

PCA with the BigML Dashboard

The BigML Team

Version 1.1



MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

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](https://creativecommons.org/licenses/by-nc-nd/4.0/). Based on work at <http://bigml.com>.

Last updated June 7, 2024

About this Document

This document provides a comprehensive description of how to reduce your dataset dimensionality using [PCA](#) with the [BigML Dashboard](#). Learn how to use the BigML Dashboard to configure, visualize, and interpret this [unsupervised](#) model and use it to transform high-dimensional datasets.

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](#). The BigML Team. June 2016. [3]
- [Time Series with the BigML Dashboard](#). The BigML Team. July 2017. [7]

To learn how to use the BigML Dashboard to build other unsupervised models read:

- [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

- 1 Introduction** **1**

- 2 Understanding PCA** **3**
 - 2.1 Principal Component Analysis 3
 - 2.2 Missing Values 4
 - 2.3 Dimensionality Reduction 4
 - 2.4 Projections 4
 - 2.5 Visualization in the Scatterplot 4

- 3 Creating PCA with 1-Click** **6**

- 4 PCA Configuration Options** **8**
 - 4.1 Standardize 8
 - 4.2 Default Numeric Value 9
 - 4.3 Sampling Options 9
 - 4.3.1 Rate 9
 - 4.3.2 Range 9
 - 4.3.3 Sampling 10
 - 4.3.4 Replacement 10
 - 4.3.5 Out of Bag 10
 - 4.4 Creating PCA with Configured Options 10
 - 4.5 API Request Preview 11

- 5 Visualizing PCA** **12**
 - 5.1 Create a Dataset From PCA 17

- 6 PCA Predictions: Projections** **19**
 - 6.1 Introduction 19
 - 6.2 Creating Projections 20
 - 6.3 Configuring Projections 24
 - 6.3.1 Limit the Number of Components 24
 - 6.3.2 Default Numeric Value 24
 - 6.3.3 Field Mapping 25
 - 6.3.4 Output Settings 26
 - 6.4 Visualizing Projections 26
 - 6.5 Consuming Projections 29
 - 6.5.1 Using Projections Via the BigML API 29
 - 6.5.2 Using Projections Via the BigML Bindings 29
 - 6.6 Descriptive Information 30
 - 6.6.1 Projection Name 30
 - 6.6.2 Description 30
 - 6.6.3 Category 31
 - 6.6.4 Tags 32

6.7	Projection Privacy	32
6.8	Moving Projections	33
6.9	Stopping Projections Creation	33
6.10	Deleting Projections	34
7	Consuming PCA	36
7.1	Downloading PCA	36
7.2	Using PCA Via the BigML API	37
7.3	Using PCA Via the BigML Bindings	37
8	PCA Limits	38
9	PCA Descriptive Information	39
9.1	PCA Name	39
9.2	Description	39
9.3	Category	40
9.4	Tags	41
9.5	Counters	41
10	PCA Privacy	43
11	Moving PCA	44
12	Stopping PCA Creation	46
13	Deleting PCA	48
14	Takeaways	50
	List of Figures	53
	List of Tables	55
	Glossary	56
	References	57

Introduction

Many datasets contain an extremely **large number of fields**, or highly **correlated fields**, resulting in suboptimal model performance. Principal component analysis (PCA) is one technique that can be used to transform a dataset in order to yield uncorrelated variables or as a first step in dimensionality reduction. Because PCA transforms the variables in a dataset without accounting for a target variable, it can also be considered an **unsupervised** Machine Learning method.

The BigML implementation of **PCA** is distinct from other approaches in that **it can handle non-numerical data types**, including **categorical** and **text** data, as well as **combinations of different data types**. This allows PCA to be performed on essentially any type of dataset with minimal preprocessing and data cleaning efforts. BigML uses a combination of **Principal Component Analysis (PCA)**¹, **Multiple Correspondence Analysis (MCA)**², and **Factorial Analysis of Mixed Data (FAMD)**³ to deal with different data types. The selection of method is performed automatically without input from the user. The effective difference between the three methods is how the data rows are transformed in order to populate the data matrix. After the matrix is populated, the same decomposition is performed in order to produce the final results.

This document provides a comprehensive description of BigML PCA including how they can be created (**Chapter 3**) and configured (**Chapter 4**). Visualizations of the results of a PCA transformation are provided to give insight into the composition of individual principal components and the proportion of variance explained (see **Chapter 5**). You can use the result of PCA to transform another dataset which contains the same fields, such as datasets resulting from a train/test split (**Chapter 6**). Finally, it is possible to download and perform PCA either locally or you can create, update, list, and delete PCA resources using the BigML API (**Chapter 7**).

In BigML, the “**Unsupervised**” tab of the main menu of your Dashboard allows you to list all of your previously created PCA resources. In the PCA list view (**Figure 1.1**), you can see, for each PCA resource, the **dataset** it was created from, as well as the PCA’s **Name**, the number of **Components**, whether or not it was **Standardized**, the **Age** (time elapsed since it was created), **Size**, and number of **batch projections** that have been created using that PCA. The SEARCH menu option in the top right corner of the PCA list view allows you to search your PCA resources by name.

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

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

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

Name	Date	Size	Status
Arrythmia diagnosis 262 components, standardized	2d	536.2 KB	0
Hotel_reviews standardized	6d 1h	1.2 MB	0
Hotel_reviews standardized	6d 1h	1.2 MB	0
Hotel_reviews standardized	6d 1h	24.2 KB	0
Hotel_reviews standardized	6d 2h	24.2 KB	0
Stumble_upon.csv standardized, default numeric value=mean	6d 2h	9.8 MB	0
Stumble_upon.csv felix standardized	6d 3h	9.8 MB	0
Stumble upon with Stumble_upon.csv standardized	6d 3h	10.6 MB	0
Fictional Wine Sales standardized, default numeric value=mean	6d 7h	50.4 KB	0
Fictional Wine Sales standardized	6d 8h	50.4 KB	1

Figure 1.1: PCA list view

When you first create an account at BigML, or every time that you start a new **project**, your list of PCAs will be empty. (See [Figure 1.2.](#))

Name	Date	Size	Status
No pcas			

Figure 1.2: PCA empty list view in the BigML Dashboard

Finally, in [Figure 1.3](#) you can see the icon used to represent a PCA in BigML.



Figure 1.3: PCA icon

Understanding PCA

This section describes the internal details of BigML **principal component analysis (PCA)**. Beyond a description of the implementation, it also provides information on how to properly make use of PCA in different scenarios, as it can be used as a **feature transformation method**, a **dimensionality reduction technique**, and a **exploratory data analysis tool**.

2.1 Principal Component Analysis

Principal component analysis (**PCA**) is a statistical technique that **transforms** a dataset defined by possibly **correlated variables** into a set of **uncorrelated variables**, called **principal components**. Each of the principal components is a **linear combination** of the original variables, is **orthogonal**¹ to all other components, and ordered according to the amount of **variance** explained. PCA preserves the dimensionality of the original dataset and is sensitive to the scaling of the variables in the original dataset.

BigML PCA incorporates **multiple factor analysis techniques**, rather than only the standard PCA implementation. Specifically, BigML utilizes **Multiple Correspondence Analysis (MCA)** if the input contains **only categorical data** and **Factorial Analysis of Mixed Data (FAMD)** if the input contains **both numeric and categorical fields**. In the case of **items** and **text** fields, data is processed using a **bag-of-words approach** allowing PCA to be applied. In MCA, categorical variables are first transformed to **one-hot**² encoded binary vectors. The order of the categorical values matches those found in the field summary. For example, a value of “Iris-virginica” for the “Species” field in the “Iris dataset” becomes [0,0,1], because “Iris-virginica” is listed third in categories. If the categorical field contains missing values, then an additional element will be appended to the end of this vector to represent missing values. For each categorical value, define a value p_j which represents the proportion of the data that contains that value. Continuing with the Iris example, we would have $p_1 = p_2 = p_3 = 0.3333$ because the Species values are equally represented in the data. Note that these values are equal to the arithmetic mean of the one-hot encoded columns. In FAMD, numeric fields are transformed identically to PCA. Because of this approach, BigML can handle categorical, text, and items fields in addition to numerical data in an automatic fashion that does not require input by the user.

PCA, MCA, and FAMD all utilize the same decomposition method to yield the final outputs, and differ only in how the data is processed when populating the data matrix.

- **PCA**: the values for each numeric field are shifted to zero mean, and divided by the standard deviation of the field if standardization is enabled (see [Section 4.1](#)).
- **MCA**: categorical variables are one-hot encoded with an additional field appended to account for missing values. Mean shifted values are divided by the expression $J \times \sqrt{J \times p_j}$ where J is the number of categorical fields and p_j is the proportion of the data which contains each class.

¹<https://en.wikipedia.org/wiki/Orthogonality>

²<https://en.wikipedia.org/wiki/One-hot>

- **FAMD**: numeric fields are transformed identically to PCA and categorical fields are divided by $\sqrt{p_j}$.

2.2 Missing Values

BigML PCA implementation **supports missing values** at **training** and **prediction** time. To compute a prediction with numeric missing values, the inner product is computed with the missing values set to zero. In the case of missing text or items fields, this is a zero vector the same length as the tag/items cloud. Categorical fields are modeled with an additional categorical value for missing values during model time.

2.3 Dimensionality Reduction

While a PCA transformation maintains the dimensions of the original dataset, it is typically applied with the goal of **dimensionality reduction**. Reducing the dimensions of the feature space is one method to help reduce **supervised** model **overfitting**, as there are fewer relationships between variables to consider. Because the **principal components** yielded by a PCA transformation are linear combinations of the original variables, we are essentially keeping the discriminative aspects of each original variable, even if we reduce the number of dimensions in the new dataset. The principal components yielded by a PCA transformation are ordered by the amount of **variance** each explains in the original dataset. Plots of the cumulative variance explained, also known as scree plots, are one way to interpret appropriate thresholds for how many of the new features can be eliminated from a dataset while preserving most of the original information (see [Chapter 5](#)).

2.4 Projections

PCA models can be used to project new data points to the componential axes. This is done by first centering and scaling the input data using the same input transformations as used while modeling, and then taking the inner products between the transformed input and the loading vectors. Note that for text inputs, the centering and scaling are done using the mean and standard deviation computed at model time, as those values are not present in the dataset field summary. Because PCA is a preprocessing technique for many **supervised** models, it is important to perform the projection on a testing set if PCA is used in the training process.

Projections accept a couple parameters to limit the number of components returned:

- **Cumulative variance**: a number between 0% and 100%. The prediction uses the minimum number of components such that the cumulative explained variance is greater than the given threshold. For example, setting a variance threshold of 95% for the Iris PCA will use only the first two components.
- **Maximum components**: you can limit the number of components by setting an integer greater than 0 for this parameter.

If no parameters are given, the full set of **principal components** is used.

Learn more about projections on [Chapter 6](#).

2.5 Visualization in the Scatterplot

Appart from the BigML visualization of the PCA components (see [Chapter 5](#)), PCA also allow the variance in a dataset to be more easily viewed, as the **principal components** are ordered by the proportion of **variance** explained in the original dataset. By performing a batch **projection** on the original dataset, or another dataset of interest, the resulting dataset can be visualized in the Dashboard using the standard **scatterplot**³. The axes can then be set to the principal components of interest, with PC1 and PC2 being the standard options to view the dimensions that display the greatest variance in the data. For

³<https://bigml.com/whatsnew#scatterplot-in-the-dashboard>

supervised problems, the coloration can be set to the target variable in order to visually explore which principal components are maximally discriminative (see Figure 2.1). Because PCA is an **unsupervised** method and would not include the target variable in the transformation, it is not guaranteed to display separation between classes if the original variables do not contain predictive power.

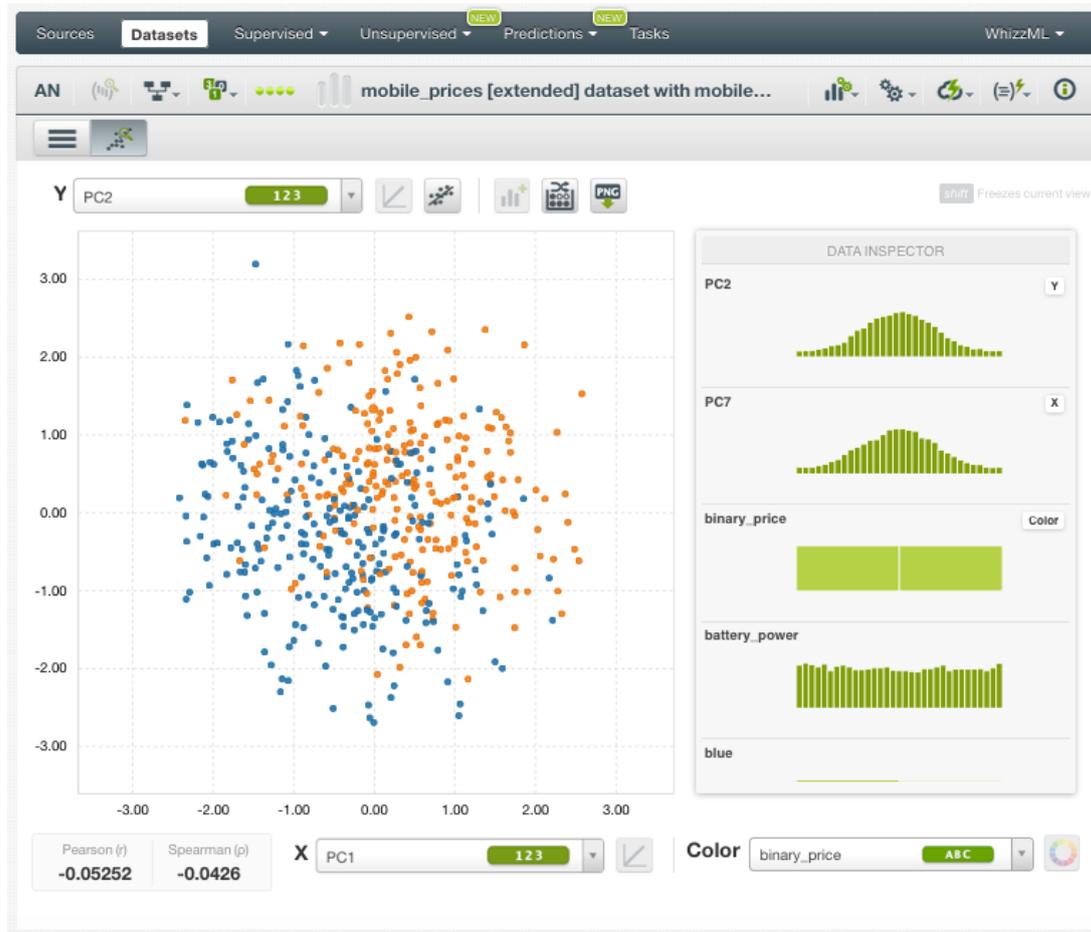


Figure 2.1: PCA visualization in the scatterplot

Creating PCA with 1-Click

To create a PCA in BigML you have two options: you can use the **1-click option** which uses the default values for all available configuration options, or you can tune the parameters in advance using the **configuration option** explained in [Chapter 4](#).

You can find the PCA option in the **1-click menu** from the dataset view. (See [Figure 3.1](#).)

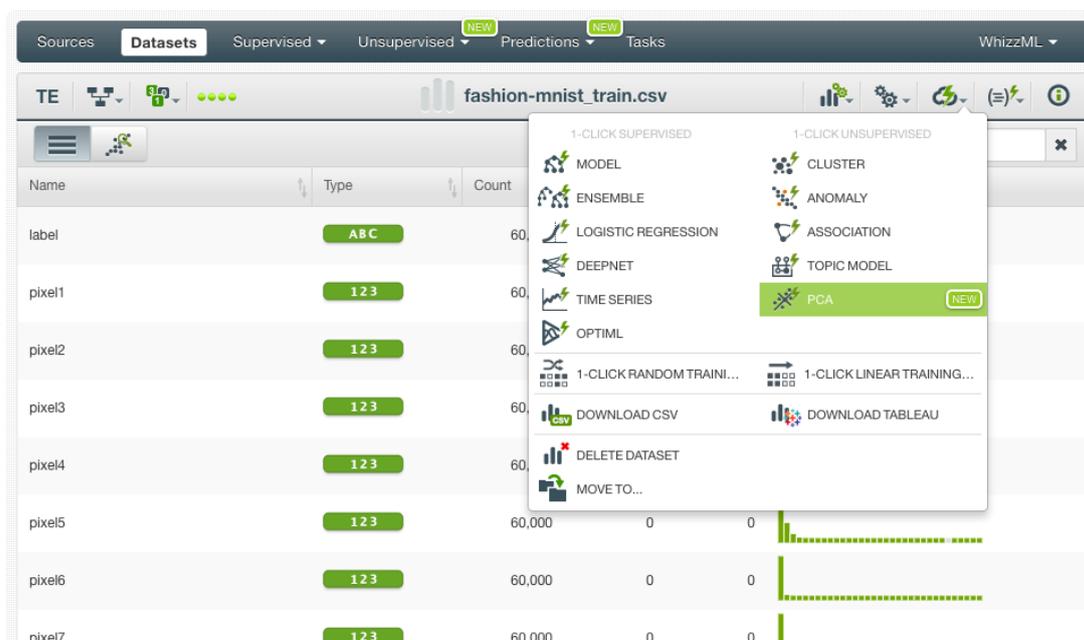


Figure 3.1: Create PCA from 1-click menu

Alternatively, you can use the PCA option in the **pop up menu** from the dataset list view. (See [Figure 3.2](#).)

