Datasets with the BigML Dashboard

The BigML Team

Version 2.2

MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

info@bigml.com

Last updated March 27, 2024

About this Document

This document provides a comprehensive description of how to work with BigML datasets using the BigML Dashboard. Datasets is, together with Sources, the two basic building blocks to bring and prepare your data for predictive modeling. Both supervised predictive models (classification and regression) and unsupervised predictive models (cluster analysis, anomaly detection, and association discovery) are built from them.

This document assumes that you are familiar with:

- Sources 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. [6]

To learn how to use the BigML Dashboard to build 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. [7]

Contents

CHAPTER 1

Introduction

A **dataset** is a structured version of your data. BigML computes some basic statistics for each one of the fields of these datasets. The main goal of datasets is enabling effective wrangling of your data, so you can build the right BigML model for your problem. This is a key step to ultimately achieve the best results for your Machine Learning tasks.

In this chapter we assume you understand what a source is, the formats BigML accepts, the types of fields allowed in the source, the types of sources BigML supports, size limits, etc. If you would like to dive deeper into sources and learn all the details, we recommend that you read the **Sources with the BigML Dashboard document** [5].

BigML also provides you with a large variety of datasets, available in BigML Gallery, which you can clone and reuse. We explain how to get them in Section 13.1.

This chapter contains comprehensive description of BigML datasets including how they can be created with just 1-click (see Chapter 3), and all configuration options available (see Chapter 4). Chapter 2 explains the technicalities behind datasets and how BigML computes statistics for each field. Chapter 5 helps you understand how BigML represents datasets in the Dashboard and the options available for you to configure your dataset to best fit your needs.

In addition, BigML presents the **dynamic scatterplot visualization**, a way to analyze your data to get better features for your Machine Learning models. (See Chapter 6 for more details). You can also find other options like filtering and sampling your dataset (see Chapter 7) and transforming your data, such as creating new fields, aggregating instances, joining and merging different datasets (see Chapter 8). The process of transforming your dataset is a fundamental step towards the creation of an effective Machine Learning solution. Moreover, you can add descriptive information to your dataset (Chapter 11), export it to several formats and download it to your machine (see Section 9.2 and Section 9.1), move it to another project (Chapter 14), and delete it permanently from your account (Chapter 16).

In BigML, the second tab of the main menu of your Dashboard allows you to list all of your available datasets (Figure 1.1). In this **dataset list view** you can see for each dataset, the **Source Details**, **Name**, **Age** (time since the source was created), **Size**, **Number of Models, Ensembles, Logistic Regressions, Clusters, Anomalies, and Associations** created. The SEARCH menu option in the top right corner of the **dataset list view** allows you to search your datasets by name. This is very handy when you have a large number of datasets, and you cannot list them all in the same page.

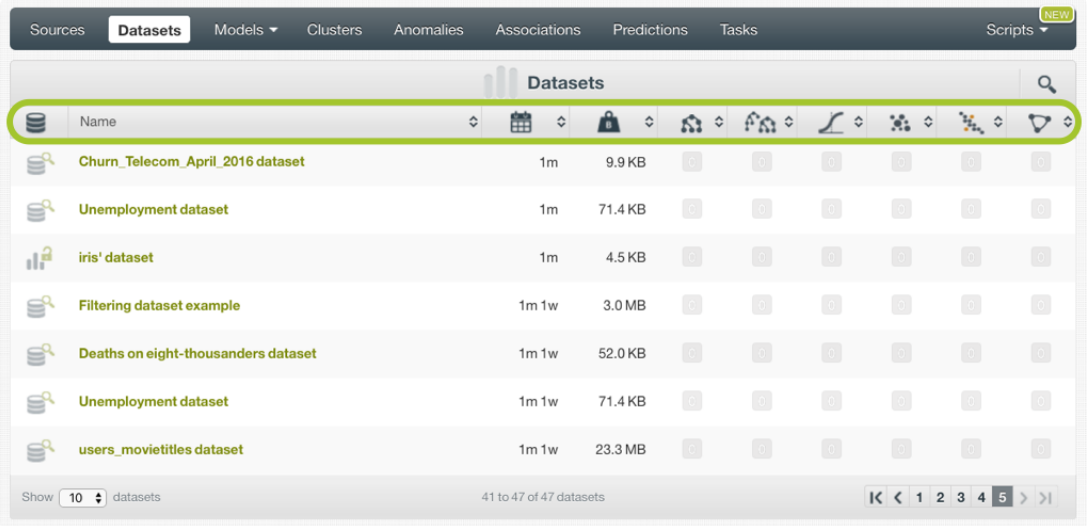

Figure 1.1: Datasets list view

By default, every time you start a new project, your list of datasets will be empty. (See Figure 1.2.)

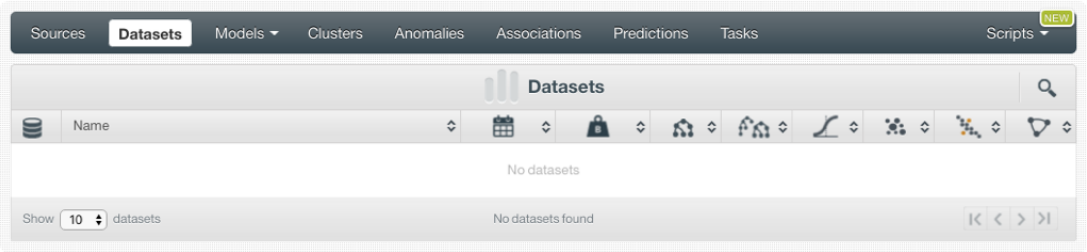

Figure 1.2: Empty Dashboard dataset view

Finally, the icon in Figure 1.3 represents a dataset.

Figure 1.3: Dataset Icon

CHAPTER 2

Understanding Datasets

A **dataset** is a structured version of your data. BigML computes both general statistics for the dataset and individual statistics per field. This chapter describes the technicalities behind datasets.

Figure 2.1 shows how BigML lists all fields, the field type, and the general statistics, including:

- **Count**: the number of instances containing data for this field.
- **Missing**: the number of instances missing a value for this field.
- **Errors**: information about ill-formatted fields that includes the total format errors for the field and a sample of the ill-formatted tokens.

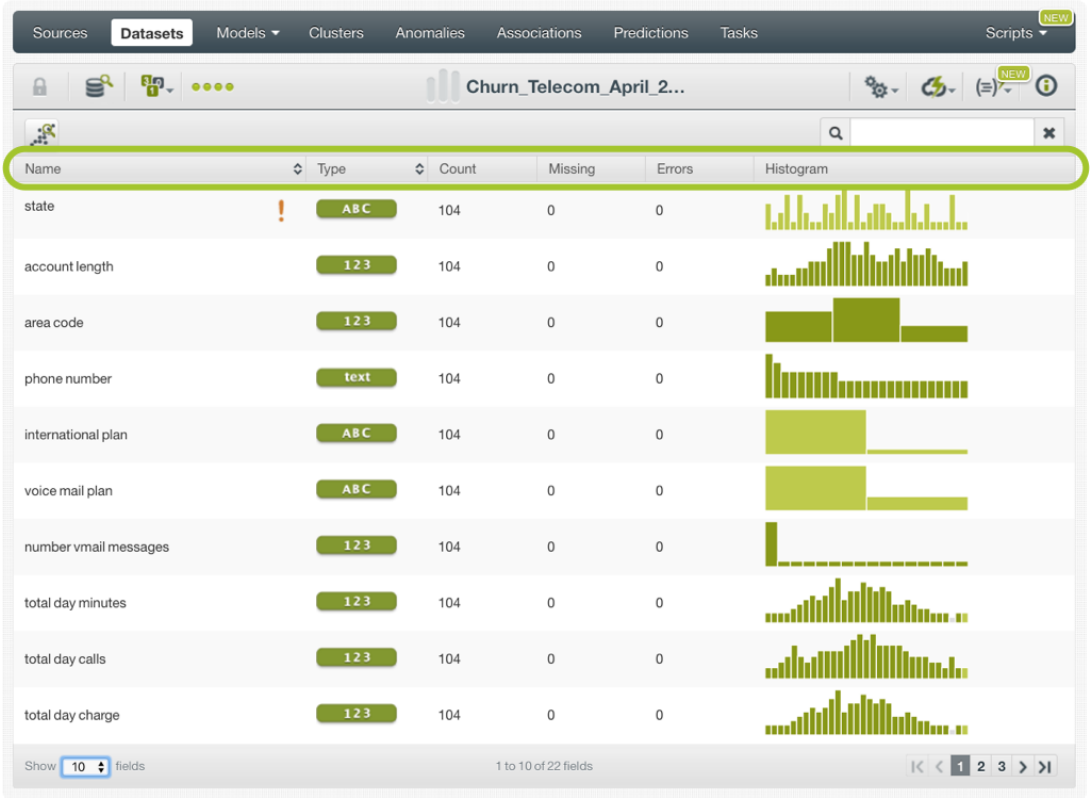

Figure 2.1: Dataset basic view

The histograms communicate the underlying distributions of your data. Depending on the size of your dataset and the number of unique values, these histograms may either be exact or may be approxima-

tions. Read this blog post[1] for more details.

The following Subsections describe how BigML processes the data and computes the statistics differently for each field type.

2.1 Statistics for Numeric Fields

BigML computes the measures below based on all instances of a given **numeric field** and displays these measures in a histogram, as seen in Figure 2.2:

- **Minimum**[2]: the minimum value found in this numeric field.
- **Mean**[3]: the arithmetic mean of non-missing field values.
- **Median**[4]: the approximate median of the non-missing values in this numeric field.
- **Maximum**[5]: the maximum value found in this numeric field.
- **Standard deviation**[6]: the unbiased sample standard deviation.
- **Kurtosis**[7]: the sample kurtosis. A measure of 'peakiness' or heavy tails in the field's distribution.
- **Skewness**[8]: the sample skewness. A measure of asymmetry in the field's distribution.

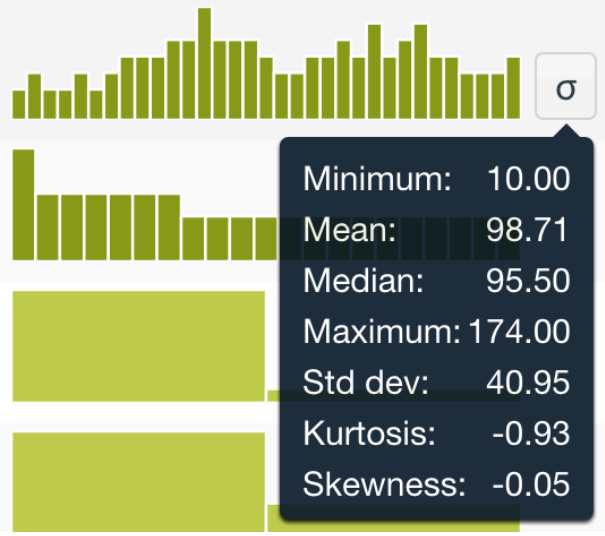

Figure 2.2: Example of histogram for numeric fields

2.2 Categorical Fields

BigML creates one bin per label contained in a **categorical** field. Each bin contains the number of instances that have a specific label, e.g. the example shown in Figure 2.3 has six labels, therefore the histogram shows six bins, and 245 instances of this field are labeled as "Spain".

[1] https://blog.bigml.com/2012/06/18/bigmls-fancy-histograms/
[2] https://en.wikipedia.org/wiki/Maxima_and_minima
[3] https://en.wikipedia.org/wiki/Arithmetic_mean
[4] https://en.wikipedia.org/wiki/Median
[5] https://en.wikipedia.org/wiki/Maxima_and_minima
[6] https://en.wikipedia.org/wiki/Standard_deviation
[7] https://en.wikipedia.org/wiki/Kurtosis
[8] https://en.wikipedia.org/wiki/Skewness

Figure 2.3: Example of histogram for categorical fields

Note: when BigML encounters binary formatted fields (all values 0 or 1), it treats them as categorical rather than numeric. You may override this default in the source configuration. (See the section Updating Field Types of the Sources with the BigML Dashboard[9] [5].)

BigML allows you to have up to 1,000 different labels in a categorical field.

2.3 Text and Items Fields

Item fields are similar to categorical fields, with the key difference that items may be provided as a set (e.g., items purchased together). Text fields are similar to item fields, but optionally include text processing (such as stemming, described below). BigML processes how frequently a given **term** appears in a field, and it shows the number of instances that used it. For this, BigML offers the tag cloud, an alternative representation to a histogram. Tag clouds are available only for text and items fields, and they are represented with a different icon placed next to the histogram, as seen in Figure 2.4. This example shows that 183 instances of this field use the term “cava”.

