrieve it. If you choose to keep the models, the OptiML will be deleted, but you can still find the single models in the corresponding tab in the resource listings in your Dashboard (see Subsection 5.0.3.)



Are you sure you want to delete this OptiML?

If you delete this OptiML it will be permanently removed. However, by default the resources created by this OptiML will not be removed unless you choose the option below.

✓ Remove all resources created by this OptiML

Please confirm this operation. Every resource created by this OptiML will be deleted permanently. This action cannot be undone.
Type **"DELETE"** in the following input box to confirm that you understand.

I want to DELETE this OptiML and all of the resources it created.

Figure 12.3: Confirmation message to delete an OptiML and its resources

Note: you cannot delete the single models that belong to an OptiML unless you delete the OptiML.

CHAPTER 13

Takeaways

This document covered OptiML in detail. We conclude it with a list of key points:

- OptiML is an **automated optimization process** for **model selection and parametrization** (or hyperparametrization) to solve classification and regression problems.
- OptiML uses Bayesian parameter optimization to search for the top performing algorithm and parametrization.
- Decision trees, ensembles, logistic regressions and deepnets will be tried during the model search.
- OptiML can create and evaluate up to 200 different models. The top performing half of these models will be returned so you can further inspect them and select the most suitable for your use case.
- OptiML just needs an existing **dataset** containing a numeric or categorical field to be selected as the objective field.
- You can use the **1-click OptiML option** or you can **configure** the several parameters provided by BigML.
- When the optimization process finishes, BigML provides different performance metrics by model so you can select the model or models that best suits your use case.
- You can find all the models automatically created by the OptiML in the Dashboard in a separate tab in each resource listing.
- You can select one or several models from the OptiML results and make predictions with them.
- You cannot delete the single models composing the OptiML output unless you remove the OptiML.
- You can create, configure, update, and use your OptiML programmatically via the **BigML API and bindings**.
- You can add **descriptive information** to your OptiML.
- You can **move** your OptiML between projects within the Dashboard, but not to an organization project.
- You can **stop** your OptiML by deleting it.
- You can permanently **delete** your existing OptiML and choose if you want to delete all models or just de-reference them.

```
Please use the noidx option in the documentclass invocation.
```

List of Figures

List of Tables

Glossary

Classification a modeling task whose objective field (i.e., the field being predicted) is categorical and predicts classes. ii, 1, 3, 47

Dashboard The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. 1

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. 3, 9

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 per-class probabilities forclassification problems or a single, real value for regression problems. 3, 9

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. 3, 9

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. 3, 9

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. 16

Monte Carlo cross-validation Monte Carlo cross-validation, randomly splits n number of times the dataset into training and test subsets. For each split, the model is built using the training data and evaluated using the test data. The advantage of this method (over k-fold cross validation) is that the proportion of the training and validation subsets is not dependent on the number of iterations (k-folds). The disadvantage of this method is that some subsets may overlap while some instances may not be selected in any subset. 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. 3, 10

Objective Field the field that a regression or classification model will predict (also known as target). 7, 14

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. 36, 42

Project an abstract resource that helps you group related BigML resources together. 2, 41

Regression a modeling task whose objective field (i.e., the field being predicted) is numeric. ii, 1, 3, 47

Standard Deviation in the OptiML result, each model has a value for the optimization metric and a standard deviation associated. The standard deviation indicates the potential variation of the optimization metric depending on the random split of the dataset to train and evaluate the model. Therefore, the range of potential values for the optimized metric can be calculated +/- the standard deviation: if a model has a 70% accuracy with an standard deviation of 5%, the accuracy may take any value between 65% and 75%. The standard deviation is calculated taking into account the different values of the optimization metric achieved by the model during cross-validation. 4, 23

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, 3

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

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

References

[1] The BigML Team. *Anomaly Detection with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.

[2] The BigML Team. *Association Discovery with the BigML Dashboard*. Tech. rep. BigML, Inc., Dec. 2015.

[3] The BigML Team. *Classification and Regression with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.

[4] The BigML Team. *Cluster Analysis with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.

[5] The BigML Team. *Datasets with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.

[6] The BigML Team. *Sources with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.

[7] The BigML Team. *Time Series with the BigML Dashboard*. Tech. rep. BigML, Inc., July 2017.

[8] The BigML Team. *Topic Models with the BigML Dashboard*. Tech. rep. BigML, Inc., Nov. 2016.

[9] 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.