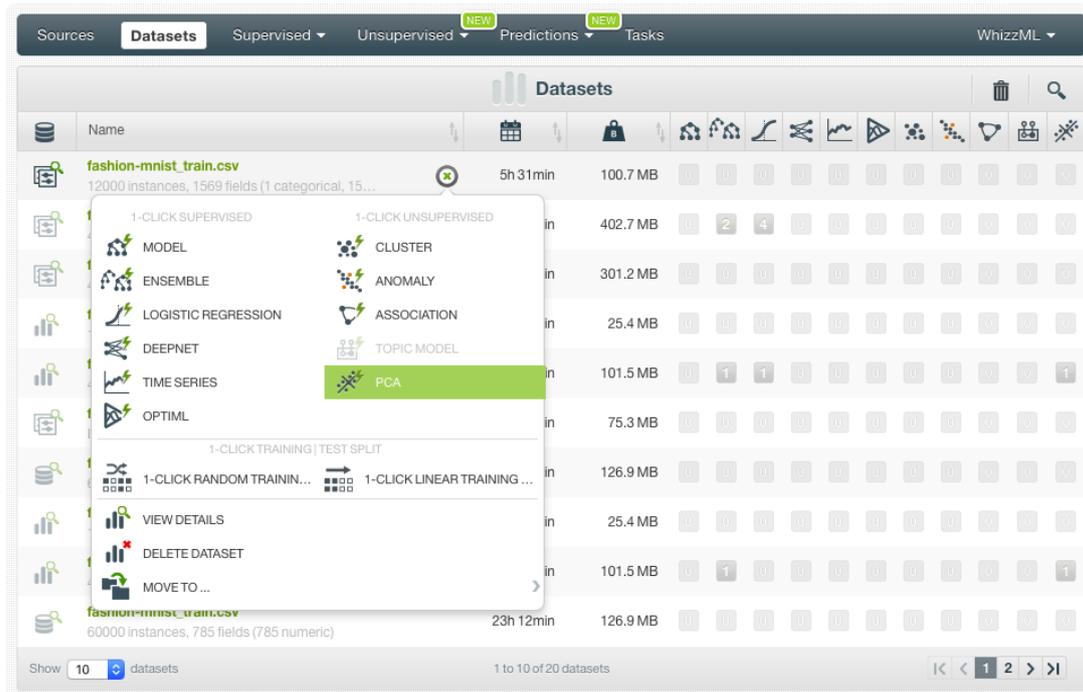


Figure 3.2: Create PCA from popu up menu

Either option builds a PCA using the default values for the available configuration options explained in the following section. (See [Chapter 4](#).)

Note: if you want to use the dataset with the **principal components** to train supervised models afterward, you will need to exclude the **objective field** field to create the PCA; otherwise you will be introducing **leakage**¹, i.e., the information of the objective field will be contained in the principal components to be used as inputs for your models.

¹<https://machinelearningmastery.com/data-leakage-machine-learning/>

PCA Configuration Options

You can configure a few parameters that affect the way BigML creates PCAs. See the following sections for a detailed explanation.

To display the configuration panel to see all options, click the PCA menu option in the **configuration menu** from the dataset view. (See [Figure 4.1](#).)

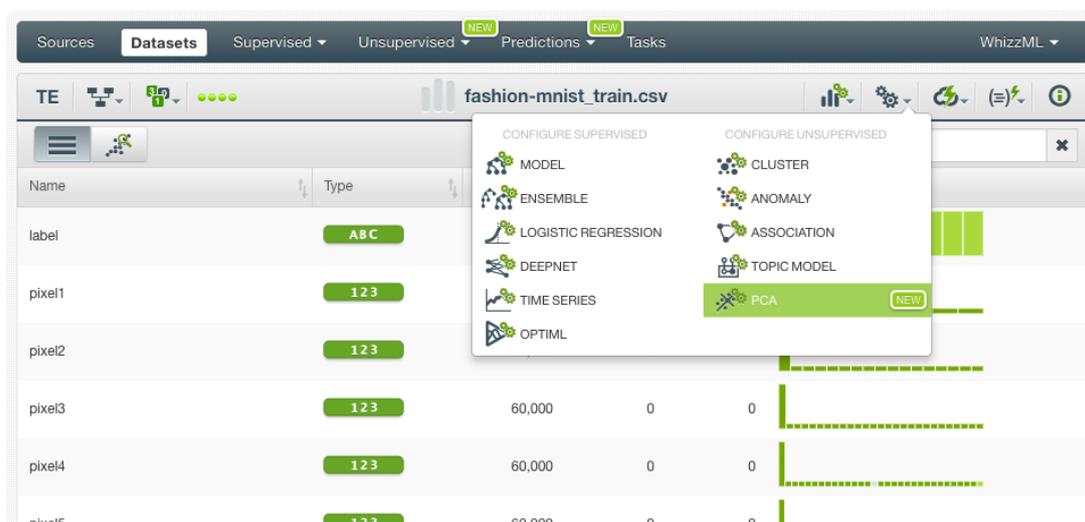


Figure 4.1: Configure PCA

4.1 Standardize

The **standardize** parameter allows you to automatically scale numeric **fields** to a 0-1 range. If standardize is enabled, the following formula is applied to each numeric field: $(field_value - mean) / stddev$. Standardizing implies assigning equal importance to all the fields when these are not measured on the same scale. If fields do not have the same scale and you create a PCA with non-standardized fields, it is often the case that each **principal component** is dominated by a single field. By default, standardization option is enabled.

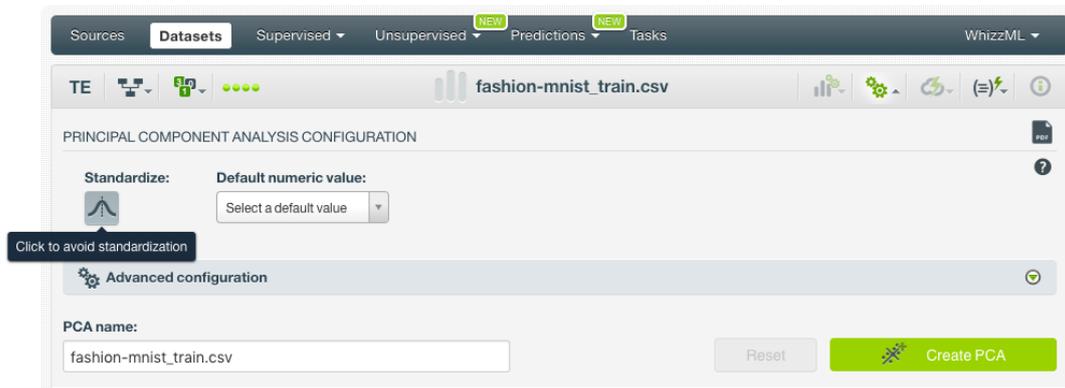


Figure 4.2: Standardize option for PCA

4.2 Default Numeric Value

PCA can include missing values as valid values for any type of fields as explained in [Section 2.2](#). 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 value. You can select to replace them with the field's **Mean**, **Median**, **Maximum**, **Minimum** or with **Zero**. (See [Figure 4.3](#).)

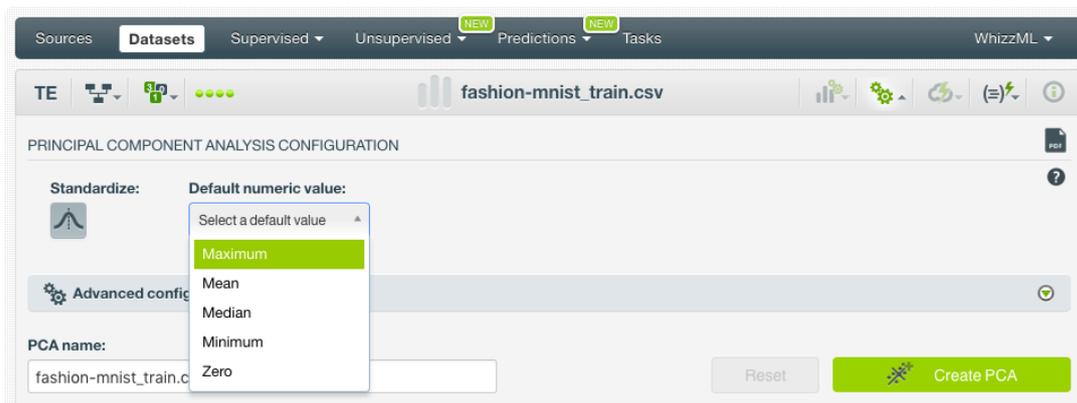


Figure 4.3: Select a default value to replace the missing numerics

Note: if your dataset does not contain missing values for your numeric fields, this parameter will not impact your PCA.

4.3 Sampling Options

Sometimes you do not need all the data contained in your training dataset to generate your PCA. If you have a very large dataset, sampling may be a good way of getting faster results. (See [Figure 4.4](#).) You can configure the sampling options explained in the following sections.

4.3.1 Rate

The rate is the proportion of instances to include in your sample. Set any value between 0% and 100%. It defaults to 100%.

4.3.2 Range

The 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.

4.3.3 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.

4.3.4 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.

4.3.5 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%.

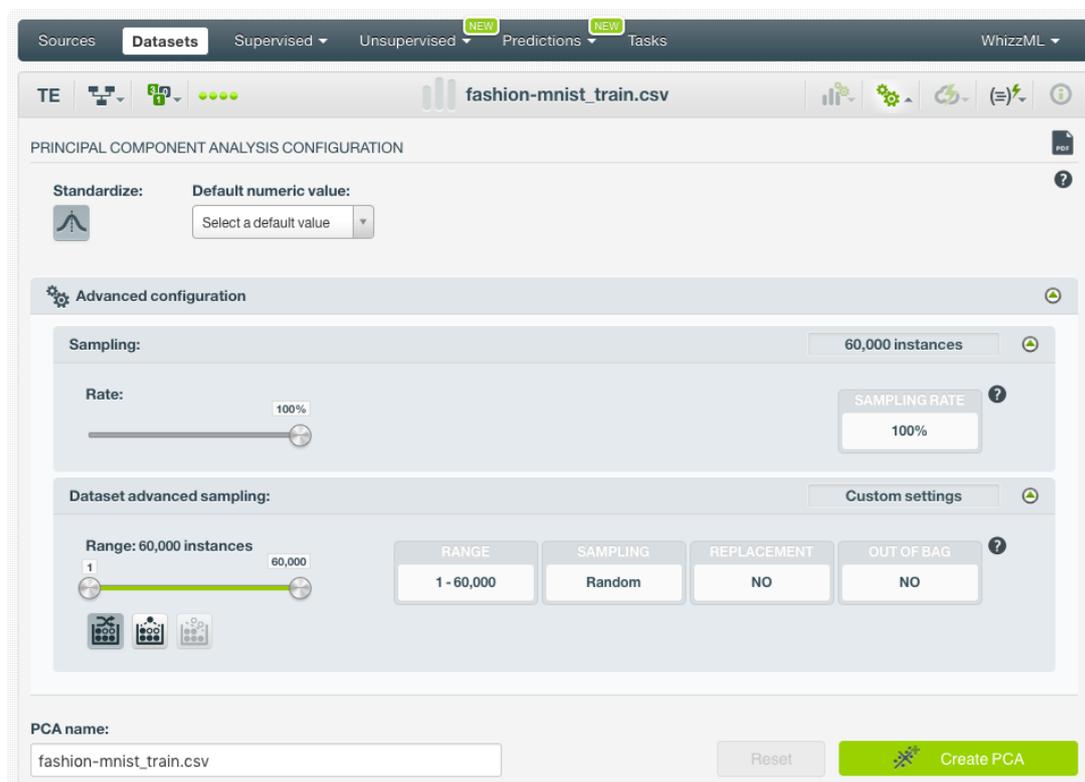


Figure 4.4: Sampling options for PCA

4.4 Creating PCA with Configured Options

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

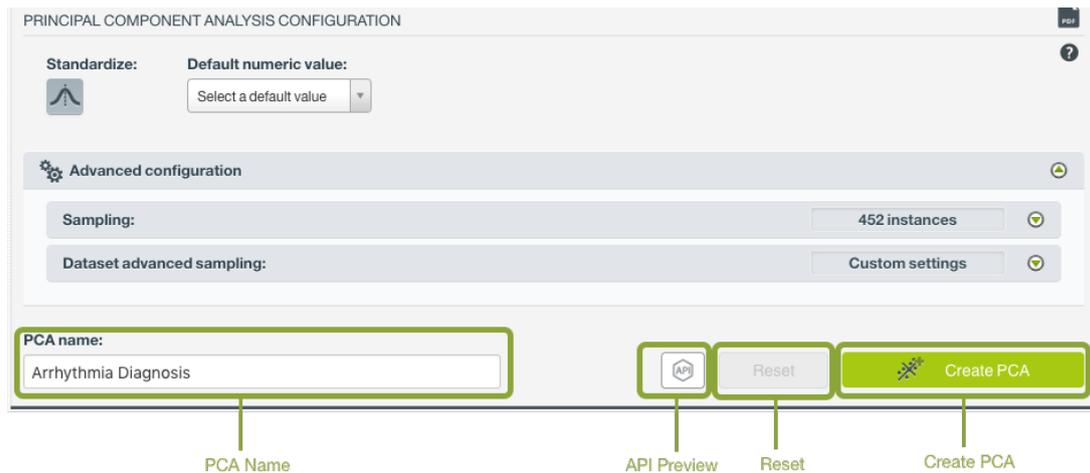


Figure 4.5: Create PCA after configuration

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 4.5)). This is to show how to create the PCA programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 4.6) below:

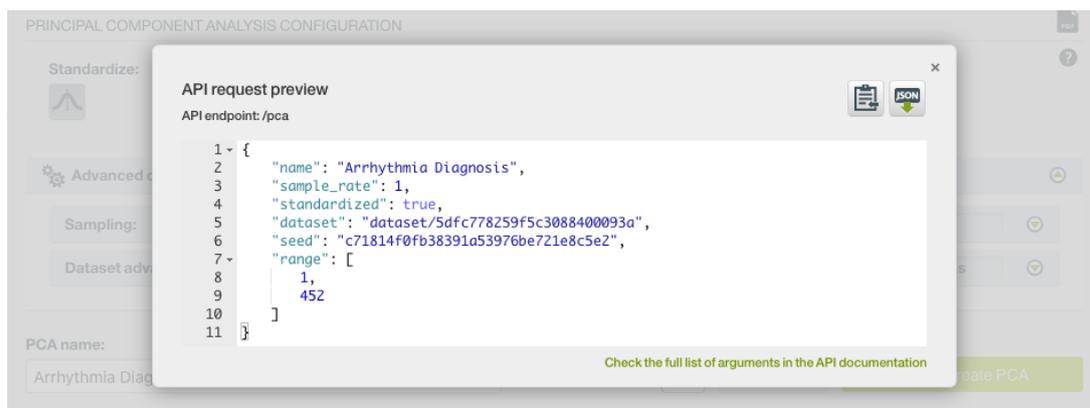


Figure 4.6: PCA API request preview

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 PCA, 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.

Visualizing PCA

BigML PCA view is composed of two main blocks: PRINCIPAL COMPONENTS (on the left) and SCREE PLOT (on the right). (See [Figure 5.1](#).)



Figure 5.1: PCA view

- PRINCIPAL COMPONENTS is a list containing the **principal components** calculated by the PCA sorted by **variance** in descending order. The principal components are named as PC N being N the ranking for each component according to their variance, i.e., PC1 is the first component with the largest variance, PC2 the second most important component and so on. The number of principal components returned should be the same as the number of input fields in the dataset if all the fields are numeric. If the dataset contains categorical, text and/or items fields then you will see a higher number of components since each class, term and item is considered a different field by the PCA (see [Chapter 2](#)). You can view up to 200 components in this list (see [Figure 5.2](#)).