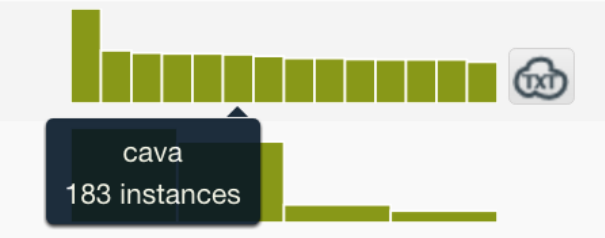

Figure 2.4: Example of histogram for text and items fields

For more details, once you click on the TXT icon to discover how often a given term appears in your dataset. (See Figure 2.5.) The bigger the term, the more frequently repeated. Check how many times each term is repeated by mousing over each term, e.g., “chardonnay” appears 155 times in this field. You can download the tag cloud in the SVG or PNG format by clicking the SVG or PNG button.

[9] https://static.bigml.com/pdf/BigML_Sources.pdf

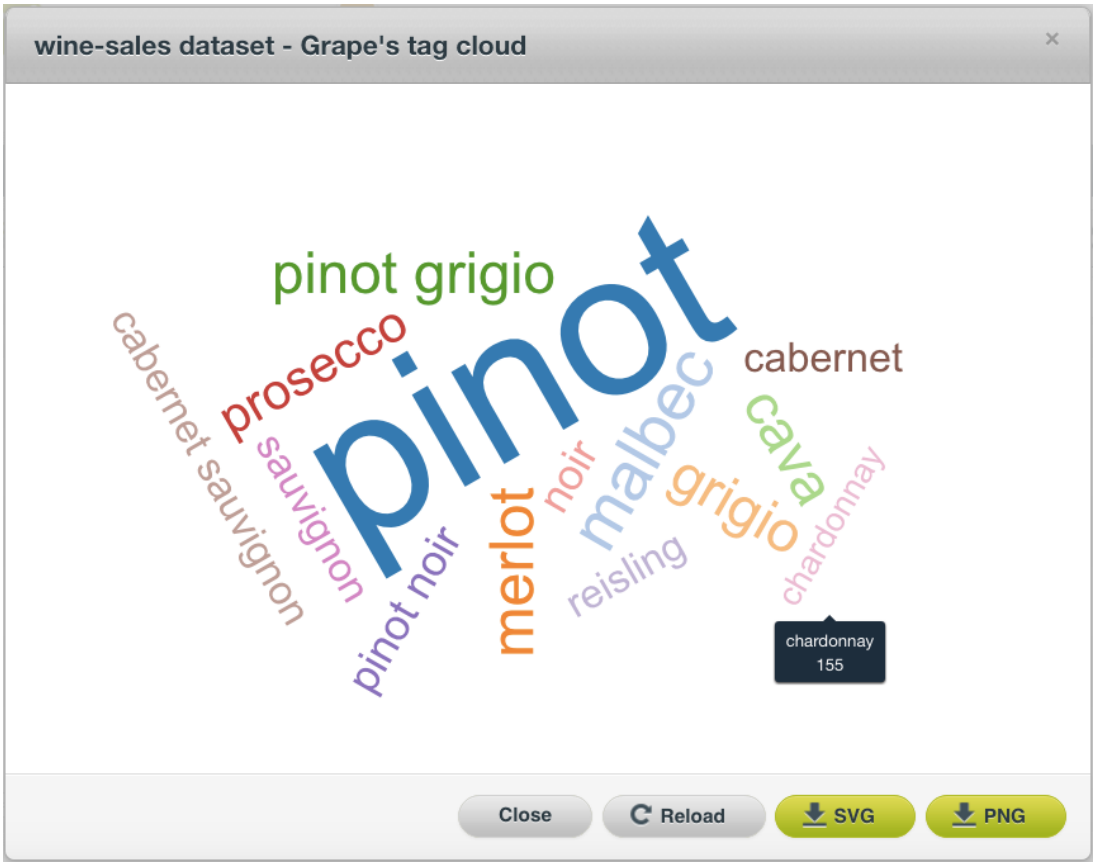

Figure 2.5: Example of a tag cloud

BigML can find up to 1,000 terms across all your text and items fields of your dataset. To find these terms, BigML parses the text considering the text analysis options configured for your source. (See the section Text Analysis of the Sources with the BigML Dashboard[10] [5].) If you used BigML default term **tokenization**, all terms will be separated considering spaces and other symbols (comma, colon, semicolon, tab, etc). Each block of text between separators is considered a term.

- If the **stopwords** option is enabled, BigML eliminates words like: a, the, is, at, on, which, etc.
- If the text field has **stemming** enabled, all terms with the same root are considered one single value; e.g., if stemming is enabled the words "great," "greatly," and "greatness" would be considered one value instead of three different values. BigML calculates how often each of these terms appear in the fields. If "great" appears 12 times and "greatness" appears eight times, the term count will account for 20 instances of the term "great."
- BigML also allows you to differentiate words when they contain upper or lower cases. When **case sensitivity** is enabled, "Great" and "great" will count as two different words in the tag cloud, otherwise they would be treated as the same word.

If BigML incorrectly detects a numeric or categorical field as a text field, you may override the field type during source configuration. (See the section Updating Field Types of the Sources with the BigML Dashboard[11] [5].)

2.4 Date-Time Fields

BigML expands each **date-time** field into up to eight numerical fields (e.g., day of the week, day of the month, etc. You can enable or disable automatic generation by switching the expand date-time

[10] https://static.bigml.com/pdf/BigML_Sources.pdf
[11] https://static.bigml.com/pdf/BigML_Sources.pdf

fields option in the configure source menu. (See the section Date-time of the Sources with the BigML Dashboard[12] [5].) When disabled, potential date-time fields will be treated as either categorical or text fields.

These expanded fields are treated as numeric fields; therefore BigML computes the same statistics mentioned above for numeric fields (Section 2.1.) Figure 2.6 shows an example of two numeric fields generated from a date-time field. The first field focuses on the year; that is why the field type has **YYYY** bold faced, while the second is for the month. The bold face in the type of field column indicates the focus of the generated field.

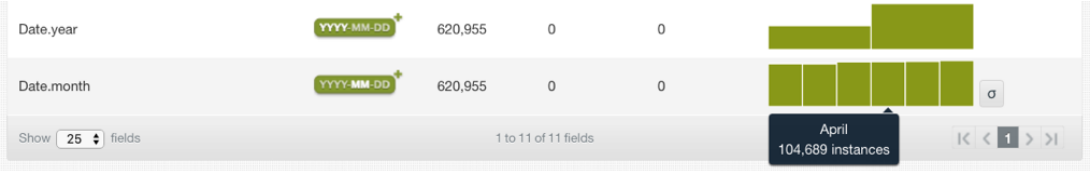

Figure 2.6: Example of numeric fields automatically generated from a date-time field

2.5 Image Fields

For every image in a dataset, there are at least two fields associated with it. The field with an **image** type points to the normalized version of the image file itself, while the field with a **path** type represents its filename.

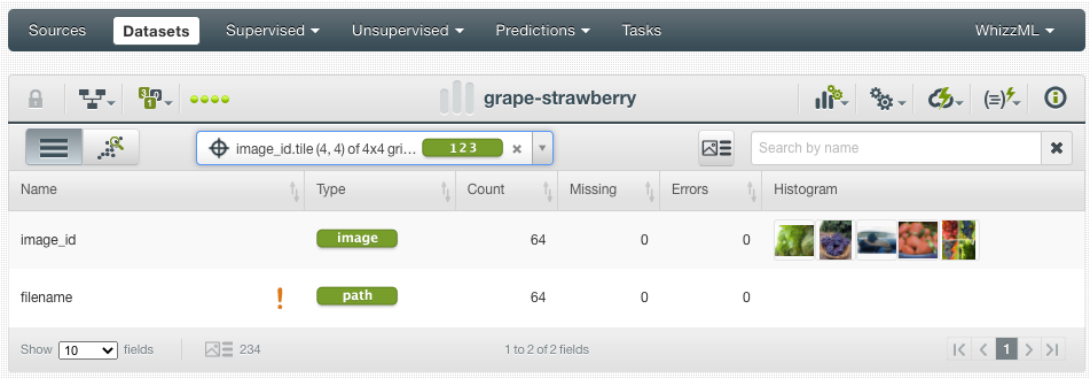

Figure 2.7: Example of a dataset having two image-related fields

In the **image** field, users can preview images in its "Histogram" column. Click on the Refresh button on the right next to the images to load a different set of images for preview.

Because all filenames are unique, the **path** field is set to non-preferred by default.

Oftentimes, there are additional fields associated with images. A common situation is automatic image labeling, when the folder names are used as image labels. In such cases, BigML automatically extracts the innermost directory in the filenames and assigns them as the labels of the images, respectively. (See the section Automatic Image Labels of the Sources with the BigML Dashboard[13] [5].)

[12] https://static.bigml.com/pdf/BigML_Sources.pdf
[13] https://static.bigml.com/pdf/BigML_Sources.pdf

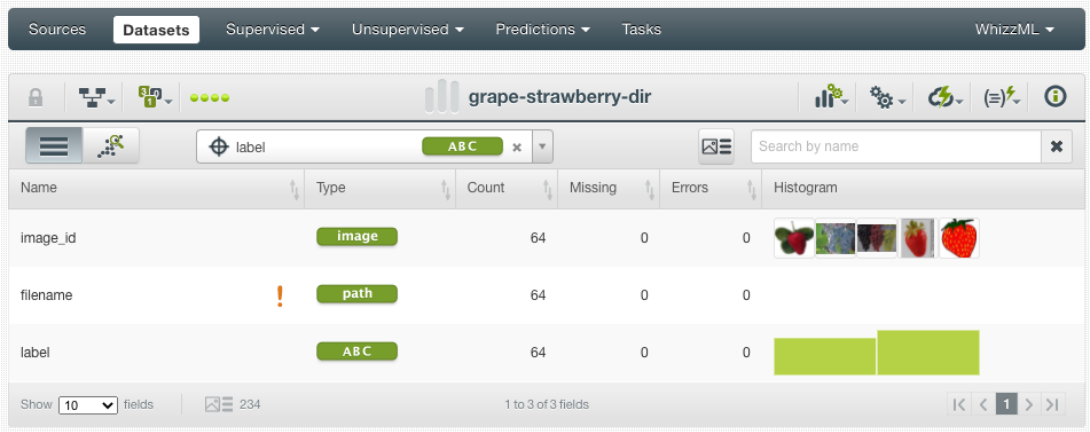

Figure 2.8: Example of a dataset having three image-related fields

When an image composite source is created, BigML automatically generates a set of image features for each image. Users can also configure and select different combinations of image features, or disable them. After a dataset is created, all fields of the image features are hidden in the dataset view, same as in the source view. However, users can click on the "show image features" icon next to the search box as shown below:

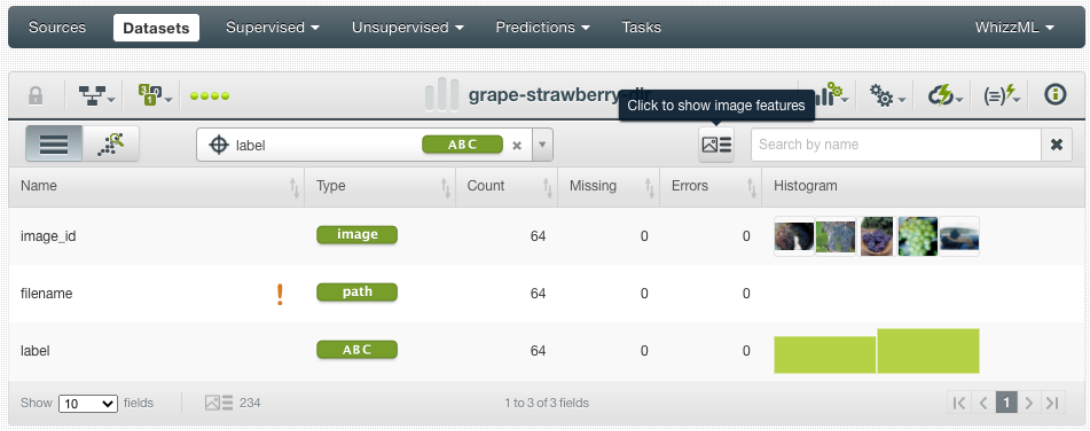

Figure 2.9: The icon to toggle between showing and hiding the fields of image features

Then users can preview all fields of the image features that came with the dataset, and their statistics in the histogram column:

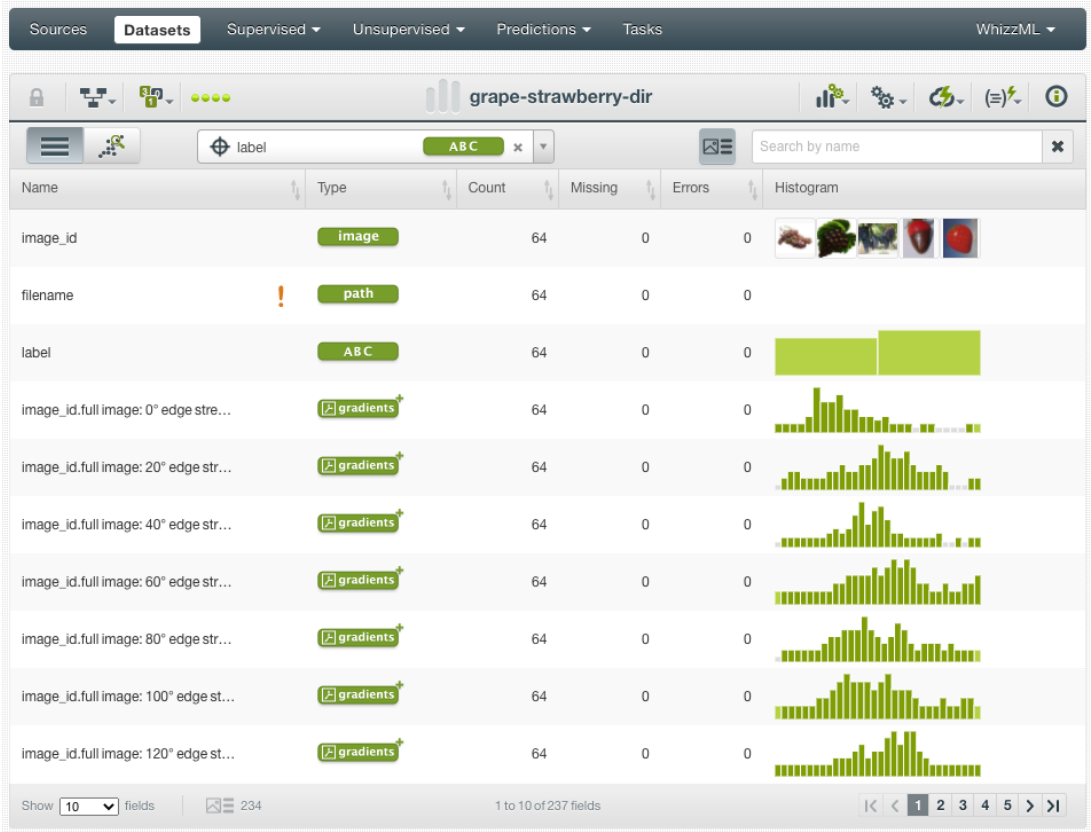

Figure 2.10: Previewing the fields of image features and their statistics

When the image feature fields are shown, users can click on the same icon to hide them.

For information about the image features and how to configure them, please refer to the section Image Analysis of the Sources with the BigML Dashboard[14][5].

[14] https://static.bigml.com/pdf/BigML_Sources.pdf

CHAPTER 3

Creating Datasets with 1-Click

In BigML you can you can create datasets in two ways: with just **1-click**, or **configuring** certain options from a source previously imported into BigML. This section describes the 1-click option.

Create your dataset from the **source view** using the 1-CLICK DATASET option in the **1-click action menu**. (See Figure 3.1.)

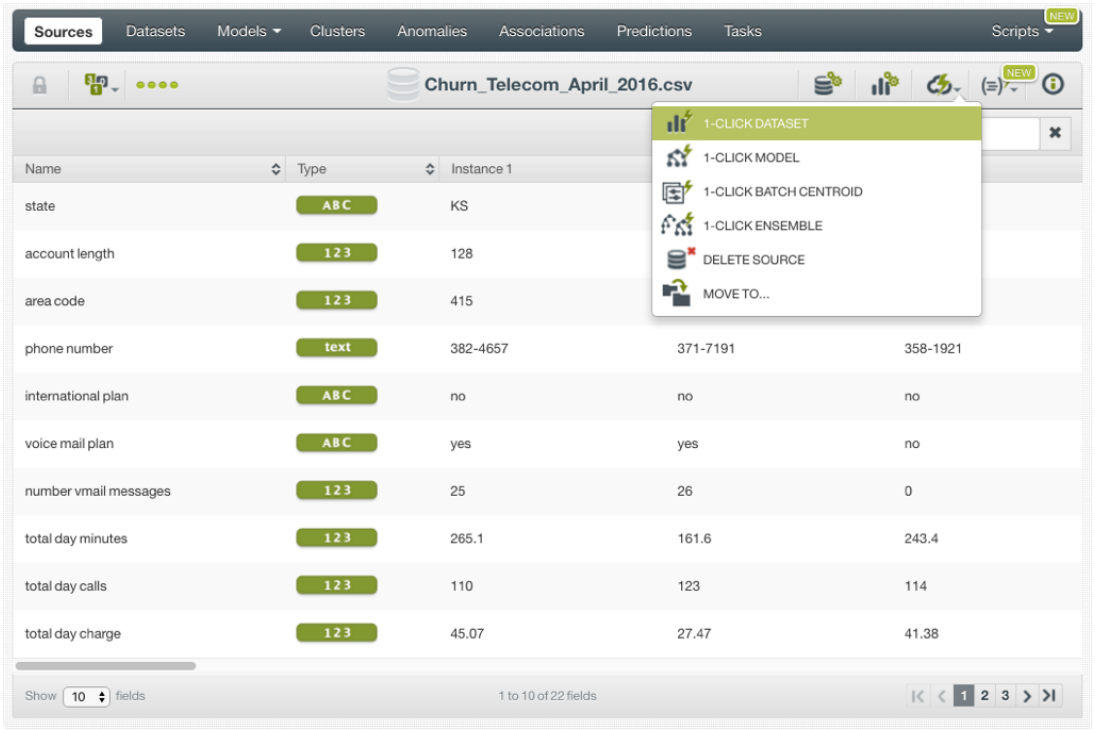

Figure 3.1: Creating a dataset from 1-click action menu

Alternatively, you can create a dataset using the **pop up menu** by selecting 1-CLICK DATASET from the **source list view**. (See Figure 3.2.)

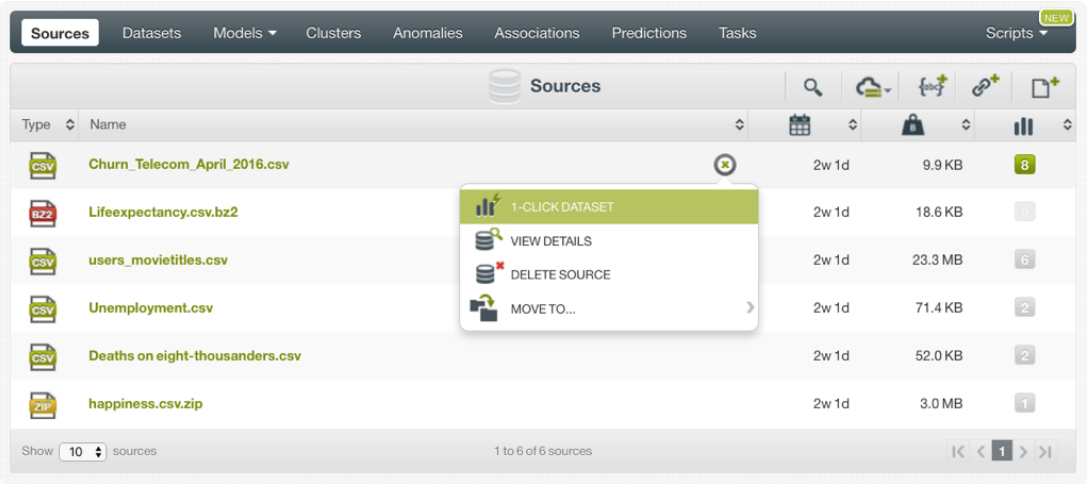

Figure 3.2: Creating a dataset from pop up menu

Note: when creating a dataset from an open composite source, the composite source will be closed during the process. Please refer to **Soruces with the BigML Dashboard document.** [5] for more information about open and closed composite sources.

When ready, your dataset will automatically be displayed on your Dashboard. (See Chapter 5.)

CHAPTER 4

Dataset Configuration Options

In addition to the **1-click** option to create a dataset, explained in Chapter 3, BigML also allows you to **configure** your dataset assigning a different name, and selecting the percentage of your source to be used to create the dataset. You can also include or exclude certain fields as you wish. The following subsections cover the available options.

4.1 Dataset Name

When you click on the CONFIGURE DATASET menu option, in the **source view** you want to use to create your dataset, you get access to the **dataset configuration panel**. Change the default name provided by BigML by typing a new name in the **dataset name** box. (See Figure 4.1).

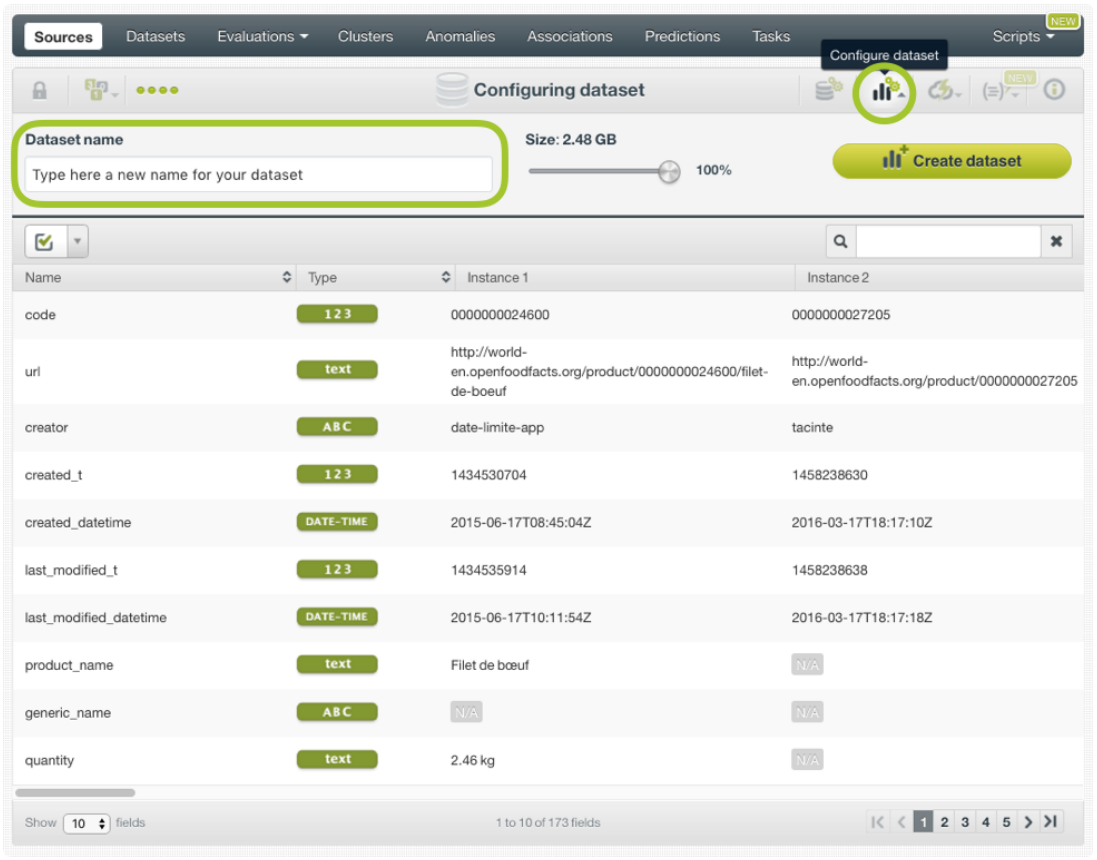

Figure 4.1: Configuration panel to assign a new name to your dataset

4.2 Dataset Size

By default, BigML uses all the data you have in your source. However, when you deal with big datasets, you can accelerate exploration of your data by setting aside a subset of your source, and build your dataset with only a small part of your data. Select the percentage you want to use to create your dataset by moving the highlighted size slider in Figure 4.2. In this example we are using the 10% of the source imported into BigML.