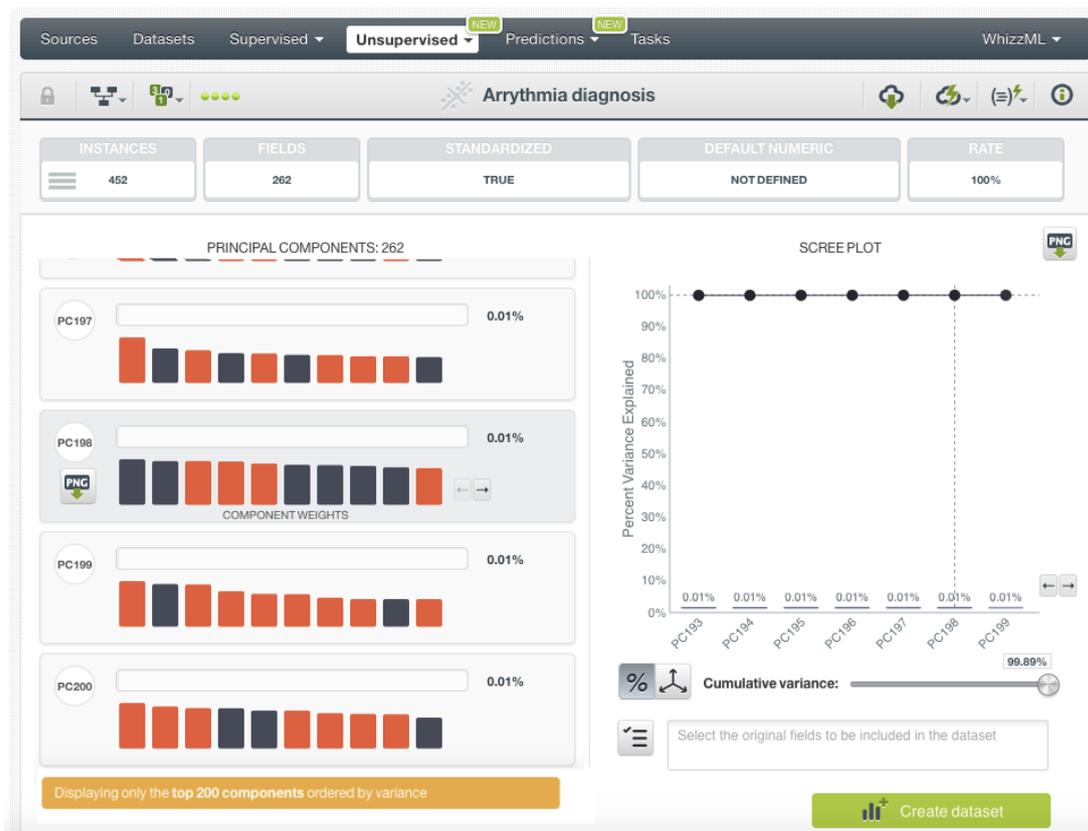


Figure 5.2: PCA total components

Each principal component includes the following information:

- **Variance:** it indicates how much variability of the data each is explained by each component (see Figure 5.3). It is always a number between 0% and 100%. The first principal component (PC1) has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is *orthogonal*¹ to the preceding components. All component variances sum up to 100%. If you want to reduce the dimensionality of a given dataset, you should choose the smallest subset of components that explains the largest variance possible to ensure you are not losing dataset information to build other Machine Learning models.

¹<https://en.wikipedia.org/wiki/Orthogonality>

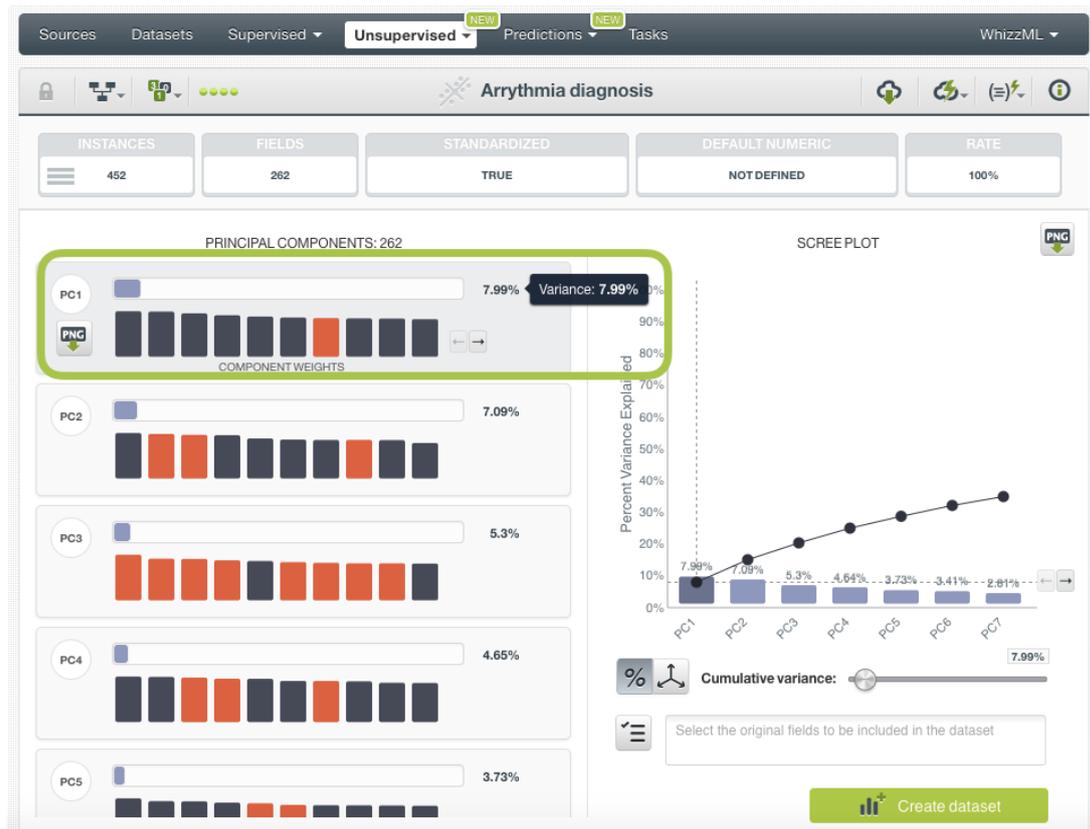


Figure 5.3: Principal components variance

- **Field weights:** below each component you can see a histogram where each bin represents an input field of the original dataset associated with a weight. The weight indicates the influence of each field in a given **principal component**. This weight can take positive (grey bins) or negative (red bins) values (see [Figure 5.4](#)). For categorical, text, or items fields, each class, term and item will have its own weight since they are considered different fields by the PCA.

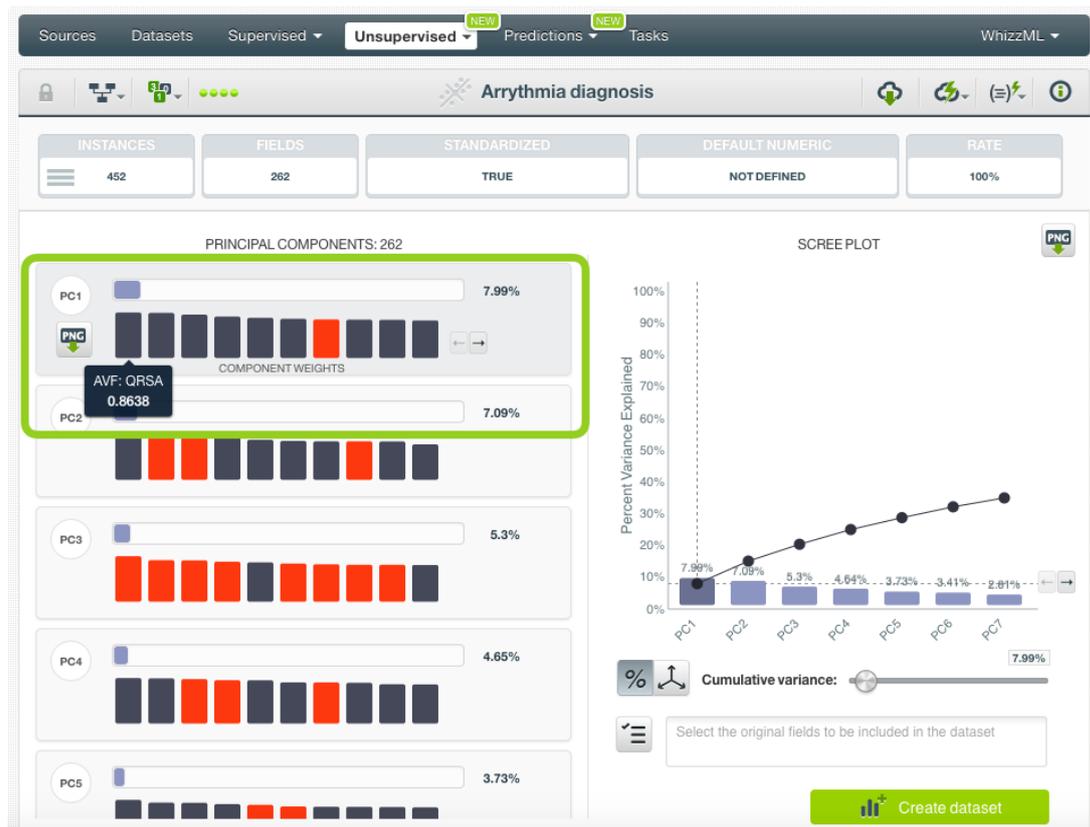


Figure 5.4: Field weights in components

- The SCREE PLOT is a graphical representation of each **principal component** (represented by each blue bar on the x-axis) and its **variance** (represented on the y-axis values). The black curve represents the **cumulative variance** of the components (see Figure 5.5). This visualization helps you to **select the subset of components** to create a new dataset either by setting a threshold for the cumulative variance or by limiting the total number of components (see slider below the chart in Figure 5.5). Unfortunately, there is not an objective way to decide the **optimal number of components** for a given cumulative variance. This depends on the data and the problem you are solving. For example, in the Figure 5.5, the first 51 components (which represent less than 20% of the original 261 dimensions) account for the 80% of the data variance which seems an acceptably large number to create a new dataset. Moreover, it looks like our dataset confirms the **Pareto Principle**² which states that 80% of the effects come from 20% of the causes, but these results may change depending on the dataset.

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

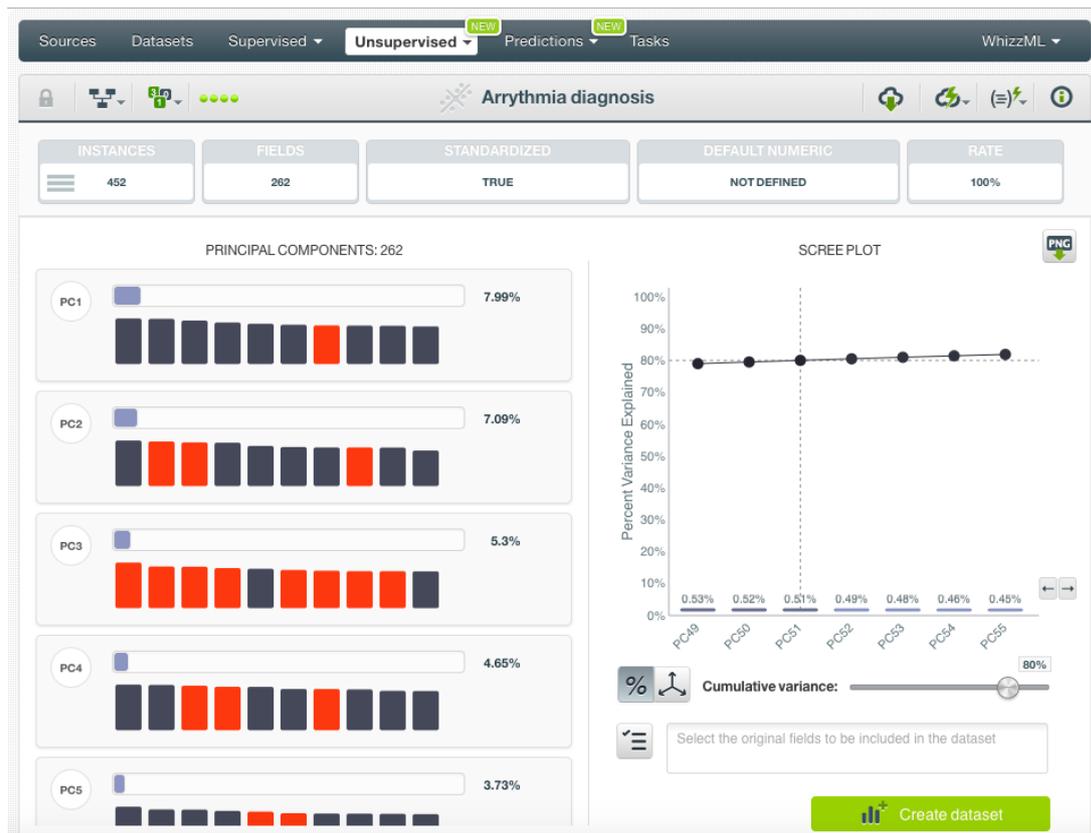


Figure 5.5: PCA scree plot

By clicking on the icon in the top right of the SCREE PLOT, you can download the chart view in PNG format. (See [Figure 5.6.](#))

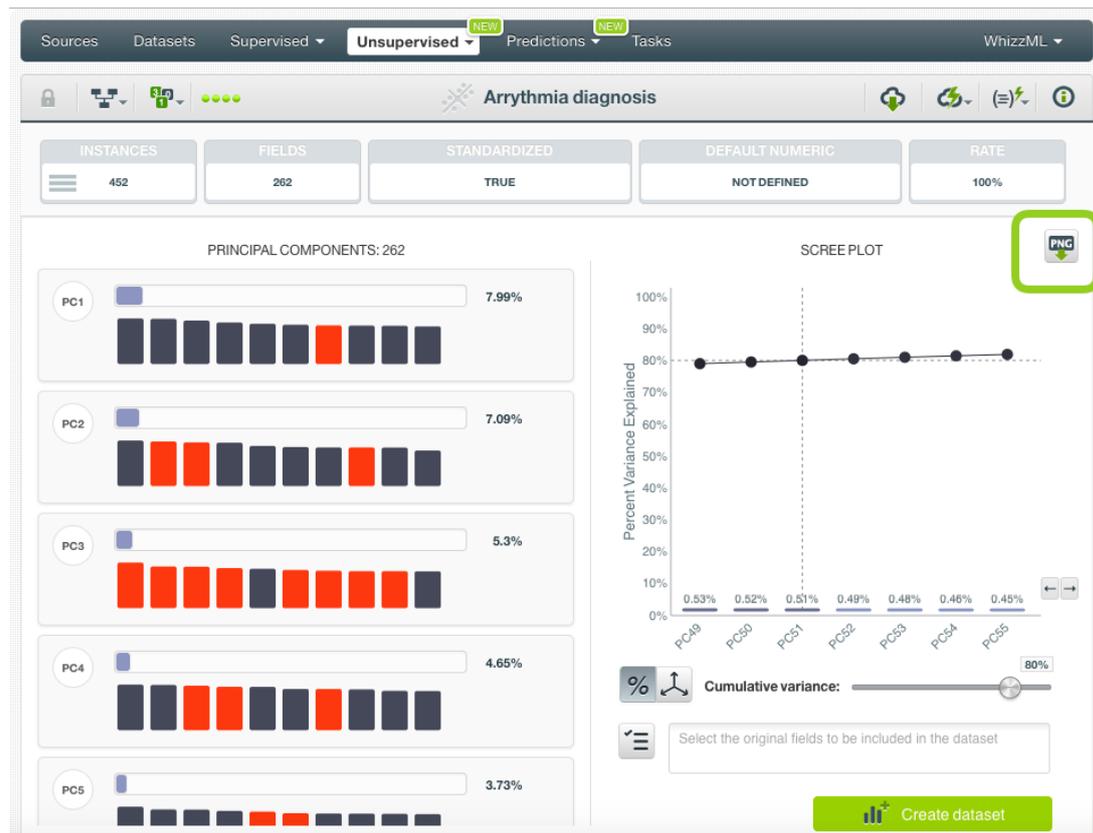


Figure 5.6: Export scree plot in PNG format

5.1 Create a Dataset From PCA

The final goal of PCA is usually reducing the dataset dimensionality to create other Machine Learning models. This is because lowering the number of fields helps to reduce noise and multicollinearity that may negatively impact the models' performance.

By clicking on the **Create dataset** button (see Figure 5.7), BigML will create a dataset in your Dashboard. This dataset will contain the **principal components** taking into account the threshold set using the slider shown in Figure 5.7. By clicking in the icons on the left of the slider, you can choose to either set a **threshold** for the **cumulative variance** or **limit the number of components**. You can also choose to include all or some of the original fields from your dataset by selecting them using the option shown in Figure 5.7. This option is very useful to include **ID fields** or when your final goal is to build supervised models and you need to include the **objective field** in the dataset.