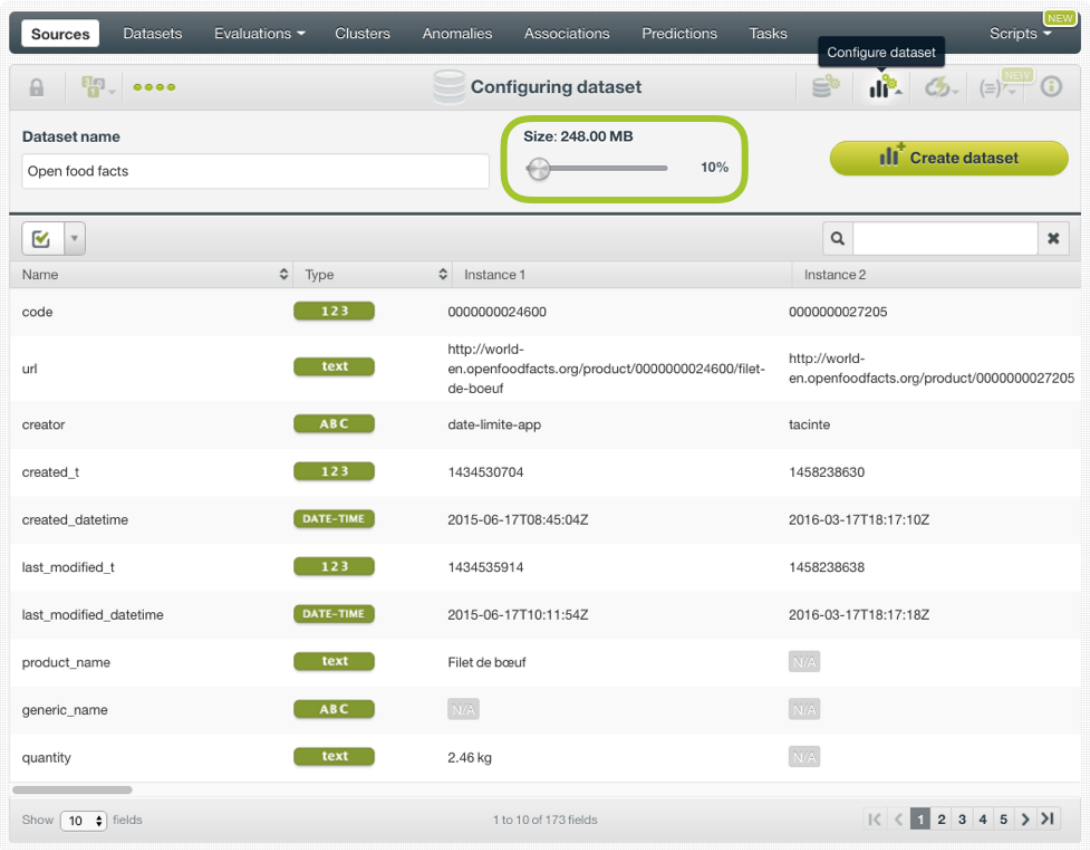

Figure 4.2: Configuration panel to select the size of your source

This option is not available when your source contains images. However, you can always use sampling (Section 7.2) after the dataset is created.

4.3 Including or Excluding Fields

The dataset configuration options allow you to include or exclude fields when creating a dataset. You can include or exclude them all, as seen in Figure 4.3. Or deselect the box associated with the fields you do not want to use, as shown in Figure 4.4. The fields you deselect will not be included in the dataset, therefore BigML will not consider them when later building your model.

Obviously, if you exclude all your fields, no dataset will be created. However, this option might be useful when your source has lots of fields but you only want to include a few of them. In this case, it would be faster to exclude them all and select the few fields you want to include one by one.

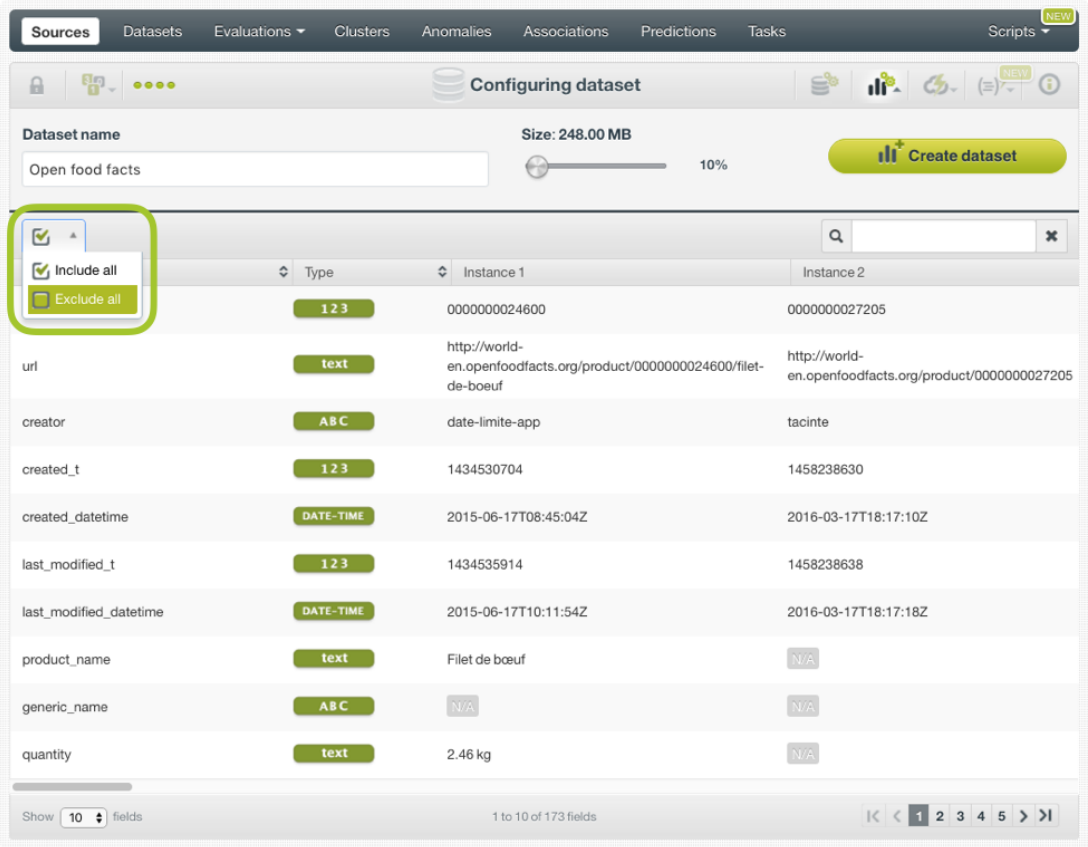

Figure 4.3: Configuration panel to include and exclude fields

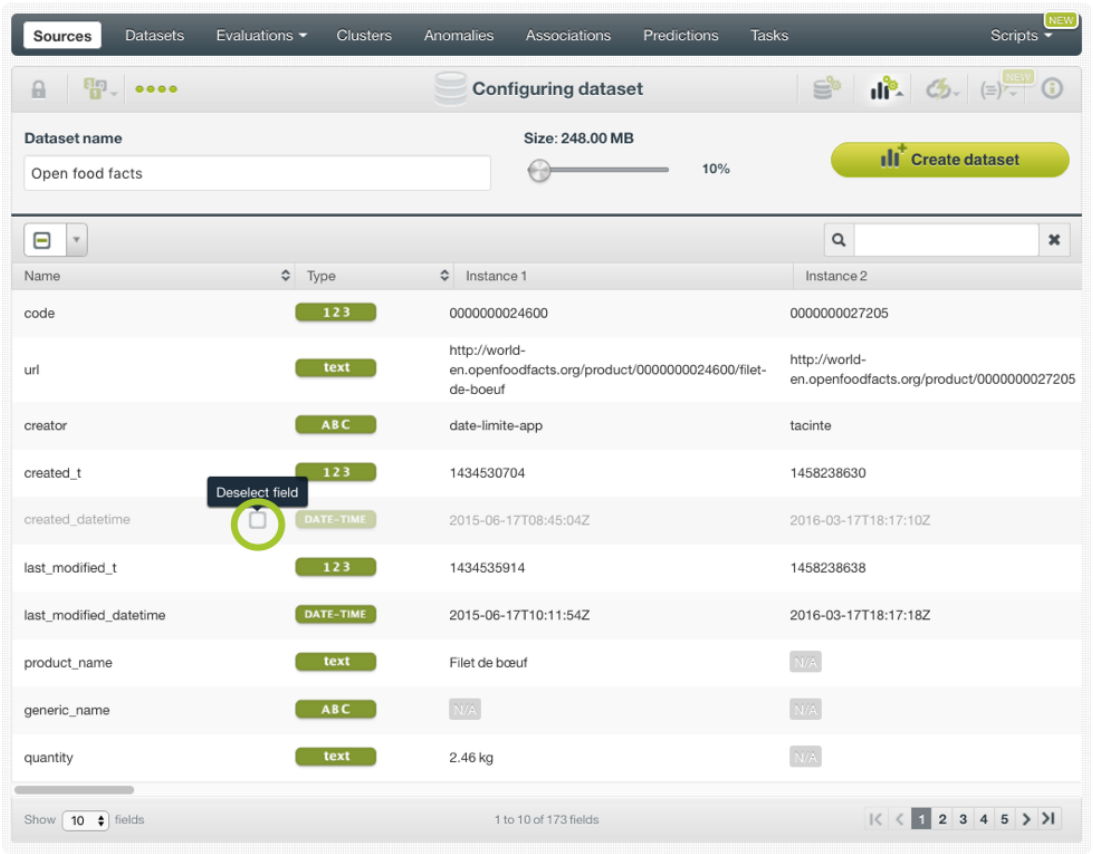

Figure 4.4: Example of a deselected field

Once you have assigned a new name to your dataset, set the percentage of your source to use, and selected the fields you need, you are ready to click the Create dataset button shown in Figure 4.4. This action will bring you to the dataset visualization, explained in the following Chapter 5.

CHAPTER 5

Visualizing Datasets

After creating a dataset, BigML automatically displays it in the **dataset view**, described in Section 5.1. The following subsections describe the dataset layout and how you can interpret your dataset.

5.1 Dataset Layout

Figure 5.1 shows the general overview of a dataset. At the top part, from left to right, you find the **navigation and status** menu. This menu lets you see the PRIVACY status of your dataset, VIEW THE SOURCE used to create this dataset, view resources created with this dataset and quickly access them through the COUNTERS, and the creation STATUS of other resources you are creating with this dataset. Next to this menu you see the **dataset name**, followed by the **actions and information** menu, at the top right. This menu allows you to access the CONFIGURE OPTIONS, the 1-CLICK ACTIONS, the 1-CLICK SCRIPTS, and the MORE INFO panels. Both menus, **navigation and status** and **actions and information** are explained in Subsection 5.1.1.

Below the navigation and status menu you can see the dynamic scatterplot icon, which gives you access to the **dynamic scatterplot view**, a different way to visualize your dataset (explained in Chapter 6). On the right side, below the actions and information menu, there is a **search box** that lets you quickly find the fields containing the word you type in.

The middle part of Figure 5.1 shows a table with six columns, where the rows are the fields and the columns have the following information for each one of the fields: field **name**, the field **type**, **count** (instances with valid values), **missing** (instances with missing values), **errors** (instances with errors), and **histograms**. This information represents the general statistics computed by BigML (count, missing, and errors), and the statistics for each field displayed in a histogram, explained in Chapter 2.

The red exclamation mark in the “state” field means that BigML has discarded this field as input field for training a model, since this field may have similar values for all the instances or very different values for each of the instances.

At the bottom left corner of Figure 5.1, there is a dropdown that lets you select the number of fields you want to see in the same dataset view: 10, 25, 50 or 100. In the center, you can see the number of fields of this view and total number of fields contained in your dataset. Finally, if your dataset has a large number of fields that cannot all fit in the same dataset view, you can select the page at the bottom right corner.

Figure 5.1: Dataset layout overview

5.1.1 Dataset Top Menus

- **Navigation and Status**:

  At the top left corner of the **dataset view** you can see the menu options shown in Figure 5.2.

Figure 5.2: Navigation and status menu options of the dataset list view

  – The PRIVACY menu option indicates whether the dataset you have open in the dataset view is **public** in the BigML Gallery or **private**, which means that only you can view that dataset unless you decide to share it with others. This process is explained in Section 13.2.

  – The VIEW SOURCE menu option lets you see the source used to create the dataset you have open in the dataset view. If you deleted the source after creating the dataset, the view source menu option will no longer be a link to the source, since there will be no source available. This is indicated in the **dataset list view**, where the source icon will show a red cross. (See Figure 5.3.)

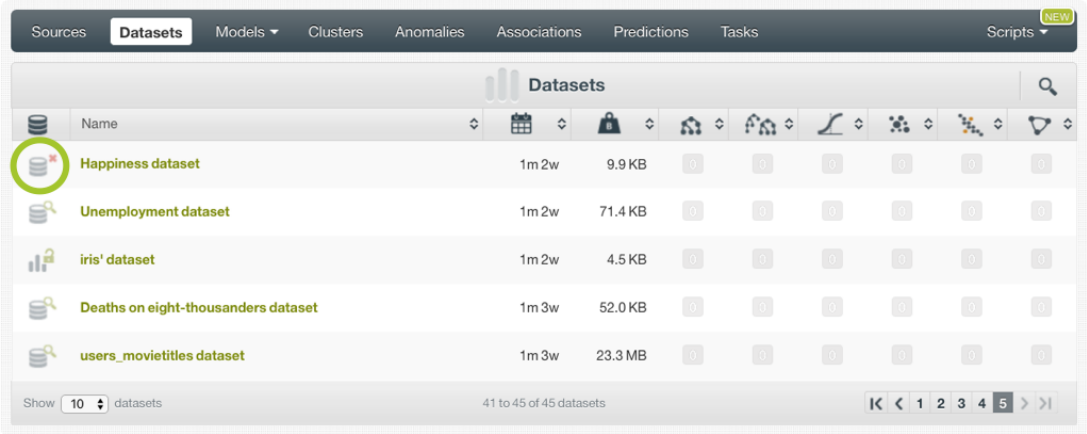

Figure 5.3: Dataset list view shows a source that has been deleted

- The COUNTERS menu option allows you to see the resources created with the dataset you have open in the dataset view.
- The RESOURCE STATUSES menu option indicates when a dataset is being used to create a resource. Thus, when you perform a task, this menu option remains *completed* when there are no resources requested, and the status changes to: *unknown, error found, waiting, queued, started, in-progress, summarized, and completed*, when you request a task. In many cases, resources progress so quickly through some of the statuses that you will not see them appear on the Dashboard. The statuses you will see most often are *in-progress* and *completed*.