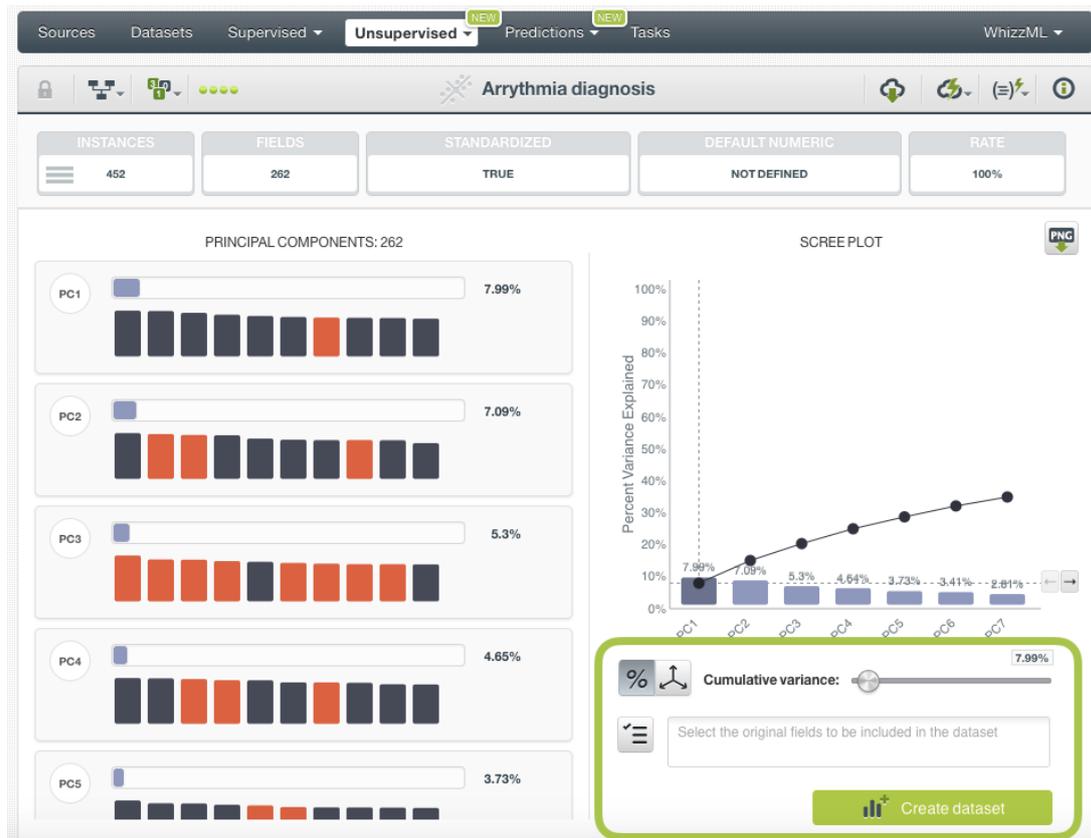


Figure 5.7: Create dataset from the PCA view

Behind the scenes, BigML creates a batch **projection** that you can access from the newly created dataset. If you want to exclude the original fields, tweak any other configuration parameters, or use another dataset, then you can easily do it from the **batch projection view** as explained in [Section 6.3](#).

PCA Predictions: Projections

6.1 Introduction

You can use your **PCA** to calculate the **principal components** for the same data used to create the model or new data that the model has not yet seen. Predictions for PCA are referred to as **projections** in BigML, since they aim to project the components given a set of values for your instances. PCA is mostly useful to transform all the dataset instances at once and not single instances one at a time. That's why BigML offers only batch projections in the Dashboard and not single projections.

The predictions tab in the main menu of the BigML Dashboard is where all of your saved predictions are listed (Figure 6.1). In the batch projections list view, you can see the icons for the **PCA** and **Dataset** used for each projection, the **Name** of the projection, the **Instances** used for the projection, and the **Age** (time since the projection was created). You can also search your projections by name clicking in the search menu option on the top right menu.

		Name	Instances	Age
		fashion-mnist_train.csv Training (80%) dataset with fashion-mnist_train.csv Training (80... variance threshold=0.78, use all fields	48K+	1d 22h
		fashion-mnist_tr...csv Test (20%) with fashion-mnist_train.csv Training (80%) variance threshold=1.0, use all fields	12K+	5d 6h
		fashion-mnist_tr... Training (80%) with fashion-mnist_train.csv Training (80%) variance threshold=1.0, use all fields	48K+	5d 8h
		fashion-mnist_train.csv Test (20%) dataset with fashion-mnist_train.csv Training (80%) use all fields	12K+	5d 22h
		fashion-mnist_train.csv Training (80%) dataset with fashion-mnist_train.csv Training (80... use all fields	48K+	5d 22h

Figure 6.1: Projections 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 projections will be empty. (See Figure 6.2.)

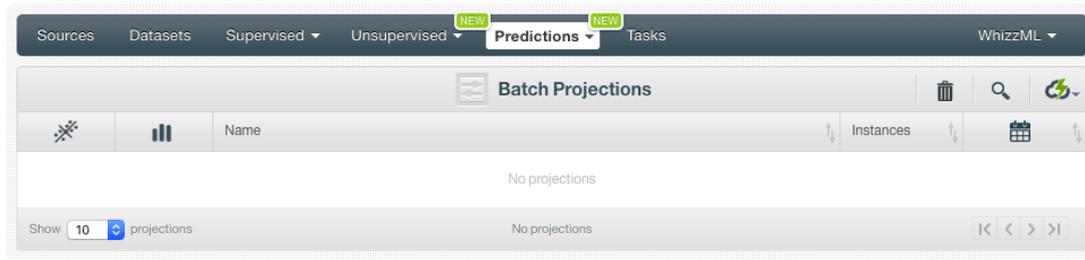


Figure 6.2: Empty Dashboard projections view

Projections are saved under the PCA PROJECTIONS option in the menu (see Figure 6.3.)

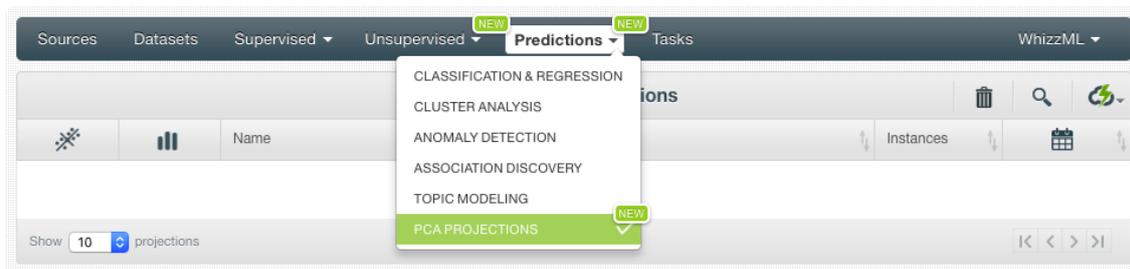


Figure 6.3: Menu options of the predictions

The icon shown in Figure 6.4 is the one used for batch projections.



Figure 6.4: Batch projection icon

6.2 Creating Projections

BigML usually provides single and batch predictions for most algorithms, i.e., predictions for a single instance or multiple instances simultaneously. However, in the case of PCA, it does not make sense to offer single projections since this model is not useful to predict only one instance at a time.

To create a batch **projection** in BigML, all you need is the **PCA** you want to use and a **dataset** containing the instances for which you want to obtain the **principal components**.

Follow these steps to create a batch projection:

1. Click on the BATCH PROJECTION option in the PCA **1-click menu**. (See Figure 6.5.)

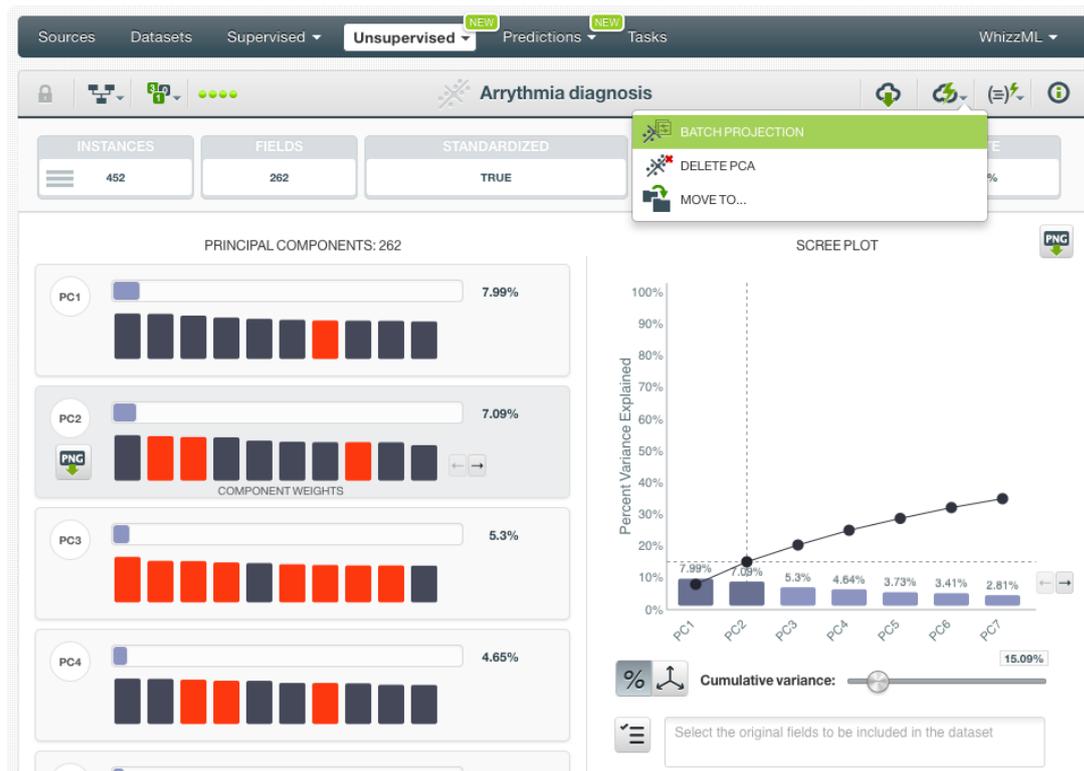


Figure 6.5: Batch projection from 1-click menu

Alternatively, click **CREATE BATCH PROJECTION** in the **pop up menu** from the PCA list view as shown in Figure 6.6.

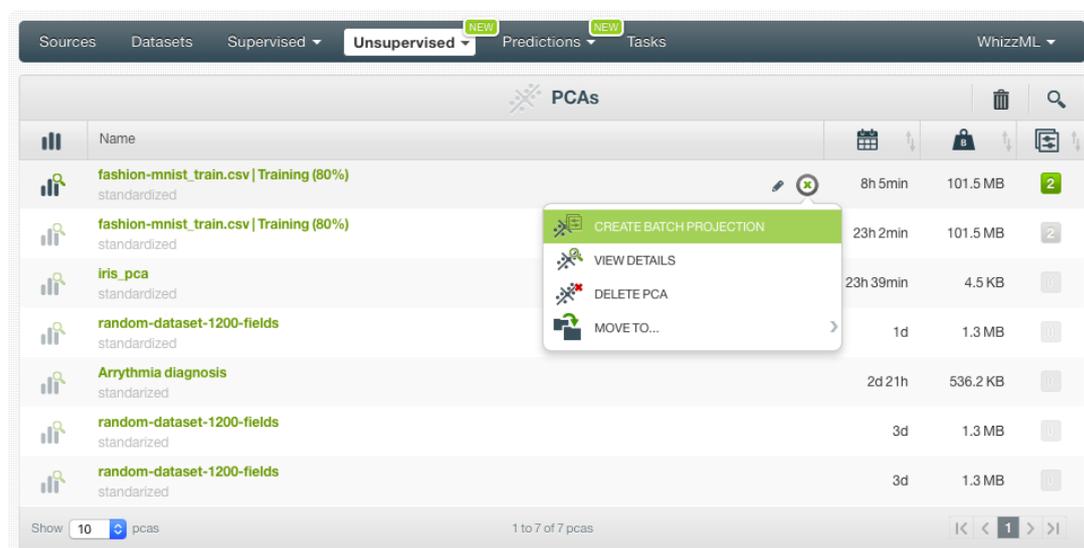


Figure 6.6: Batch projection from pop up menu

2. **Select the dataset** containing all the instances you want to predict. (See Figure 6.7.) The dataset should contain the input fields used by the PCA to calculate the components. BigML batch projections can handle missing data in your dataset as explained in Chapter 2. From this view you can also select another PCA from the PCA selector.

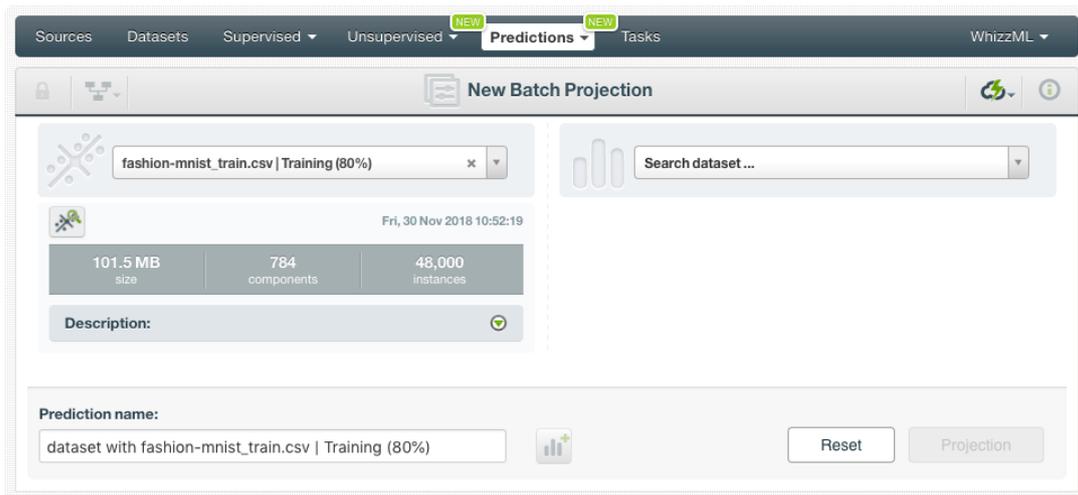


Figure 6.7: Select dataset for batch projections

- After you select the PCA and the dataset, the batch projection **configuration options** will appear along with a **preview of the prediction file**. (See [Figure 6.8](#).) The default format is a CSV file including all your dataset fields and adding N extra columns for the N components of the PCA. You can configure this file using the output settings explained in [Section 6.3](#).

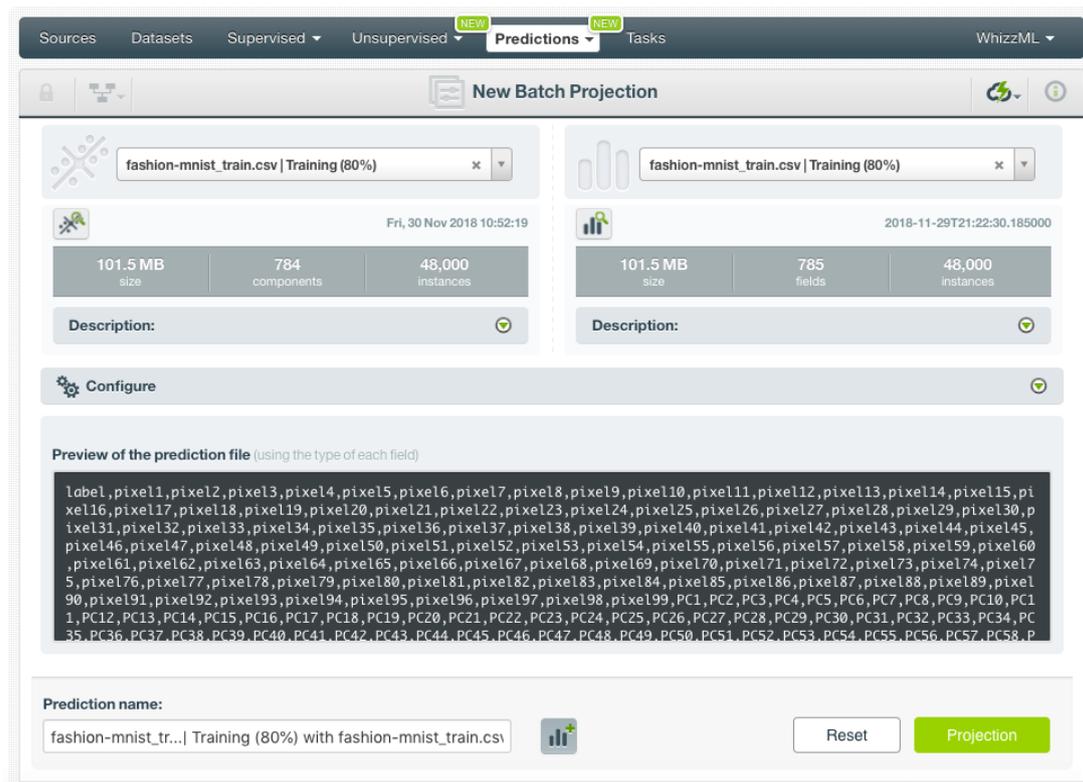


Figure 6.8: Configuration options displayed and output preview

- BigML generates an **output dataset** with your batch projections 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 in the icon shown in [Figure 6.9](#).