  In BigML, tasks are asynchronous, this means that the request to create a resource exits right away without waiting for its completion. You can request the creation of several resources in a very short period of time, and they will either run in parallel or will be queued, so the order of tasks is maintained. Some tasks may take a few minutes to process, depending on the size of your dataset and the subscription plan[1] you have purchased, which determines how many tasks you may run in parallel at a given time.

[1] https://bigml.com/pricing#subscriptions

- **Actions and Information**:

  At the top right corner of the **dataset view** you can see the menu options shown in Figure 5.4.

Figure 5.4: Actions and information menu options of the dataset list view

  - The CONFIGURE OPTIONS gives you access to the different configuration panels for models, ensembles, logistic regressions, clusters, anomalies and associations. This menu also lets you access the configuration panels to split your dataset, sample it, filter it, and to add new fields to your dataset. These options are explained in Section 7.1, Section 7.2, Section 7.3 and Section 8.1, respectively.
  - The 1-CLICK ACTIONS gives you access to create your models, ensembles, logistic regressions, clusters, anomalies, or associations with just 1-click with default values. This menu also lets you automatically split your dataset, export it in the CSV or Tableau file format, move it to other projects, and delete it. These options are explained in Subsection 7.1.1, Section 9.1, Section 9.2, Chapter 14 and Chapter 16, respectively.
  - The 1-CLICK SCRIPTS , lets you add your Machine Learning Scripts to execute them anytime, with just 1-click, regardless of the view, from the BigML Dashboard.
  - The MORE INFO option leads you to three panels with information about your dataset. The **details panel** shows the size, number of fields, and number of instances contained in your dataset. The **info panel** where you can update the name of your dataset, add a description, tags, and assign a category (see Chapter 11). And the **privacy panel** with privacy details of your dataset (see Chapter 12).

5.2 Updating Fields

You can change field **names**, **labels or tags**, and **descriptions** by clicking the edit icon next to the field name. You can also set a field as non-preferred, which will not be taken into account to generate your models. From the same **editing panel**, you can set a field as the objective field for models, ensembles, and logistic regressions. (See Figure 5.5.) By default, BigML takes the last valid field (numeric or categorical fields) as the objective field. Please read the section Objective Field of the Classification and Regression with the BigML Dashboard[2] [3] to learn more about the objective field.

Note: the non-preferred fields and the objective field are inherited when you clone a dataset from the BigML Gallery. Also when you create new datasets from existing ones by splitting into subsets, sampling, filtering, or adding new fields.

[2] https://static.bigml.com/pdf/BigML_Classification_and_Regression.pdf

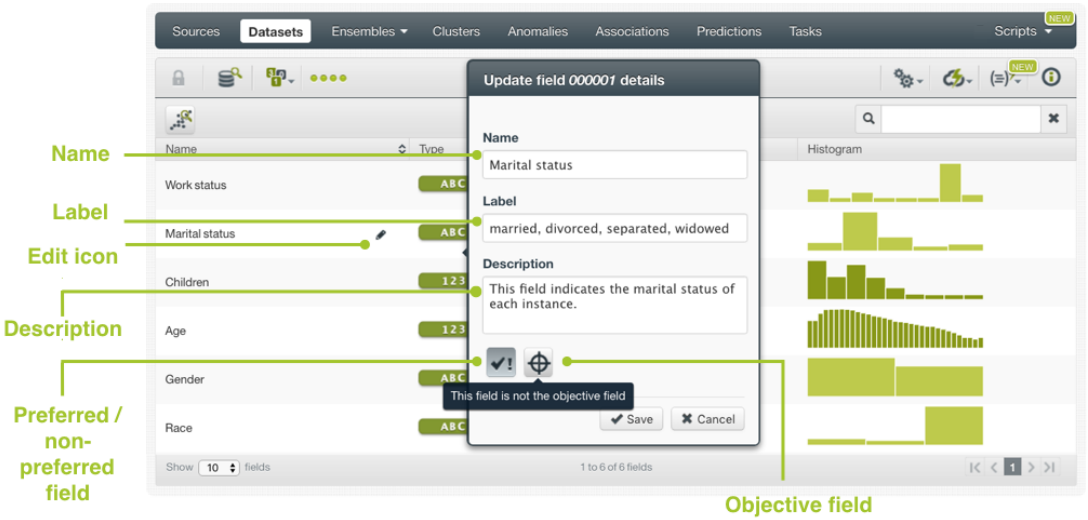

Figure 5.5: Updating field view

CHAPTER 6

Dynamic Scatterplot View

The **dynamic scatterplot view** is a graph that BigML provides to visualize a sample of your dataset (maximum of 500 instances) differently. The scatterplot is very useful to detect interesting patterns in your data, correlations among your fields, or anomalous data points amidst other observations.

To visualize your dataset fields in the dynamic scatterplot view you need to click on the scatterplot icon (see Figure 6.1).

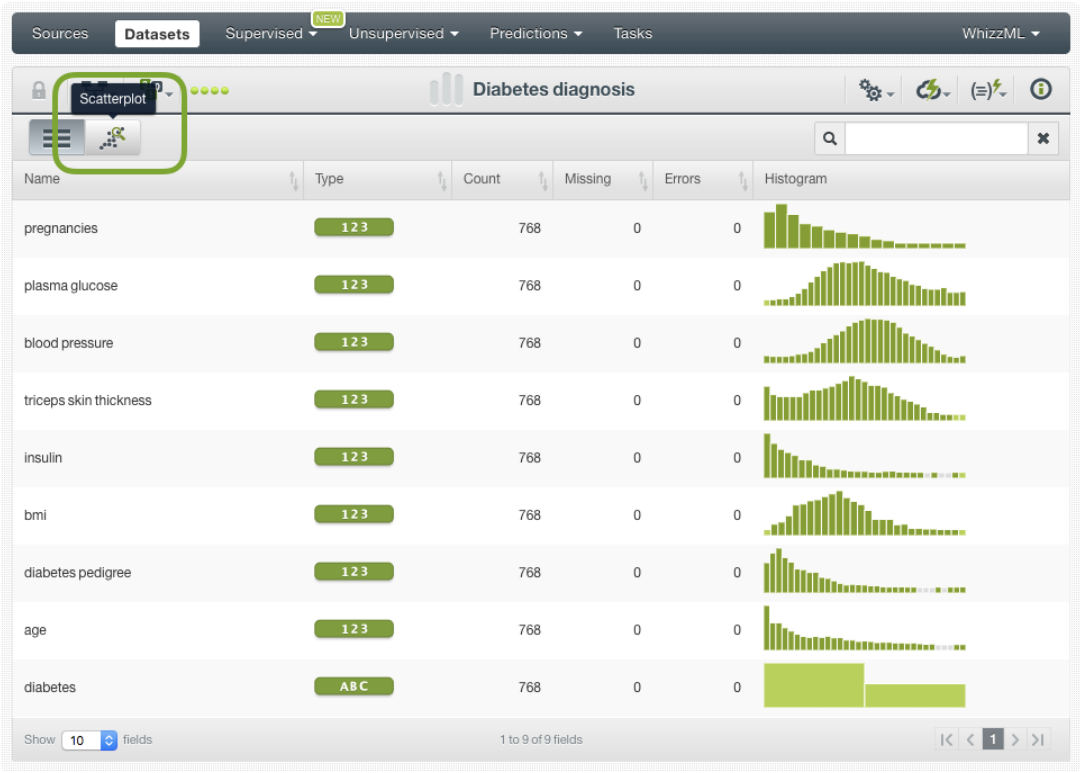

Figure 6.1: Access to the dynamic scatterplot view

6.1 Dynamic Scatterplot Chart Options

Figure 6.2 shows how BigML displays the dynamic scatterplot view. This view has two parts: a **graph** in the center of the image, and a **data inspector** on the right hand side.

Figure 6.2: Dynamic scatterplot view

1. The **graph** in the center of the image is the visualization itself. You can configure the graph options highlighted in Figure 6.2 using:
   - **Fields selectors**: you can select two fields from your dataset, one field for each axis (Y and X). Fields must be either categorical or numeric (text, items and date-time fields are not supported as scatterplot axes).
   - **Logarithmic scale**[1]: either axis with numeric fields may be plotted logarithmically, useful when your values have a large range.
   - **Regression line**[2]: you can show and hide a regression line when the two selected fields are numeric. The regression line fits a simple linear regression to your data, useful for highlighting trends.
   - **Create a dataset**: you can select an area in the chart (by clicking and dragging in the chart surface) and create a new dataset containing only the data points in the selected area.
   - **Get new sample**: when your dataset is very large, BigML automatically takes a random sample of 500 instances so you can better visualize the data points in the chart. With this option you can visualize a new sample of your dataset.
   - **Export chart** you can export your chart as an image (PNG) with or without the legend.
   - Freeze the current view by mousing over the data point you are interested in and pressing shift on your keyboard. You can release the view by pressing the esc on your keyboard.
   - **Zoom**: you can zoom in by selecting the area in the chart that you want. You can zoom out again by clicking anywhere within the chart area.

[1] https://en.wikipedia.org/wiki/Logarithmic_scale
[2] https://en.wikipedia.org/wiki/Simple_linear_regression

- For your numeric fields, BigML also computes the Pearson[3] and Spearman[4] coefficients so you get a measure of the linear correlation between your chosen fields.
- **Color selector**: you can select a field to color your chart points (text, items and date-time fields are not supported).

Figure 6.3: Dynamic scatterplot options

2. Finally, the **data inspector** on the right hand side shows all your dataset fields, distributions, and values (see Figure 6.4). When you mouse over a data point in the chart, you will see the values for each field highlighted in the corresponding histograms. You can freeze this data point view by pressing the shift key. To release it, just press esc on your keyboard.

[3] https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient
[4] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

Figure 6.4: Dynamic scatterplot data inspector

6.2 Handling Missing Values in the Scatterplot Chart

The Dynamic Scatterplot chart handles differently the missing values of numeric fields depending on whether the numeric field with missing values is selected in the plot axes or in the color scale selector.

If the values of any of the fields in the fields selectors are missing for some instances in the current sample, a label will inform about the number of missing values found in the sample.

1. If the values of any of the fields in the axes (or both) are missing for all the instances in the sample, a message will appear informing about it and no points will be drawn in the plot. You can try to request another sample to get some new data.

   If only part of the instances in the current sample have missing values for any of the fields in the axes, the points representing those instances will not be drawn.

Figure 6.5: Dynamic scatterplot missing values in axes

2. If the entire sample has missing values in the field chosen as color selector, you will see a border but no filling for the circles that represent your sample data. If only some instances have missings, the circle will be filled with the regular background color and also a line texture.

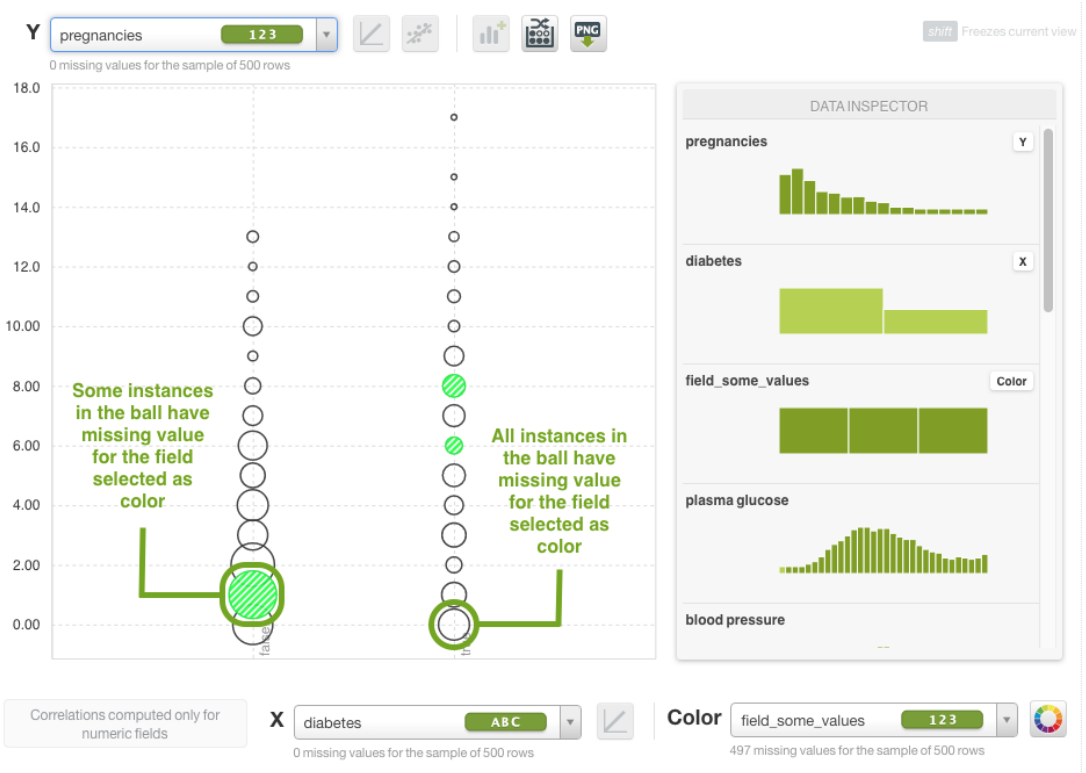

Figure 6.6: Dynamic scatterplot missing values in color selector

CHAPTER 7

Sampling and Filtering Datasets

BigML allows you to easily sample, and filtering your dataset, two key tasks in any Machine Learning process. The following subsections explain how to perform both tasks in the BigML Dashboard.