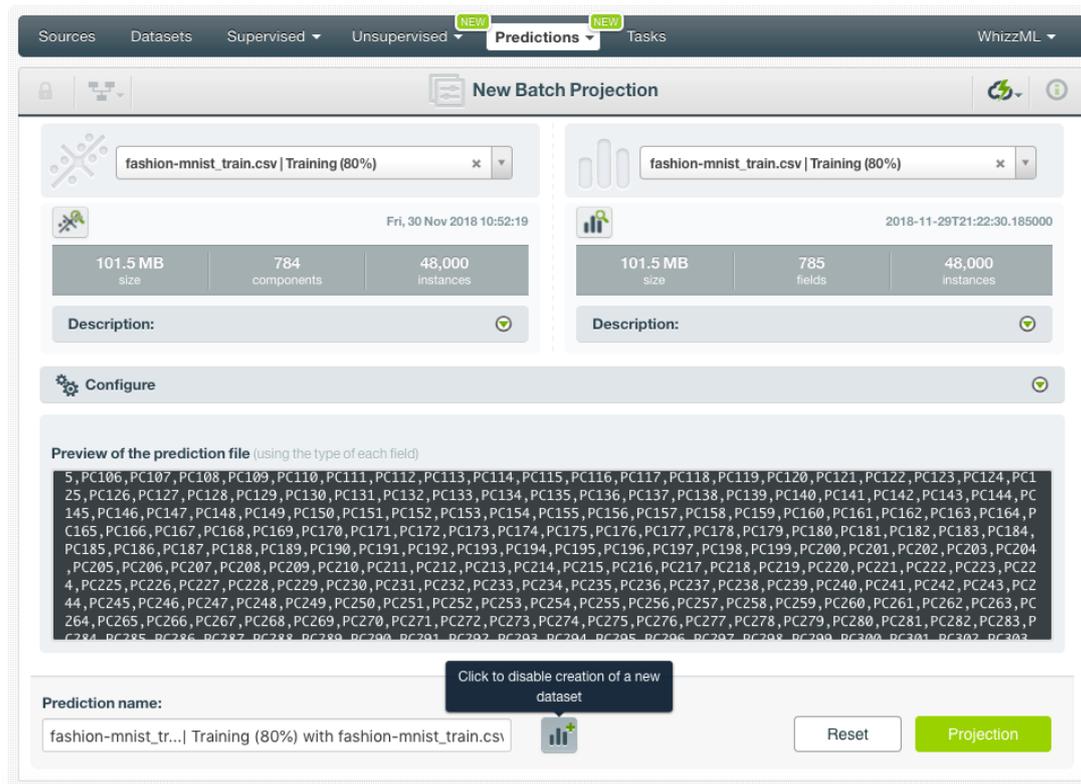


Figure 6.9: Create dataset from batch projection

5. Finally click on the green **Projection** button to generate your batch projection.
6. When the batch projection is created, you will be able to **download the batch projection** containing all your dataset instances along with the PCA components. If you did not disable the option to create a new dataset, you will also be able to access the **output dataset** from the batch projection view. (See Figure 6.10.)

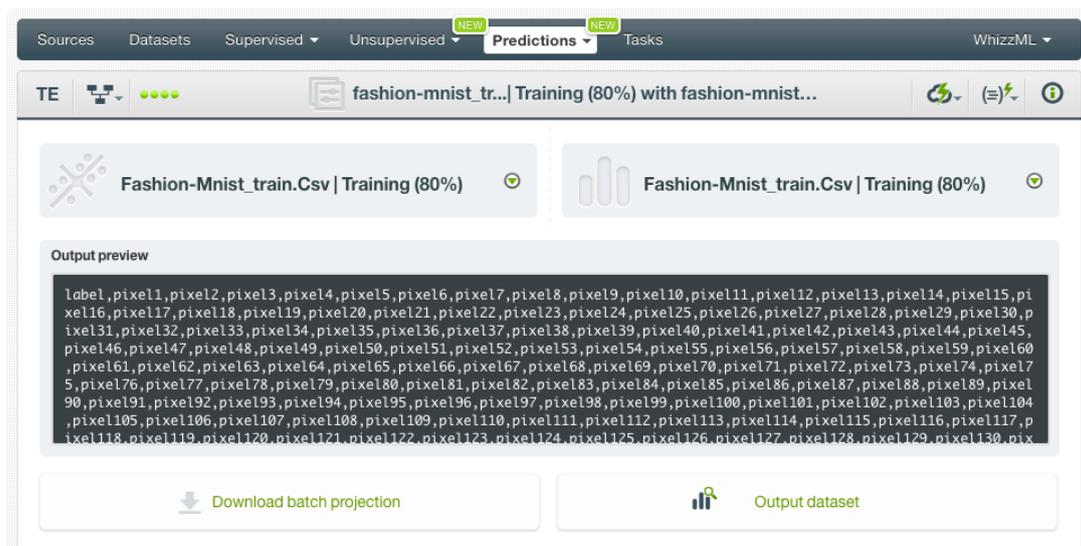


Figure 6.10: Download batch projection and access output dataset

6.3 Configuring Projections

BigML provides several options to configure your projections, such as **limiting the number of components** (see [Subsection 6.3.1](#)), defining the **field mapping** performed by BigML (see [Subsection 6.3.3](#)), and the **output file settings** (see [Subsection 6.3.4](#)).

6.3.1 Limit the Number of Components

By default, BigML returns all the **principal components** in your batch projections. However, you can limit the total number of components to be returned by using either these two options:

- **Cumulative variance:** you can limit the number of components by setting the maximum data **variance** that you want them to account for. The right amount of variance depends on the dataset; however, a cumulative variance between 80% and 90% is usually enough to avoid losing too much information from the original data.
- **Maximum number of components:** you can select the exact number of components to be included in the batch projection. This option is useful if you have an ideal number of dimensions that you want to get in the new reduced dataset. Again, the optimal number of components will depend on each specific case.

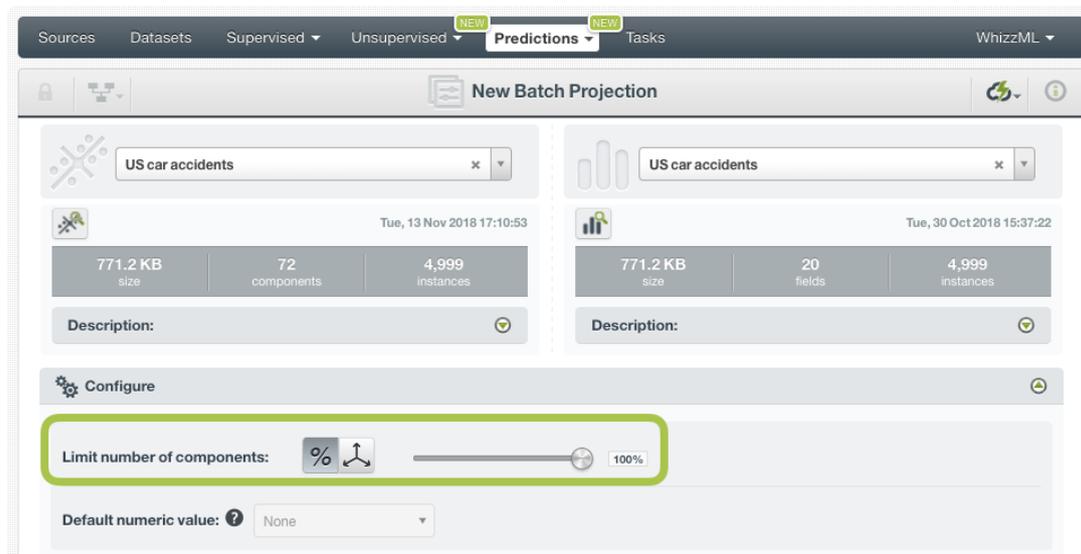


Figure 6.11: Set a limit for the number of components to be returned

6.3.2 Default Numeric Value

For batch projections, the selector to configure the **default numeric value** to replace the missing values in your dataset will always be **disabled**. This is because this value needs to be the same as the one used to create the PCA (see [Section 4.2](#)). If you did not use any default numeric value to create the PCA, the selector in the batch projection will show a “None” value as you can see in [Figure 6.12](#).

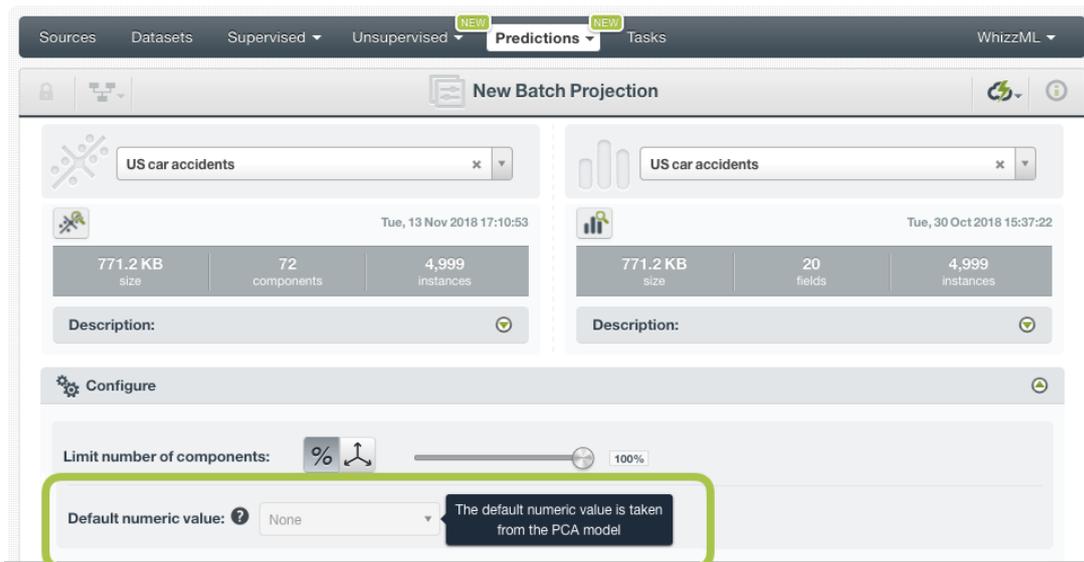


Figure 6.12: See the default numeric value to be used in batch projections

6.3.3 Field Mapping

You can specify which fields in the PCA match with which fields in the dataset containing the instances you want to project. BigML automatically matches fields by **name**, but you can set an automatic match by **field ID** by clicking in the green switcher shown in Figure 6.13. You can also **manually** search for fields or remove them if you do not want to consider them during the scoring.

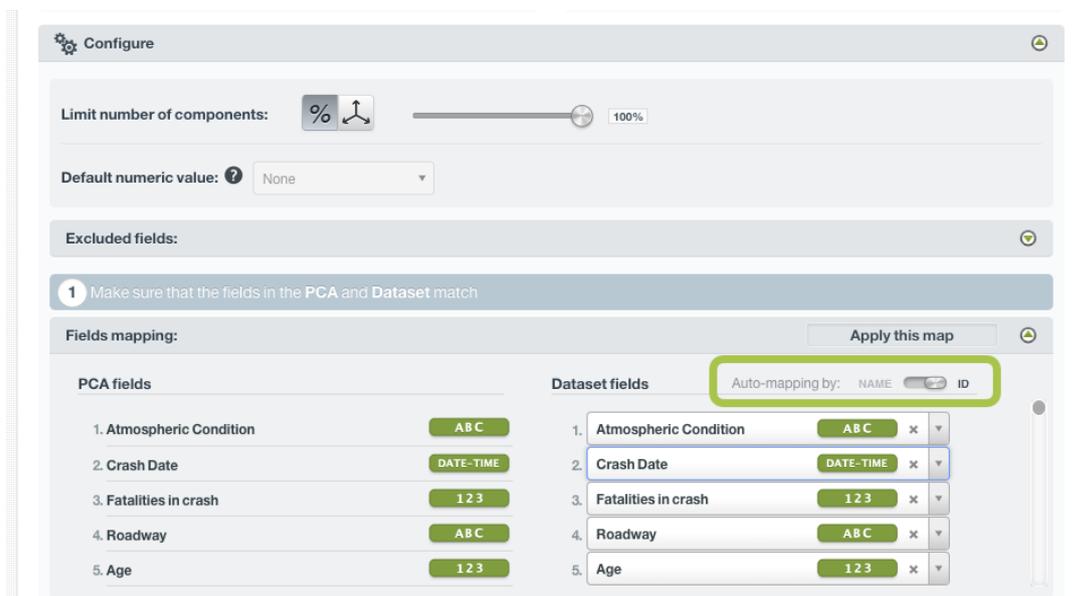


Figure 6.13: Field mapping for batch projections

Note: the field mapping from the BigML Dashboard has a limit of 200 fields. For batch projections with higher number of fields you can use [BigML API](https://bigml.com/api/batchprojections)¹.

¹<https://bigml.com/api/batchprojections>

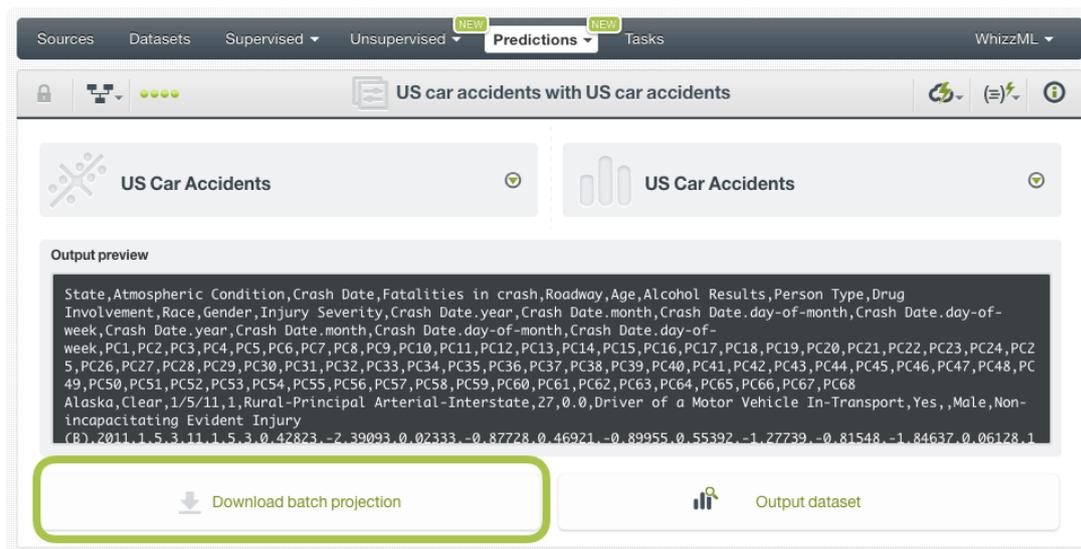


Figure 6.15: Download batch projection output file

By default, it will be a CSV file containing all the dataset fields along with all the **principal components** from the PCA as appended columns. You can customize the output file settings as explained in [Subsection 6.3.4](#).

See an output CSV file example in [Figure 6.16](#) where the last columns contain the components for each instance.

```
Pregnancies,Glucose,Insulin,BMI,PC1,PC2,PC3,PC4
0,137,168,43.1,2.288,3.44,9.76,23.67
11,143,146,36.6,0.254,53.56,2.12,5.76
4,103,192,24.0,0.966,25.7,1.89,5.87
7,106,0,22.7,0.235,2.57,3.65,7.23
2,109,0,42.7,0.845,4.98,3.98,12.45
1,122,220,49.7,0.634,1.56,2.78
8,155,495,34.0,28.92,5.98,1.78
4,76,0,34.0,0.391,21.3,3.67,7.34
1,118,94,33.3,0.261,23,1.45,9.45
```

Figure 6.16: An example of a batch prediction CSV file

Output Dataset

By default, BigML automatically creates a dataset out of your batch projection score. You can disable this option by configuring your batch projection as explained in [Section 6.3](#). You can access your output dataset from the batch projection view as shown in [Figure 6.17](#).

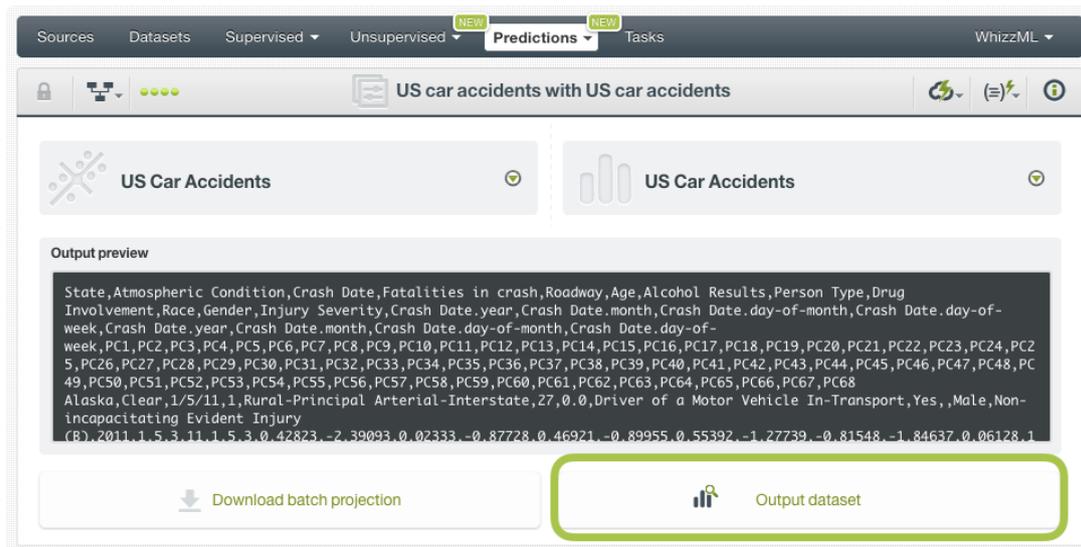


Figure 6.17: Access batch projection output dataset

In the output dataset you can find N additional **fields** (named by default “PC N ” being “ N ” an integer starting at 1) containing each of the components from the PCA (see Figure 6.18).