You can find these options from the CONFIGURE DATASET menu as shown in Figure 7.1.

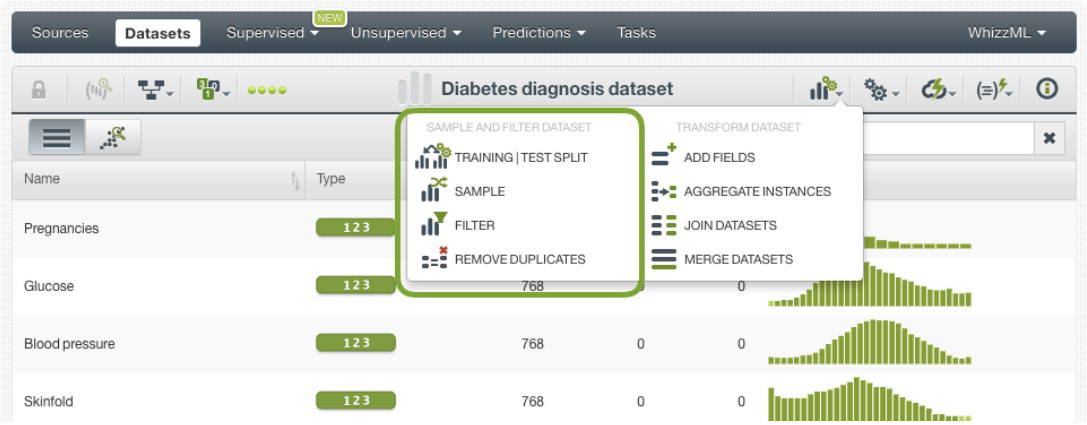

Figure 7.1: Sampling and filtering options

BigML also offers 1-click splitting options, an easy way to get two subsets from a single dataset, one for training and other for testing supervised models. You can find these options in the 1-click menu as shown in Figure 7.2.

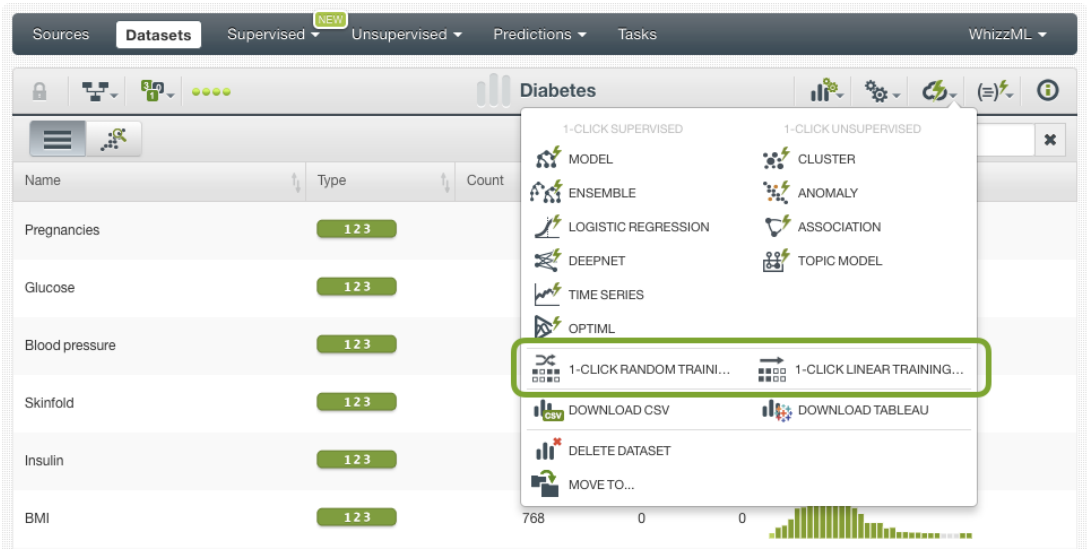

Figure 7.2: 1-click options for splitting datasets

7.1 Splitting Datasets in Training/Test

For most Machine Learning tasks, it is essential to evaluate your model to get an estimate of its performance. To do so, you need to split your dataset into two different subsets. Use the bigger subset (called training data) to build your model, and later test the performance of your model against the smaller subset (called test data). It is important to note that the test data is data that the algorithm never saw when building the model. By doing this, you will able to measure the real performance of your model when a new case appears. The complete evaluation process is explained in the Classification and Regression with the BigML Dashboard[1] [3].

This section explains how to split your dataset into two different subsets. BigML offers you two ways: using the corresponding 1-click action that lets you get a selected training dataset containing 80% of the data, and another one containing the remaining 20% for testing; or configuring those ratios by using the configuration option. The below sections cover both options.

7.1.1 Splitting Datasets with 1-Click

This option divides your dataset in two subsets, 80% of your data to train the model and the 20% left to test it. BigML provides two different splitting options: a **random** and a **linear** option. If you are training a **classification** **or** **regression** model, you usually use the random split which randomly takes instances for each subset. If you are training a **time series** model, you need to use the linear split which assumes that the instances are chronologically oredered in the dataset and takes the first 80% for training and the last 20% for testing.

From the **dataset view**, select the most suitable option for your use case 1-CLICK RANDOM TRAINING|TEST or 1-CLICK LINEAR TRAINING|TEST. (See Figure 7.3.)

[1] https://static.bigml.com/pdf/BigML_Classification_and_Regression.pdf

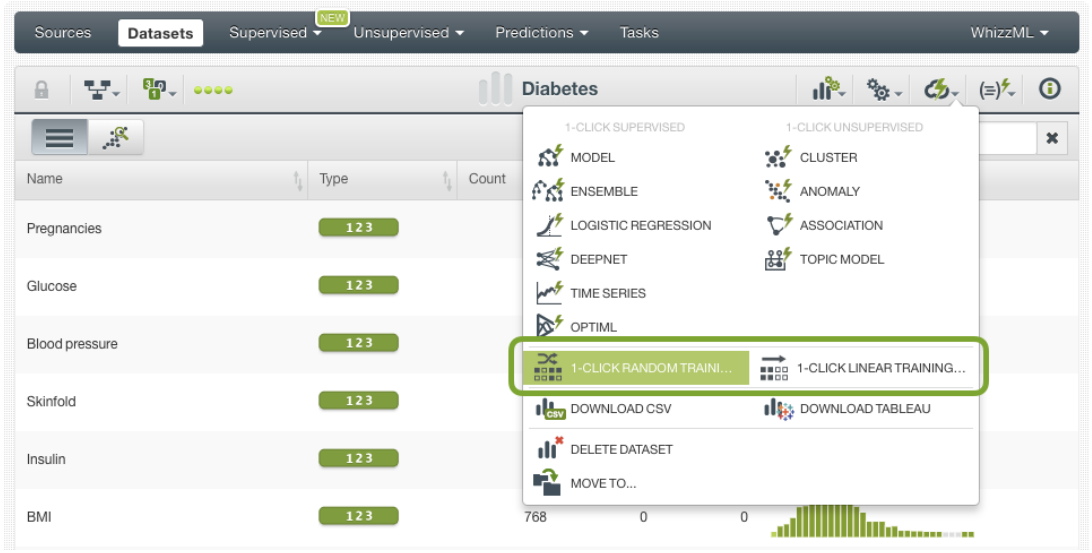

Figure 7.3: One-click training|test split

When BigML processes this request, both subsets are automatically created and displayed in your Dashboard. You can see the two separate subsets in the **dataset list view**. (See Figure 7.4.)

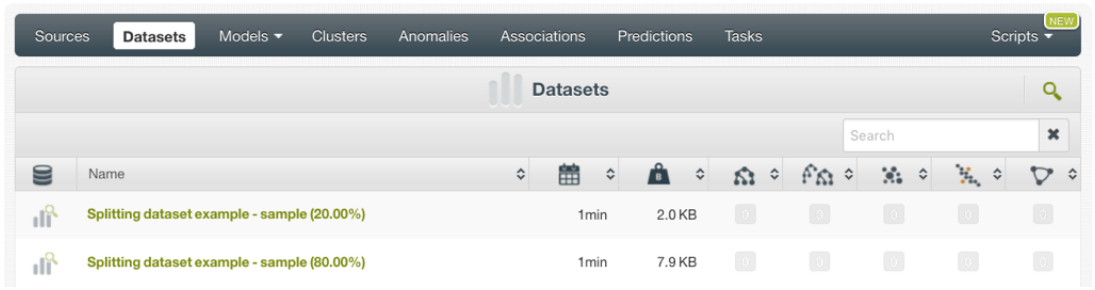

Figure 7.4: Training|test subsets in the dataset list view

7.1.2 Configuring Training/Test Split Options

BigML lets you select the percentage of your data for training and for testing.

From the **dataset view**, click on the **configure option menu** and select TRAINING AND TEST SET SPLIT. (See Figure 7.5.)

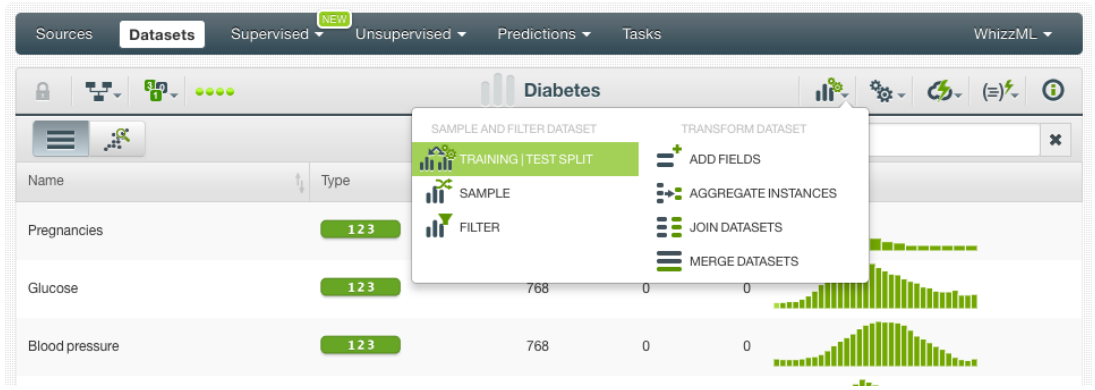

Figure 7.5: Access to configure the training|test split

You can configure the **percentage** for training and testing using the slider shown in Figure 7.6. In this

example we choose 80% and 20% respectively. You can also input any string to the **seed** parameter to generate deterministic samples and get repeatable results. If you use the same seed for a given dataset, each time you make the training/test split the training and test subsets will contain the same instances. Otherwise, the instances for each subset will be randomly selected and you will get different training and test sets each time you make a split for a given dataset. BigML also provides an option so you can make the split **linear** instead of random, i.e., the subsets will be created taking into account the order of the instances in your dataset (the first subset of instances for training and the last subset for testing). This option needs to be activated in case you want to train and test a time series model since the instances are chronologically distributed. You can also **name** your training and test sets differently.

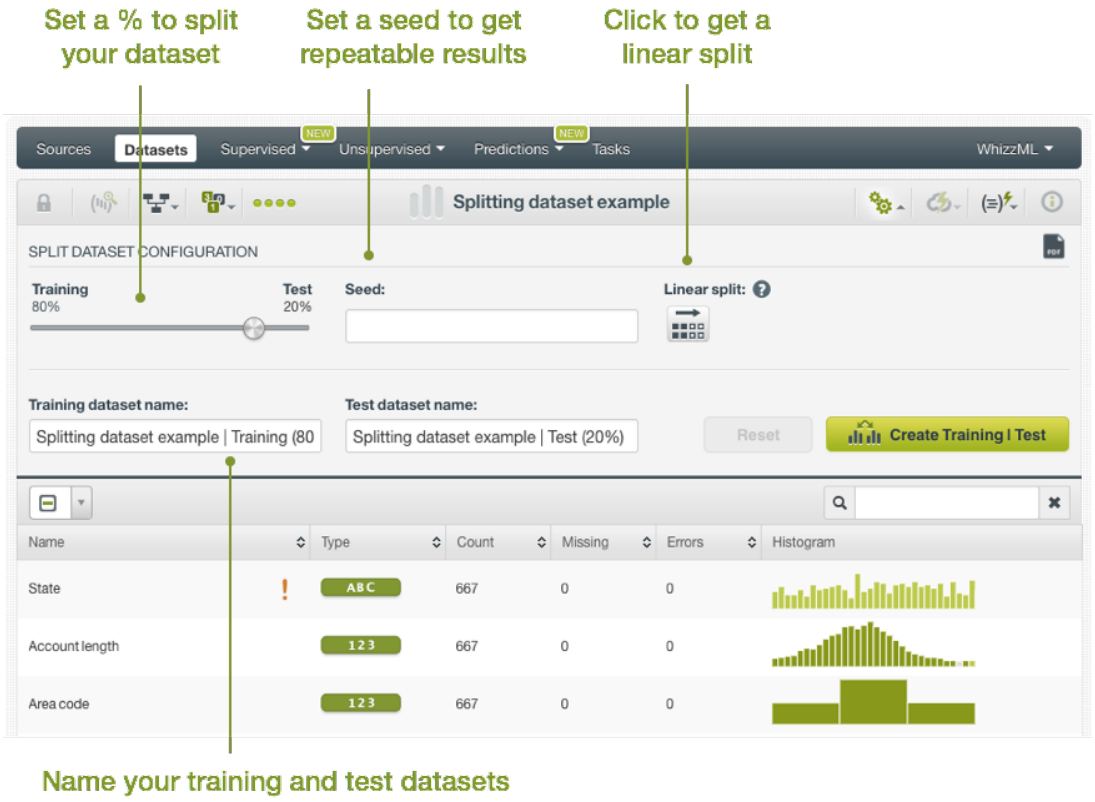

Figure 7.6: Training|test splits configuration panel

7.2 Sampling Datasets

Most of the time, you do not need all the data to generate your models. If you have very large datasets, sampling may be a good way of getting results and iterating faster. Sampling your data is straight forward with BigML. Simply open the **configure option menu** and select SAMPLE DATASET. (See Figure 7.7.)

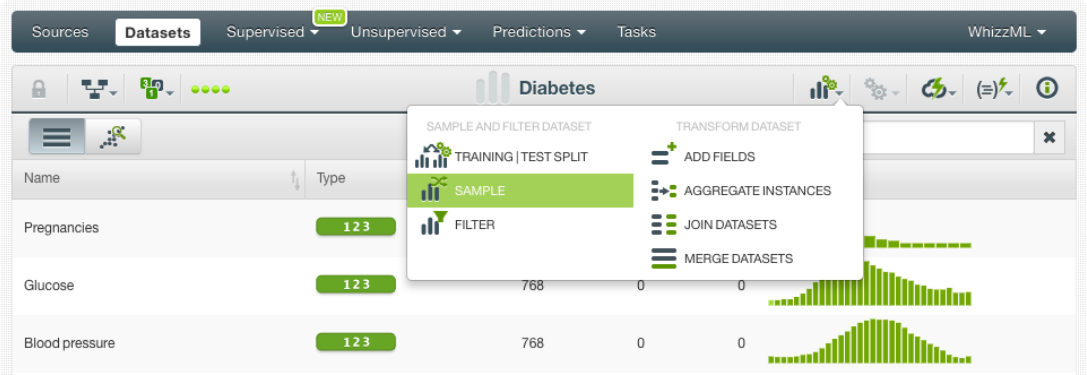

Figure 7.7: Access to sample your dataset

Find in the sections below a detailed explanation of all the configuration options that BigML offers to sample your dataset.

7.2.1 Sampling

You can easily configure the **sampling rate** by moving the slider in the **configuration panel for sampling**, or by typing the percentage in the tiny input box, both highlighted in Figure 7.8. The rate is the proportion of instances to include in your sample. After that, you can also name your sampled dataset differently.

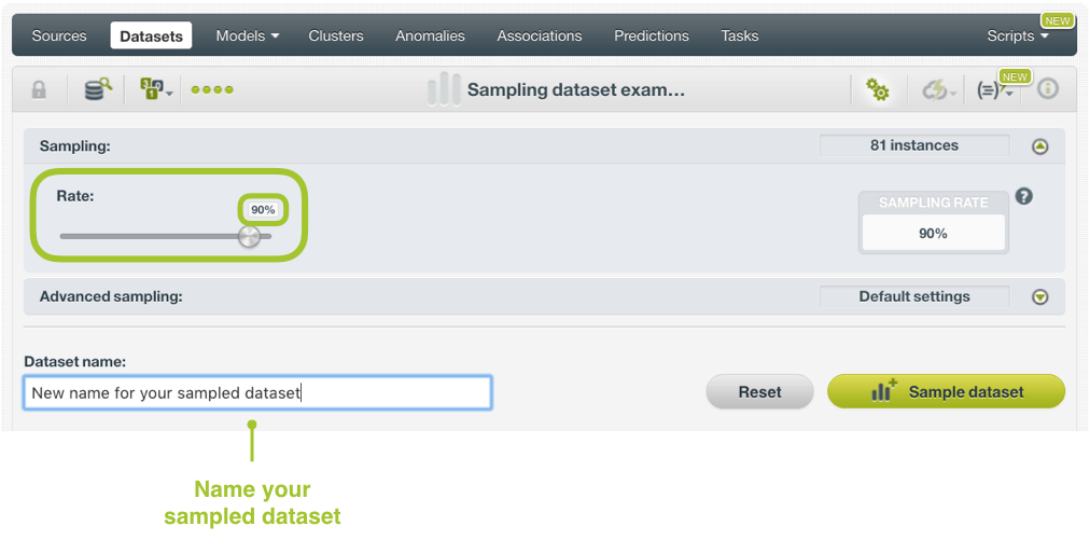

Figure 7.8: Configuration panel for sampling

7.2.2 Advanced Sampling

If you prefer to sample differently your dataset, configure the following advanced options in the **configuration panel for advanced sampling**: (See Figure 7.9)

7.2.2.1 Range

Specify a subset of instances, when the instances are ordered, from which to sample. For example, choose a **range** from instances 100 to 200. The specified rate will be applied over the subset configured. This option may be useful when you have temporal data, and you want to train your model with historical data and test it with the most recent one to check if it can predict based on time.

7.2.2.2 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, except when the rate is 100% and do not use repetition. If you choose **deterministic sampling**, the random-number generator will always use the same **seed**, thus producing repeatable results. This lets you work with identical samples from the same dataset.

7.2.2.3 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.

7.2.2.4 Out of Bag

Create a sample containing only **out-of-bag** instances for the currently defined rate, the final total number of instances for your sample will be one minus the rate configured for your sample (when replacement is false). This can be useful for splitting a dataset into training and testing subsets. It is only selectable when a sample rate is less than 100%.

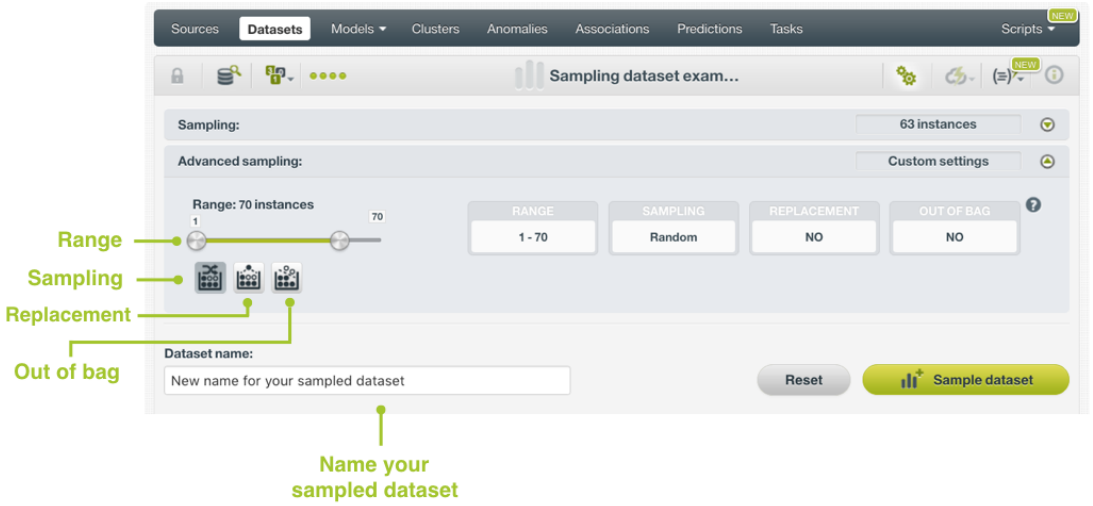

Figure 7.9: Configuration panel for advanced sampling

7.3 Filtering Datasets

BigML lets you transform your original dataset in several ways. This section covers how to create a new dataset by filtering instances. You may use the **pre-defined operations** criteria available in the filters selector, or you may customize your filter using Flatline **formulas**.

Access this option by clicking the **configure option menu** and selecting FILTER DATASET. (See Figure 7.10.)

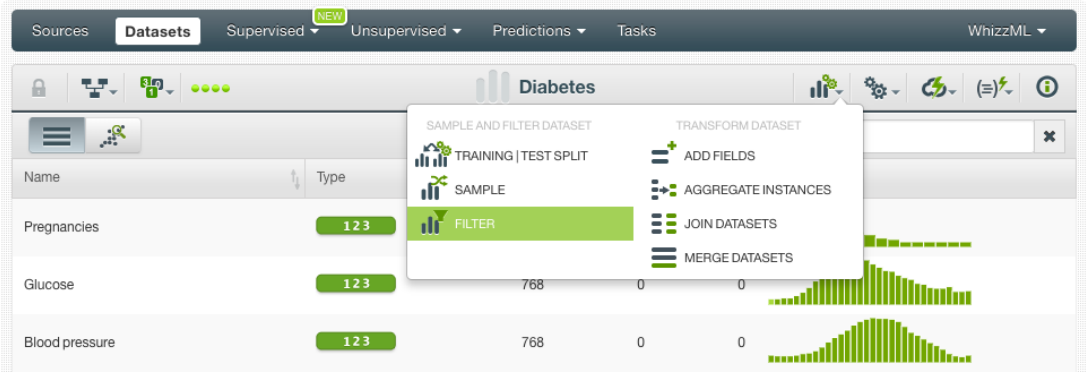

Figure 7.10: Access to filter your dataset

This leads you to the **configuration panel for filtering** (Figure 7.11) where you can choose the field you want to filter, and decide which operation you wish to apply. Add up to ten different filtering conditions manually by clicking the Add condition button shown in this panel. You can add as many filtering conditions as you want by using flatline formulas. Please read Subsection 7.3.7 or the Flatline manual[2] for your reference, which is also available from the **help panel**. The help panel may be useful when you want to quickly find the definition of each operation. Finally, you can name your filtered dataset differently before you click the Create dataset button.

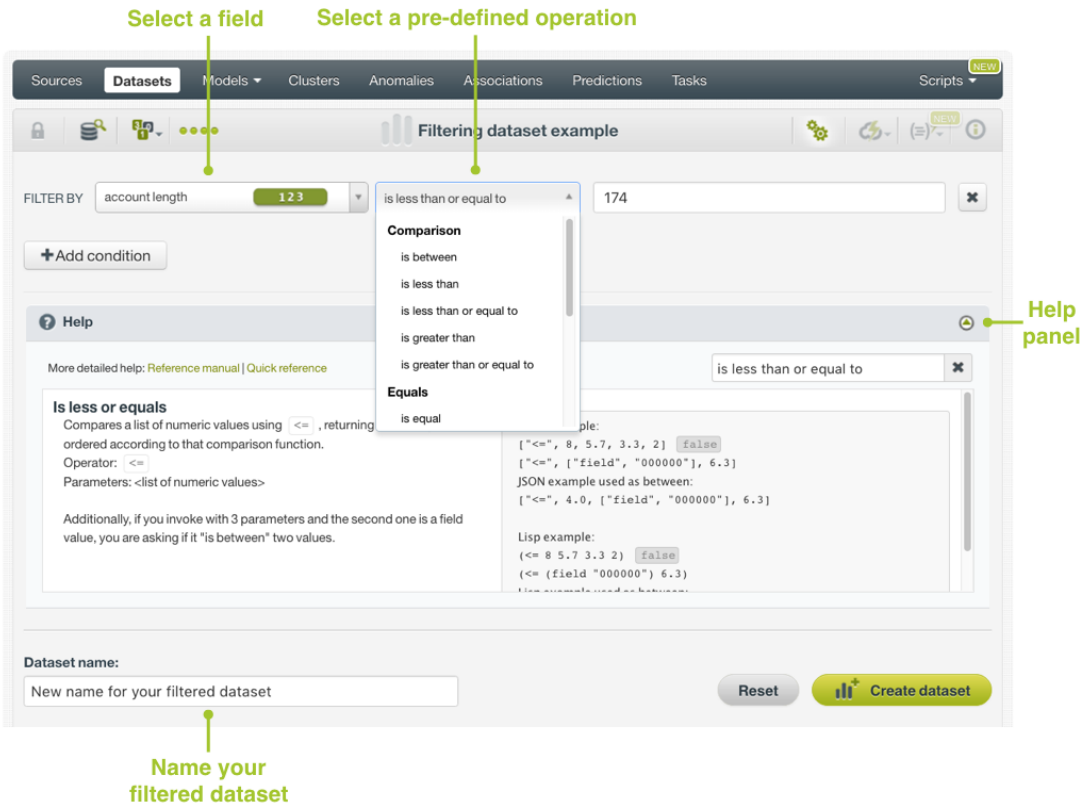

Figure 7.11: Configuration panel for filtering

You may want to filter different instances from your dataset depending on your goals. For instance, you might need to find the instances that have missing values in a certain field, or instances that contain values higher than X for another field, etc. The following subsections cover which operations are available

[2] http://flatline.readthedocs.org/

per field type.