Name	Type	Count	Missing	Errors	Histogram
PC1	123	4,999	0	0	[Histogram]
PC2	123	4,999	0	0	[Histogram]
PC3	123	4,999	0	0	[Histogram]
PC4	123	4,999	0	0	[Histogram]
PC5	123	4,999	0	0	[Histogram]
PC6	123	4,999	0	0	[Histogram]
PC7	123	4,999	0	0	[Histogram]
PC8	123	4,999	0	0	[Histogram]
PC9	123	4,999	0	0	[Histogram]
PC10	123	4,999	0	0	[Histogram]

Figure 6.18: Batch projection output dataset

Batch Projection 1-Click Menu

From the batch projection view you can perform the following actions shown in Figure 6.19

- **BATCH PROJECTION AGAIN:** this option will redirect you to the batch projection creation view where you will have the same PCA and dataset already selected. It is a quick way if you want to

create the batch projection again using a different configuration.

- **BATCH PROJECTION WITH ANOTHER DATASET:** this is an easy way to create a batch projection using the same PCA and a different dataset.
- **BATCH PROJECTION USING ANOTHER PCA:** this is an easy way to create a batch projection using the same dataset and a different PCA.
- **NEW BATCH PROJECTION:** this will redirect you to the batch projection creation view where you will be able to select a dataset and a PCA to create your batch projection.

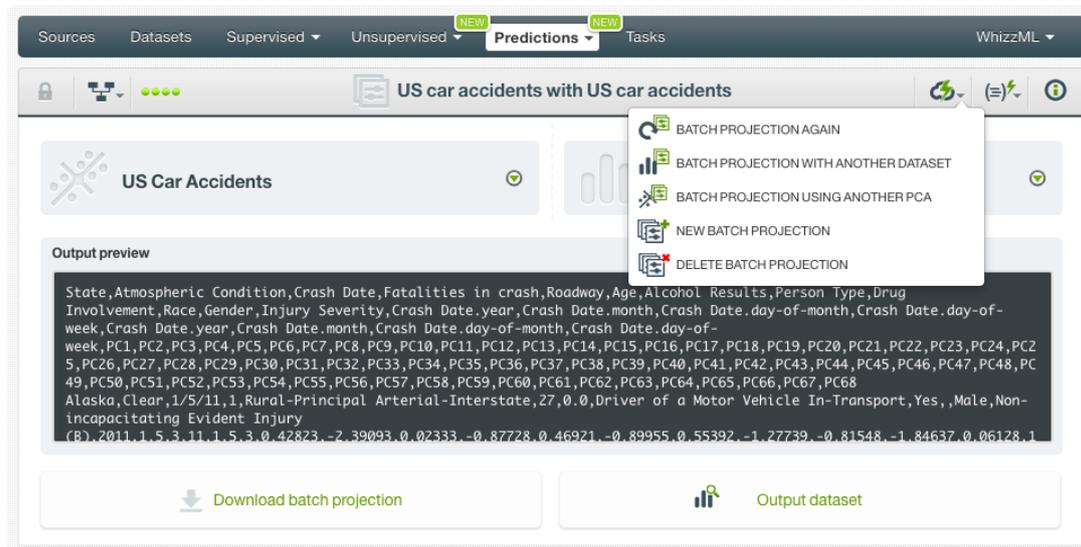


Figure 6.19: Batch projection 1-click menu

6.5 Consuming Projections

You can fully use batch projections via the BigML API and bindings. The following subsections explain both tools.

6.5.1 Using Projections Via the BigML API

You can perform all the projections actions explained in this document such as creating, configuring, retrieving, listing, updating, and deleting projections via the BigML API.

The example below shows how to create a batch projection with after the `BIGML_AUTH` environment variable that contains your authentication credentials is properly set:

```
curl "https://bigml.io/batchprojection?${BIGML_AUTH}" \
  -X POST \
  -H 'content-type: application/json' \
  -d '{"pca": "pca/5423625af0a5ea3eea000028",
      "dataset": "dataset/54222a14f0a5eaaab000000c"}'
```

For more information on using projections through the BigML API, please refer to [projections REST API documentation](#)².

6.5.2 Using Projections Via the BigML Bindings

You can also create, configure, retrieve, list, update, and delete batch projections via **BigML bindings** which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML

²<https://bigml.com/api/projections>

offers bindings in multiple languages including Python, Node.js Java, Swift and Objective-C. You can find below an example to create projections with the Python bindings.

```
from bigml.api import BigML
api = BigML()
prediction = api.create_batch_projection("pca/50650bdf3c19201b64000020",
                                       "dataset/23150b789fdh0175940283")
```

For more information on BigML bindings, please refer to the [bindings page](#)³.

6.6 Descriptive Information

Each projection has an associated **name**, **description**, **category** and **tags**. Those options are editable through the MORE INFO menu on the top right of the projection view. (See [Figure 6.20](#).)

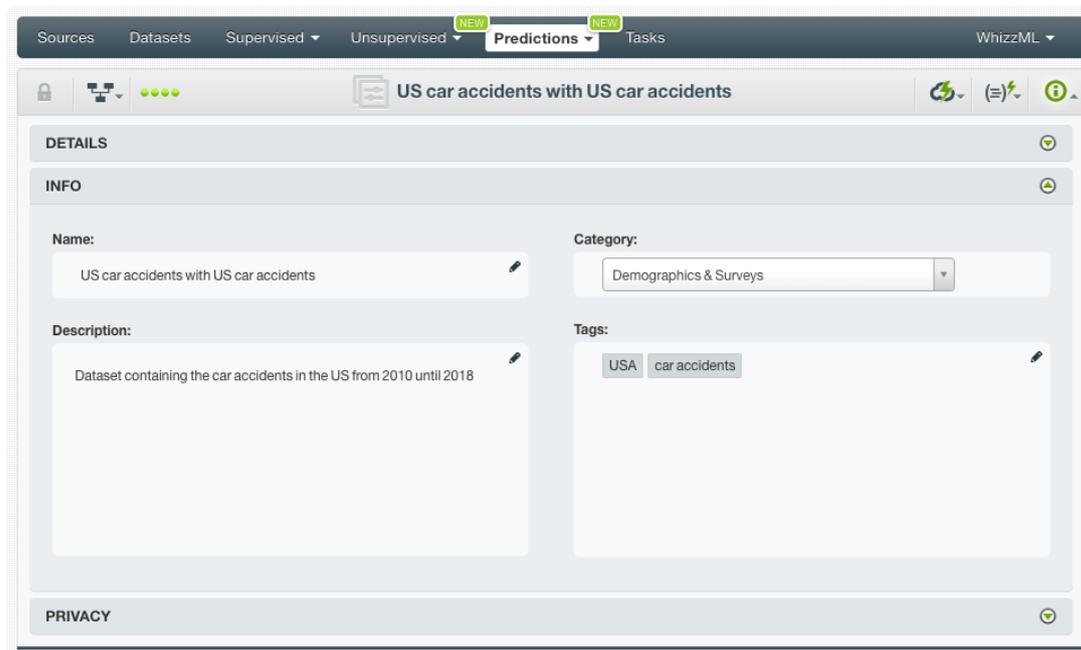


Figure 6.20: Edit projection metadata from More info panel

6.6.1 Projection Name

If you do not specify a **name** for your scores, BigML assigns a default name by combining your dataset name and the PCA name: “<PCA name> with <dataset name>”.

Projection names are displayed on the list view and also on the top bar of a projection view. Projection names are indexed to be used in searches. You can rename your scores at any time from the MORE INFO menu option.

The name of a projection cannot be longer than **256** characters. More than one projection can have the same name even within the same project, but they will always have different identifiers.

6.6.2 Description

Each projection score also has a **description** that it is very useful for documenting your Machine Learning projects. Single and batch scores take the description from the PCA used to create them.

³<https://bigml.com/tools/bindings>

Descriptions can be written using plain text and also [markdown](#)⁴. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See [Figure 6.21](#).)

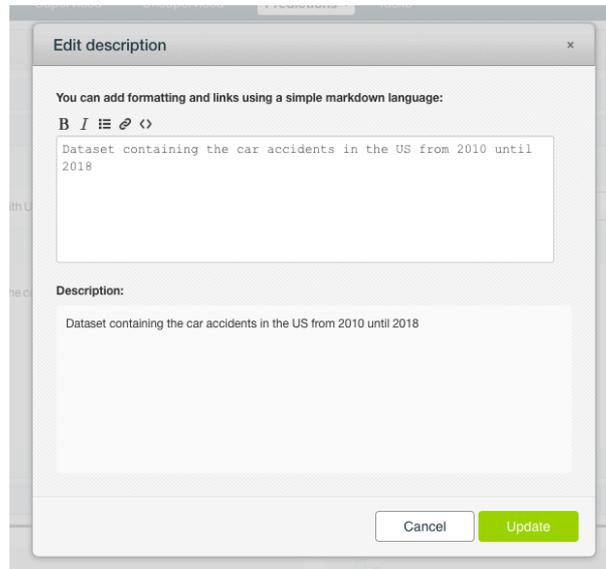


Figure 6.21: Markdown editor for projection descriptions

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

6.6.3 Category

Each projection has associated a **category**. Categories are useful to classify projections according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or multiple customers. By default, batch projections take the category from the PCA used to create them.

A projection category must be one of the categories listed on [Table 6.1](#).

⁴<https://en.wikipedia.org/wiki/Markdown>

Table 6.1: Categories used to classify projections 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

6.6.4 Tags

A projection can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide projection with some extra information. Projections inherit the tags from the PCA used to create it.

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

6.7 Projection Privacy

The link displayed in the **privacy panel** is the private URL of your projection, so only a user logged into your account is able to see it. Batch projections cannot be shared from the BigML Dashboard by sharing a link as you can do it with other resources.

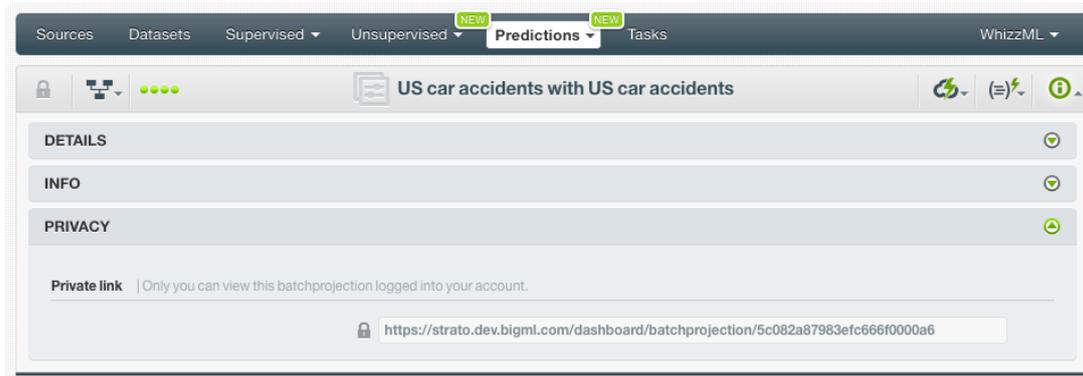


Figure 6.22: Private link of a projection

6.8 Moving Projections

When you create a projection it will be assigned to the same project where the original PCA is located. You cannot move projections between projects as you do with other resources.

6.9 Stopping Projections Creation

Batch projections are asynchronous resources so you can stop their creation before the task is finished. You can use the DELETE BATCH PROJECTION option from the **1-click menu**. (See Figure 6.23.)

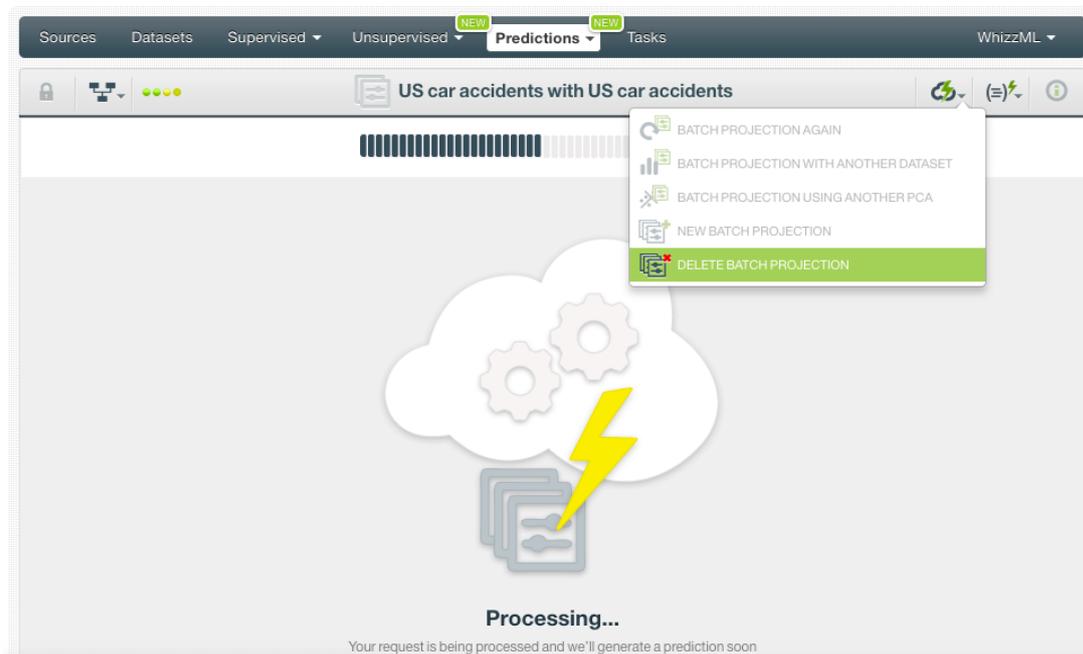


Figure 6.23: Stop batch projection from the 1-click menu

Alternatively, you can use the DELETE BATCH PROJECTION from the **pop up menu** on the projection view. (See Figure 6.24.)

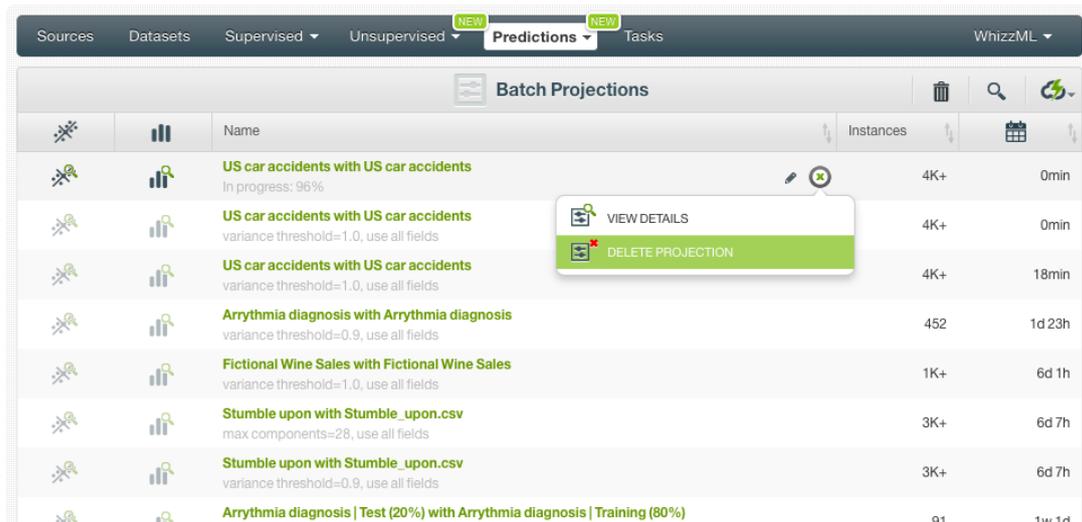


Figure 6.24: Stop batch projection from the pop up menu

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

6.10 Deleting Projections

You can delete your batch projections by clicking on the DELETE BATCH PROJECTION in the **1-click menu** (see Figure 6.25).

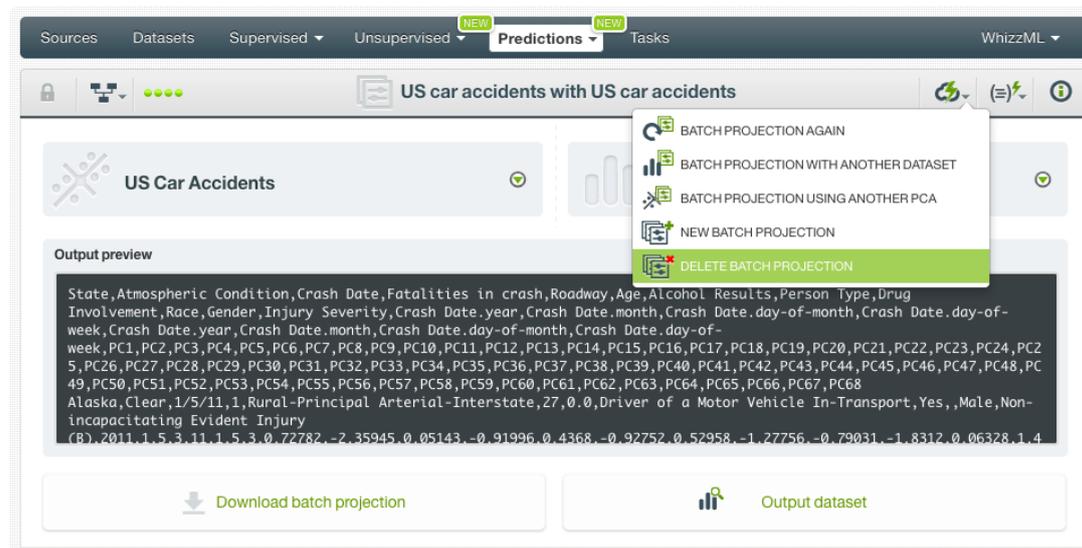


Figure 6.25: Delete batch projection from the 1-click menu

Alternatively, you can click the DELETE BATCH PROJECTION in the **pop up menu** from the projection list view (see Figure 6.26.)

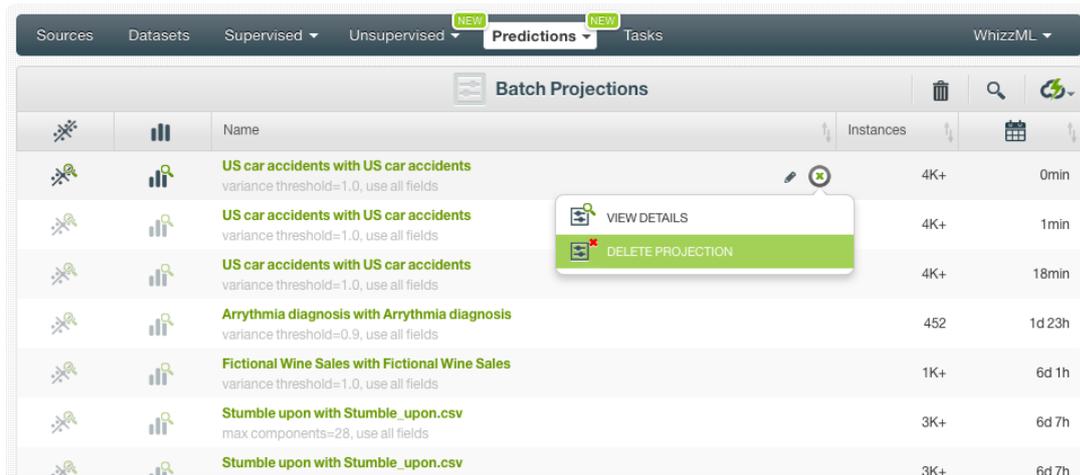


Figure 6.26: Delete batch projection from pop up menu

A modal window will be displayed asking you for confirmation. Once a projection is deleted, it is permanently deleted and there is no way you (or even the IT folks at BigML) can retrieve it. (Figure 6.27.)

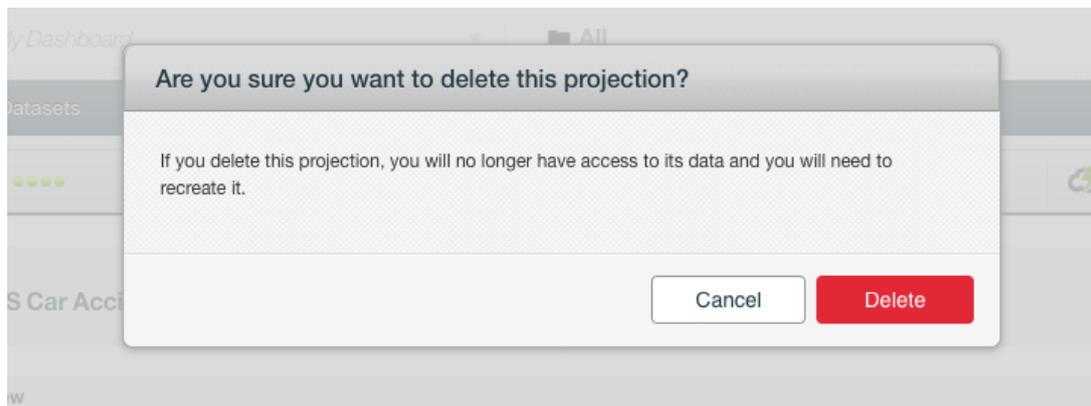


Figure 6.27: Delete projection confirmation

Consuming PCA

Similarly to other models in BigML, you can **download** PCA and use them locally to make **projections**. You can also create and consume your PCA programmatically via the **BigML API and bindings**. The following subsections explain these three options.

7.1 Downloading PCA

You can download your PCA in Python. Just click on the DOWNLOAD ACTIONABLE PCA menu option and select your preferred language. (See [Figure 7.1.](#))

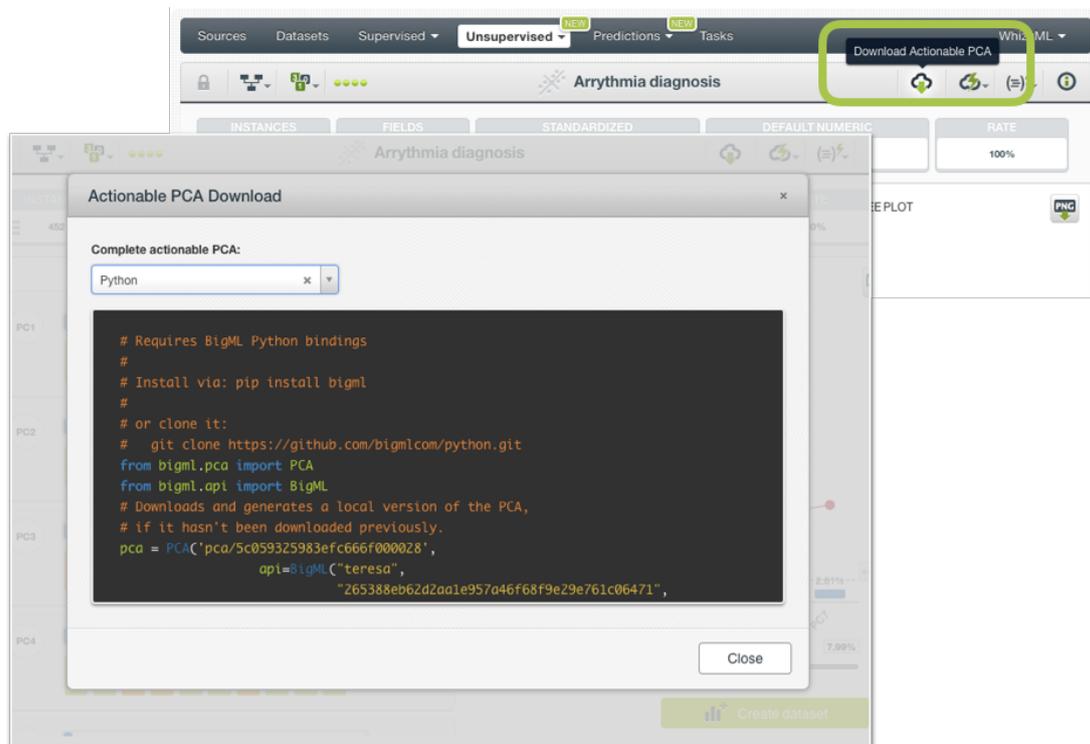


Figure 7.1: Downloading PCA

You can predict the **principal components** for your new data locally, free of latency, and at no cost by downloading your PCA. It works the same way as **local predictions** for models and ensembles.

7.2 Using PCA Via the BigML API

PCAs have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them to project new data.

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

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

For more information on using PCAs through the BigML API, please refer to [PCA REST API documentation](#)¹.

7.3 Using PCA Via the BigML Bindings

You can also create and use PCA 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. You can find below an example to create a PCA with the Python bindings.

```
from bigml.api import BigML  
api = BigML()  
pca = api.create_pca('dataset/57506c472275c1666b004b10')
```

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

¹<https://bigml.com/api/pca>

²<https://bigml.com/tools/bindings>

PCA Limits

BigML PCAs have a few limitations regarding the type of input data they can support. BigML also impose some limits on the components to visualize a PCA. You can find all limits listed below:

- **Categorical fields:** a maximum number of 1,000 distinct classes per field is allowed.
- **Text fields:** a maximum of 1,000 terms are considered per dataset.
- **Item fields:** a maximum of 10,000 items are considered per dataset.
- **Principal components:** a maximum of 200 components are displayed in the PCA view.

PCA Descriptive Information

PCAs have an associated **name**, **description**, **category**, and **tags**. The following subsections briefly describe each concept. See in [Figure 9.1](#) the options under MORE INFO menu to edit PCAs.

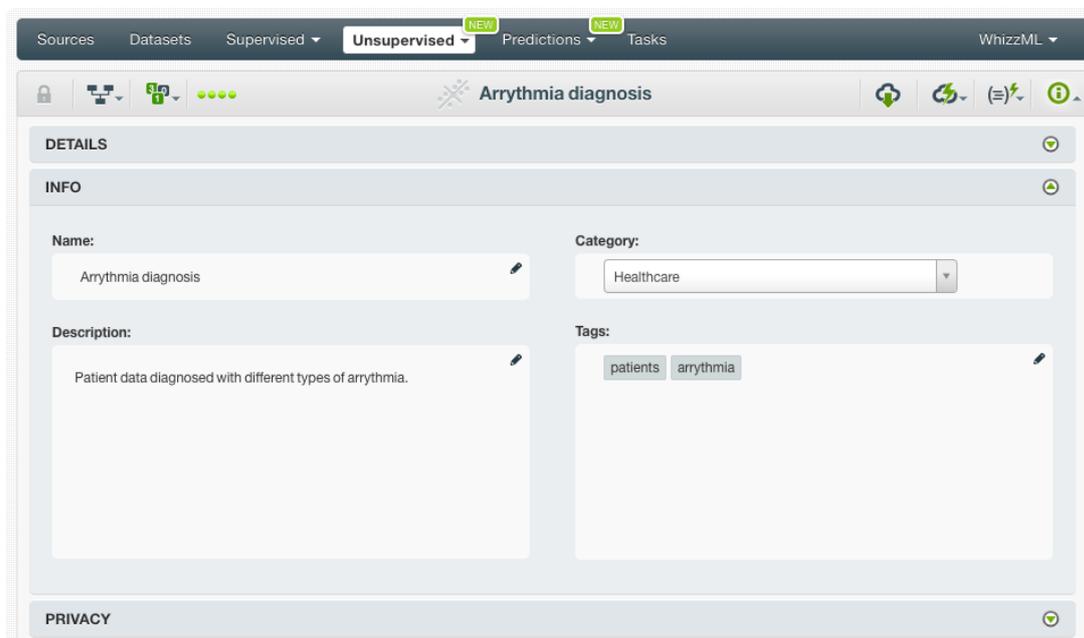


Figure 9.1: Editing PCAs

9.1 PCA Name

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

9.2 Description

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

Descriptions can be written using plain text and also [markdown](#)¹. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See [Figure 9.2](#).)

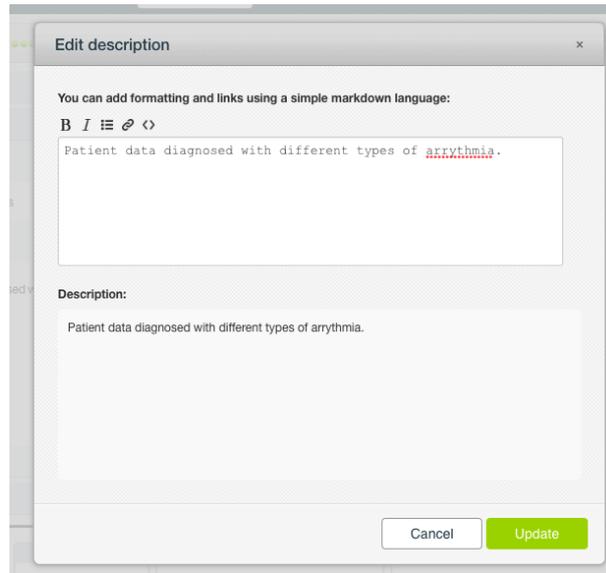


Figure 9.2: Markdown editor for PCA descriptions

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

9.3 Category

A **category**, taken from the dataset used to create it, is associated with each PCA. Categories are useful to classify PCAs 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 PCA category must be one of the **24** categories listed in [Table 9.1](#).

¹<https://en.wikipedia.org/wiki/Markdown>

Table 9.1: Categories used to classify PCAs 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

9.4 Tags

A PCA can also have a number of **tags** associated with it that can help to retrieve it via the BigML API or to provide PCA with some extra information. PCAs inherit the tags from the dataset used to create them. Each tag is limited to a maximum of 128 characters. Each PCA can have up to **32** different tags.

9.5 Counters

For each PCA, BigML also stores a number of counters to track the number of other resources that have been created using the PCA as a starting point. Display the counters by mousing over the menu option at the top of the PCA view. Click on VIEW # BATCH PROJECTIONS FROM THIS PCA menu option see all batch projections. (See [Figure 9.3.](#))



Figure 9.3: Counters for PCA

PCA Privacy

Privacy options for PCA can be defined in the **More Info** menu option. (See [Figure 10.1](#).) There are two levels of privacy for BigML PCA:

- **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 PCA. The first one is a sharing link that you can copy and send to others so they can visualize and interact with your PCA. The second one is a link to embed your PCA directly on your web page.



Figure 10.1: PCA privacy

Moving PCA

When you create a PCA, it will be assigned to the same **project** where the original dataset is located. PCAs can only be assigned to a single project. However, you can move PCAs **between projects in your Dashboard** or to other **Organization projects**. The menu option to do this can be found in two places:

1. Click MOVE TO... within the **1-click menu** from the PCA view. (See [Figure 11.1.](#))

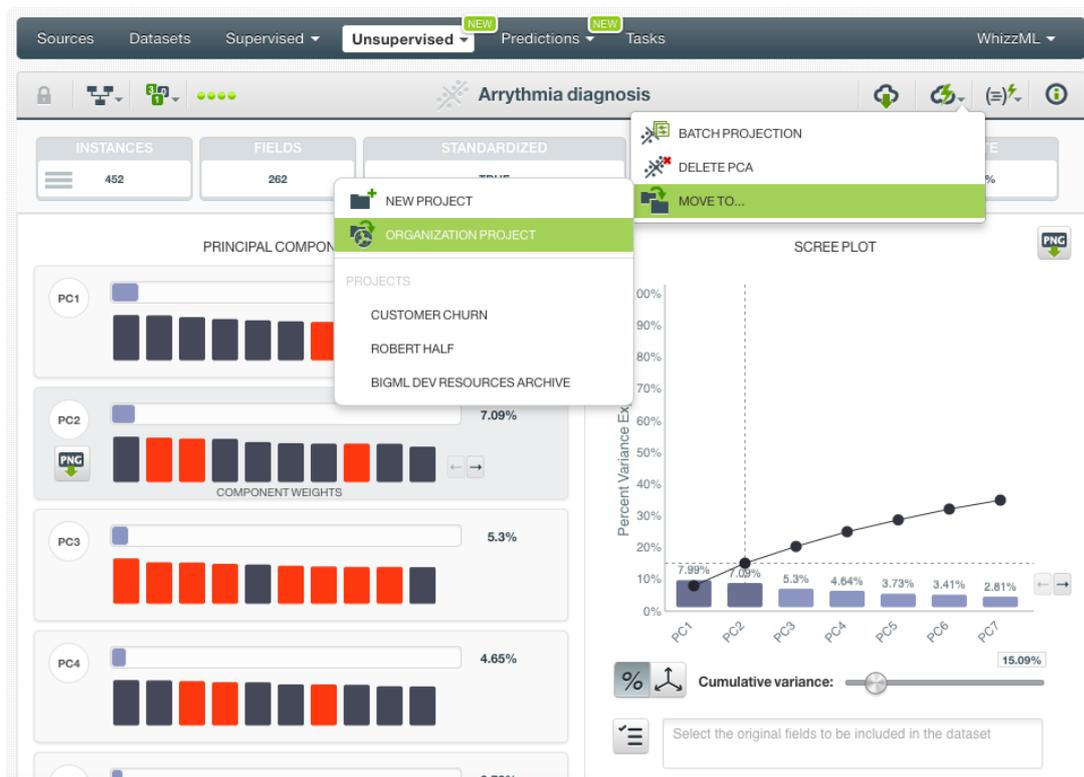


Figure 11.1: Change project from 1-click menu

2. Click MOVE TO... within the **pop up menu** from the PCA list view. (See [Figure 11.2](#))

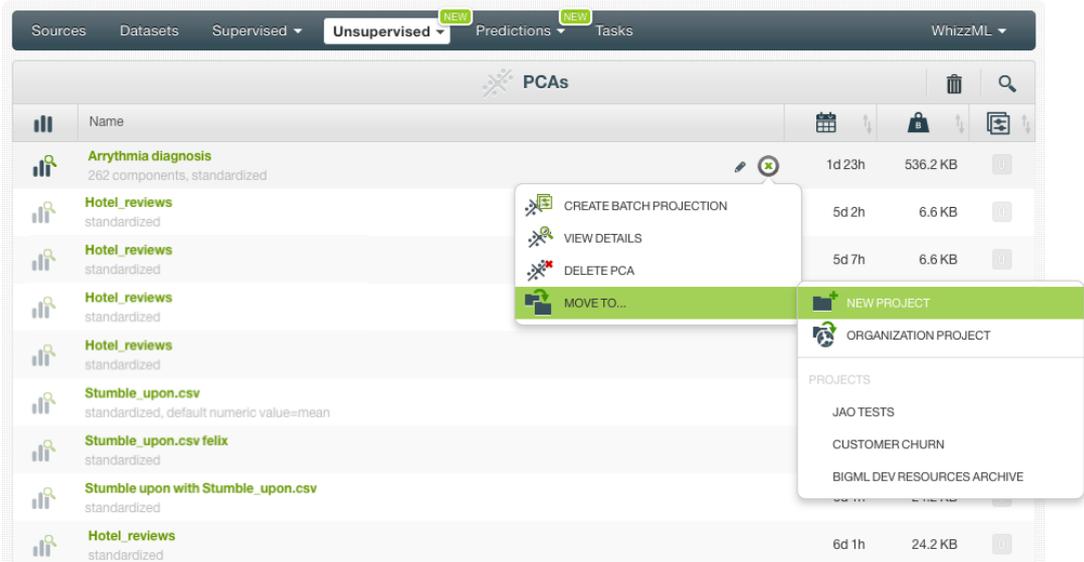


Figure 11.2: Change project from pop up menu

Stopping PCA Creation

You can stop the creation of a PCA before the task is finished by clicking the **DELETE PCA** option in the **1-click menu** from the PCA view. (See [Figure 12.1](#).)

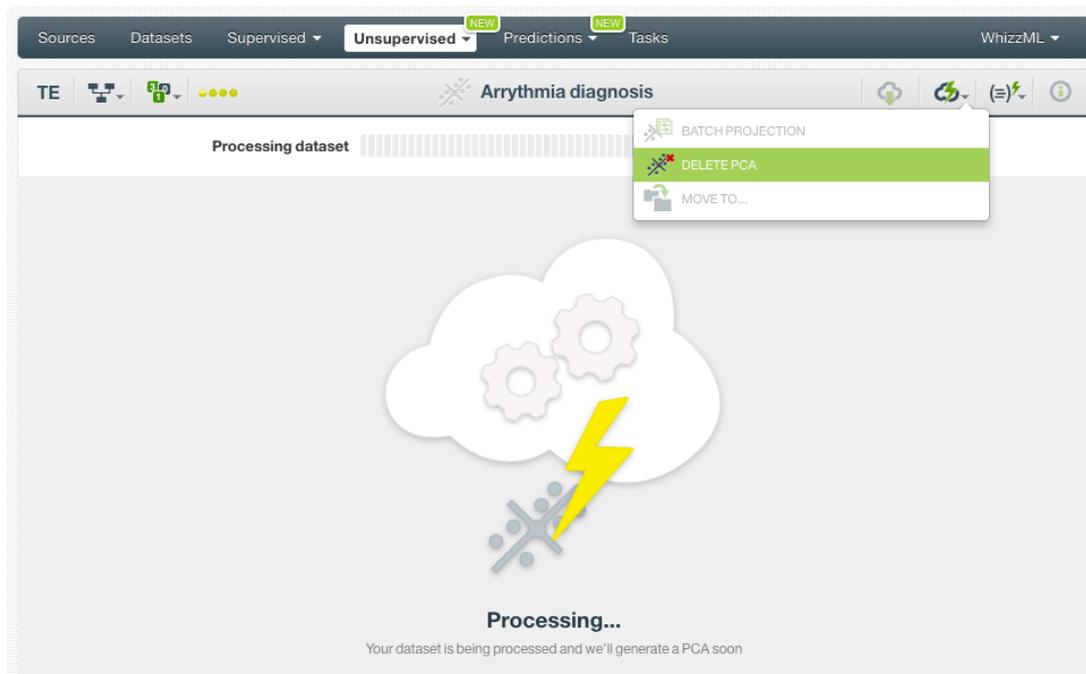


Figure 12.1: Stop PCA creation from 1-click menu

Alternatively, click the **DELETE PCA** in the **pop up menu** from the PCA list view. (See [Figure 12.2](#).)

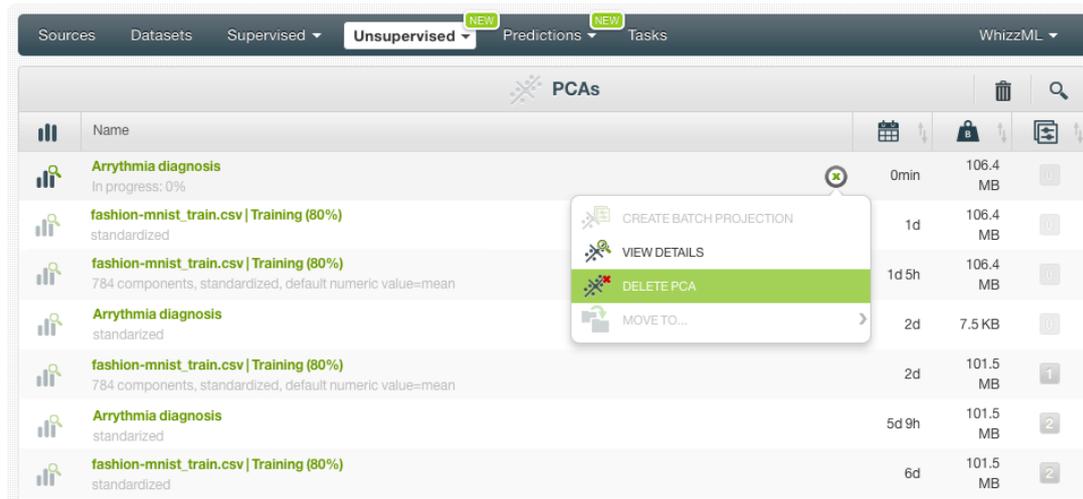


Figure 12.2: Stop PCA creation from pop up menu

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

Deleting PCA

You can delete your PCA by clicking the DELETE PCA option in the **1-click menu** from the PCA view. (See [Figure 13.1.](#))

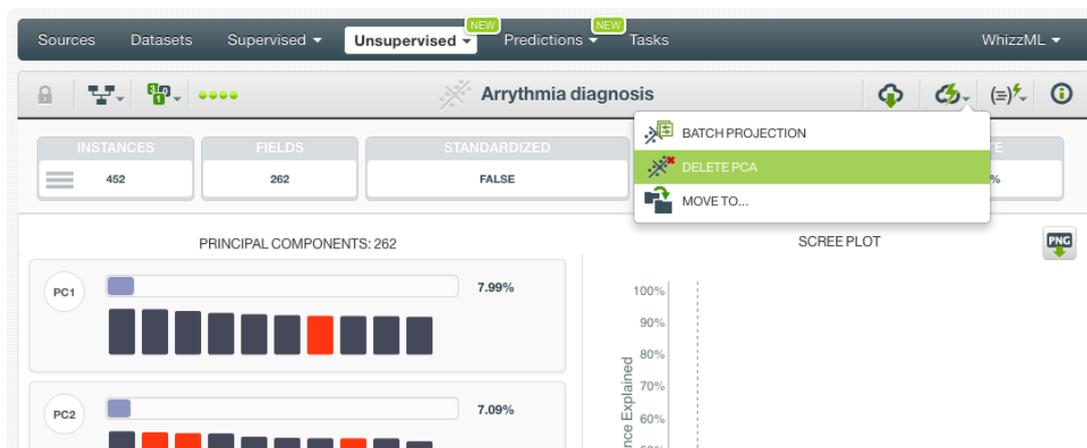


Figure 13.1: Delete PCA from 1-click menu

Alternatively, click the DELETE PCA in the **pop up menu** from the PCA list view. (See [Figure 13.2.](#))

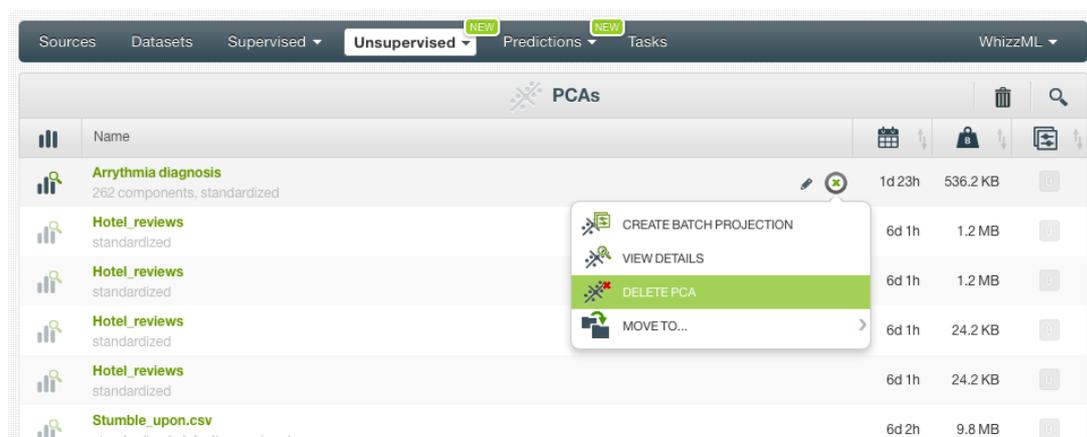


Figure 13.2: Delete PCA from pop up menu

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

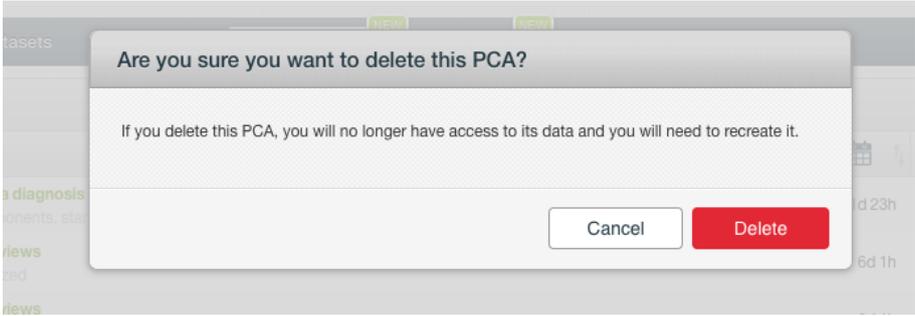


Figure 13.3: Confirmation message to delete a PCA

Takeaways

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

- PCA is an **unsupervised** learning method used to reduce the dimensionality of your datasets.
- BigML PCA is distinct from other approaches to the PCA algorithm because it lets you transform **many different data types** in an automatic fashion that does not require you to configure it manually.
- BigML PCA can handle **numeric** and non-numeric data types, including **text**, **categorical**, **items** fields, as well as combinations of these data types.
- BigML PCA incorporates multiple factor analysis techniques, specifically, **Multiple Correspondence Analysis (MCA)** if the input contains only categorical data, and **Factorial Analysis of Mixed Data (FAMD)** if the input contains both numeric and categorical fields
- BigML PCA also supports missing data.
- To create a PCA you just need an existing **dataset**. Then this PCA model can be used to make a batch projection to calculate the components for any dataset that uses the same input fields as the model. (See [Figure 14.1](#).)
- You can use the **1-click option** to create your PCA or you can **configure** the parameters provided by BigML beforehand.
- When the PCA has been created, you get a list of your PRINCIPAL COMPONENTS ordered by the amount of data variance they explain.
- You can decide to limit the number of your components to create a new dataset by using the SCREE PLOT.
- You can create a new **dataset** from the PCA view.
- You can use your PCA to **make projections** on multiple instances in batch.
- You can create, configure, update, and use your PCA programmatically via the **BigML API and bindings**.
- You can download your PCA to **locally** make projections on new instances.
- You can add **descriptive information** to your PCA.
- You can **move** your PCA between projects.
- You can **share** your PCA with other people using the secret link or embedding them into your own applications.
- You can **stop** your PCA creation by deleting them.
- You can permanently **delete** your existing PCA.

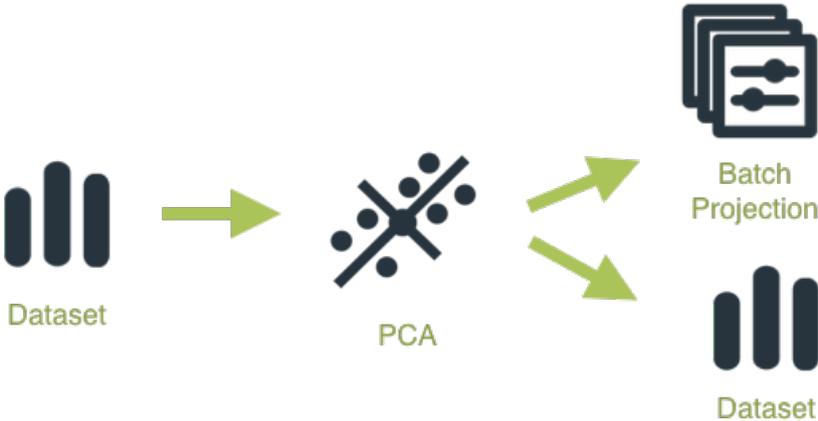


Figure 14.1: PCA workflow

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

List of Figures

1.1	PCA list view	2
1.2	PCA empty list view in the BigML Dashboard	2
1.3	PCA icon	2
2.1	PCA visualization in the scatterplot	5
3.1	Create PCA from 1-click menu	6
3.2	Create PCA from popu up menu	7
4.1	Configure PCA	8
4.2	Standardize option for PCA	9
4.3	Select a default value to replace the missing numerics	9
4.4	Sampling options for PCA	10
4.5	Create PCA after configuration	11
4.6	PCA API request preview	11
5.1	PCA view	12
5.2	PCA total components	13
5.3	Principal components variance	14
5.4	Field weights in components	15
5.5	PCA scree plot	16
5.6	Export scree plot in PNG format	17
5.7	Create dataset from the PCA view	18
6.1	Projections list view	19
6.2	Empty Dashboard projections view	20
6.3	Menu options of the predictions	20
6.4	Batch projection icon	20
6.5	Batch projection from 1-click menu	21
6.6	Batch projection from pop up menu	21
6.7	Select dataset for batch projections	22
6.8	Configuration options displayed and output preview	22
6.9	Create dataset from batch projection	23
6.10	Download batch projection and access output dataset	23
6.11	Set a limit for the number of components to be returned	24
6.12	See the default nueric value to be used in batch projections	25
6.13	Field mapping for batch projections	25
6.14	Output file settings for batch projections	26
6.15	Download batch projection output file	27
6.16	An example of a batch prediction CSV file	27
6.17	Access batch projection output dataset	28
6.18	Batch projection output dataset	28
6.19	Batch projection 1-click menu	29

6.20	Edit projection metadata from More info panel	30
6.21	Markdown editor for projection descriptions	31
6.22	Private link of a projection	33
6.23	Stop batch projection from the 1-click menu	33
6.24	Stop batch projection from the pop up menu	34
6.25	Delete batch projection from the 1-click menu	34
6.26	Delete batch projection from pop up menu	35
6.27	Delete projection confirmation	35
7.1	Downloading PCA	36
9.1	Editing PCAs	39
9.2	Markdown editor for PCA descriptions	40
9.3	Counters for PCA	42
10.1	PCA privacy	43
11.1	Change project from 1-click menu	44
11.2	Change project from pop up menu	45
12.1	Stop PCA creation from 1-click menu	46
12.2	Stop PCA creation from pop up menu	47
13.1	Delete PCA from 1-click menu	48
13.2	Delete PCA from pop up menu	48
13.3	Confirmation message to delete a PCA	49
14.1	PCA workflow	51

List of Tables

- 6.1 Categories used to classify projections by BigML 32
- 9.1 Categories used to classify PCAs by BigML 41

Glossary

Dashboard The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. [ii](#)

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](#), [8](#)

Local predictions the predictions made in your local environment, faster, at no cost, by downloading your model. [36](#)

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

Overfitting the process of tailoring the model to fit the training data at the expense of generalization. [4](#)

PCA Principal Component Analysis is an unsupervised Machine Learning technique used to transform a dataset in order to yield uncorrelated features and reduce dimensionality to build other models. [ii](#), [1](#), [3](#), [19](#)

Principal components the set of uncorrelated variables that PCA calculates as linear transformations of the original dataset field values. Each component has a variance associated which indicates the amount of the variability in the data that it captures. [3](#), [4](#), [7](#), [8](#), [12](#), [14](#), [15](#), [17](#), [19](#), [20](#), [24](#), [27](#), [36](#)

Project an abstract resource that helps you group related BigML resources together. [2](#), [19](#), [44](#)

Projections PCA predictions are called projections in BigML. For PCA, only batch projections are offered in the Dashboard, i.e., projections for multiple instances simultaneously. A batch projection is computed as the inner product of each instance in the components attribute of the PCA and the input vector. [4](#), [18](#), [19](#), [20](#), [36](#)

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](#), [4](#)

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](#), [1](#), [5](#)

Variance (PCA) the variance of each component in PCA means the total variability of the data explained by that component. A higher variance for a given component makes it a better candidate to select as input for other models. [3](#), [4](#), [12](#), [13](#), [15](#), [24](#)

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.

bigml[®]