7.3.1 Filtering By Numeric Fields

To filter your numeric fields, choose between the following **operations**:

- **Comparison** (See Figure 7.12.)
  - **Is between**: includes instances containing values within the specified range
  - **Is less than**: includes instances containing values below the specified level
  - **Is less than or equal to**: includes instances containing values equal or below the specified level
  - **Is greater than**: includes instances containing values above the specified level
  - **Is greater than or equal to**: includes instances containing values equal or above the specified level

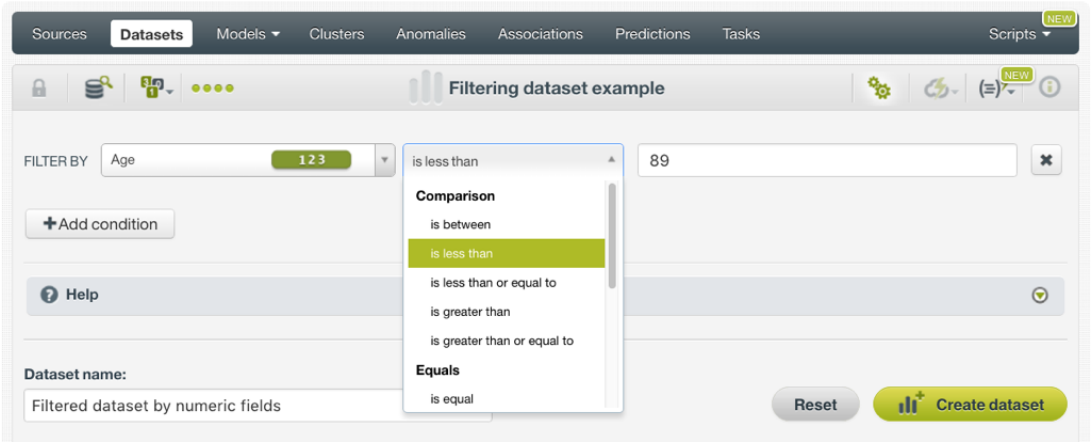

Figure 7.12: Filtering a dataset by a numeric field with **comparison** operations

- **Equals** (See Figure 7.13.)
  - **Is equal**: includes instances containing the specified value/values
  - **Is not equal**: excludes instances containing the specified value/values

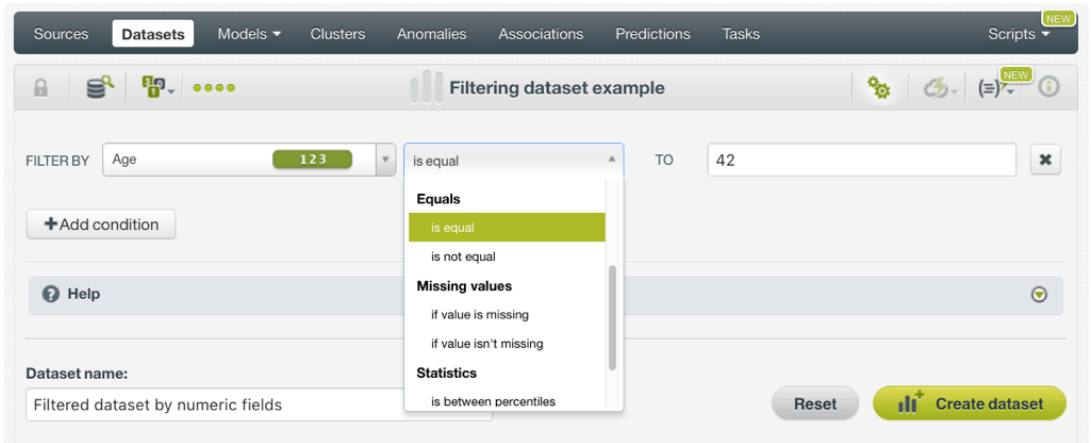

Figure 7.13: Filtering a dataset by a numeric field with **equals** operations

- **Missing values** (See Figure 7.14.)
  - **If value is missing**: includes instances containing missing values for the selected field

  - **If value isn't missing**: excludes instances containing missing values for the selected field

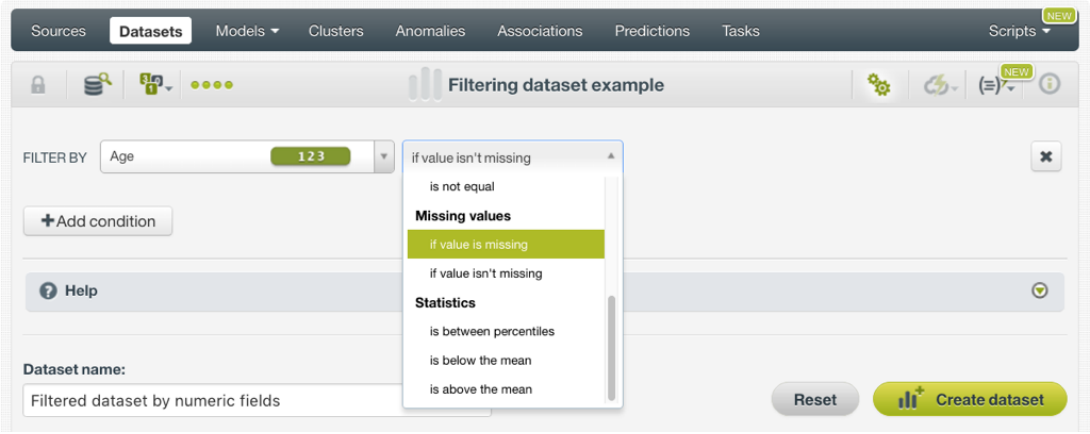

Figure 7.14: Filtering a dataset by a numeric field with **missing values** operations

- **Statistics** (See Figure 7.15.)
  - **Is between percentiles**: includes instances within the specified percentiles E.g., a percentile between 0 and 0.3 includes the first 30% of the instances.
  - **Is below the mean**: includes instances below the mean of the selected field
  - **Is above the mean**: includes instances above the mean of the selected field

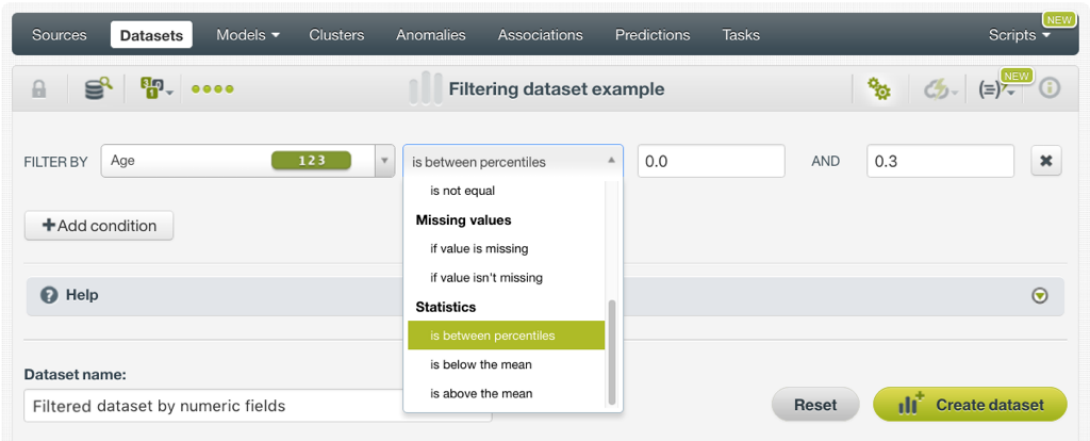

Figure 7.15: Filtering a dataset by a numeric field with **statistics** operations

7.3.2 Filtering By Categorical Fields

BigML lets you decide which operation you want to apply to filter your field. The following operations are applicable to categorical fields and all field types supported by BigML. (See Figure 7.16 and Figure 7.17.)

- **Specific values**
  - **Equals**: includes instances containing the specified value/values
  - **Does not equal**: excludes instances containing the specified value/values

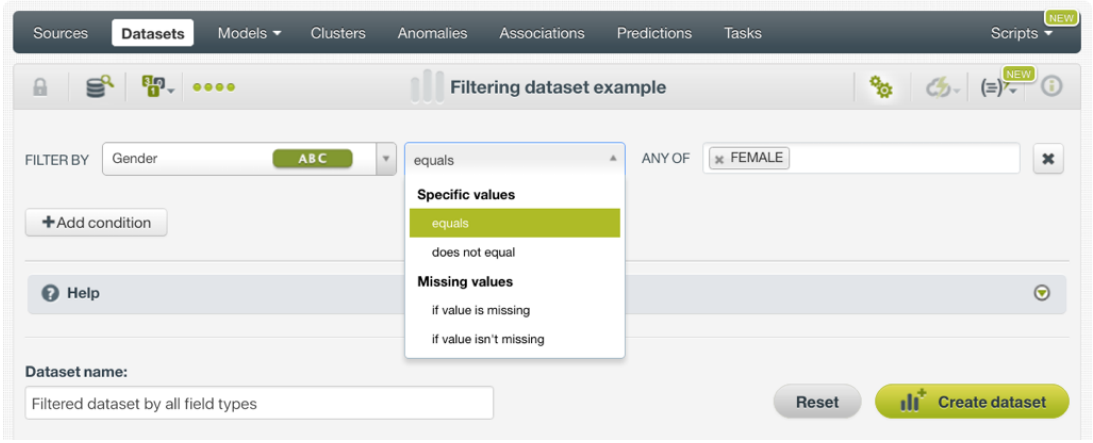

Figure 7.16: Filtering a dataset by all field types with **specific values** operations

- **Missing values**
  - **If value is missing**: includes instances containing missing values for the selected field
  - **If value isn't missing**: excludes instances containing missing values for the selected field

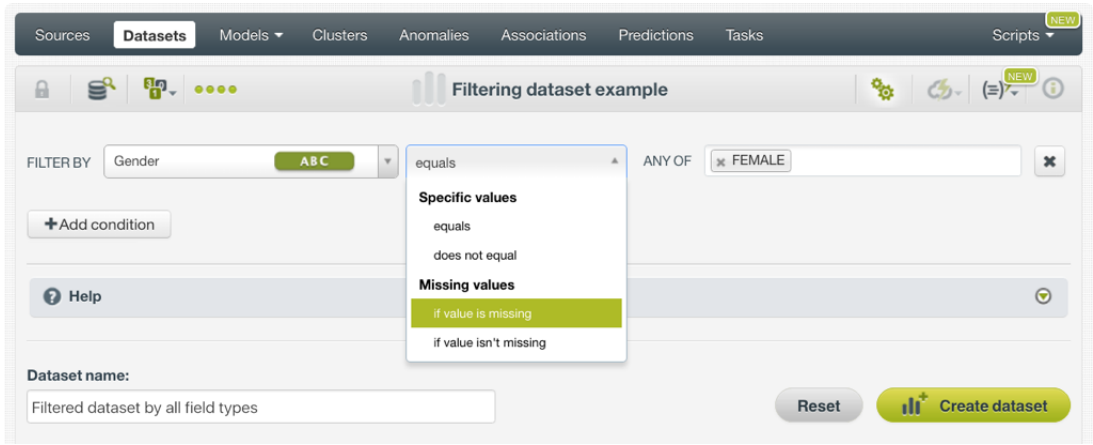

Figure 7.17: Filtering a dataset by all field types with **missing values** operations

7.3.3 Filtering By Text Fields

To filter your text fields you can choose between the following **operations**:

- **Equals** (See Figure 7.18.)
  - **Is equal**: includes instances containing the specified value/values
  - **Is not equal**: excludes instances containing the specified value/values

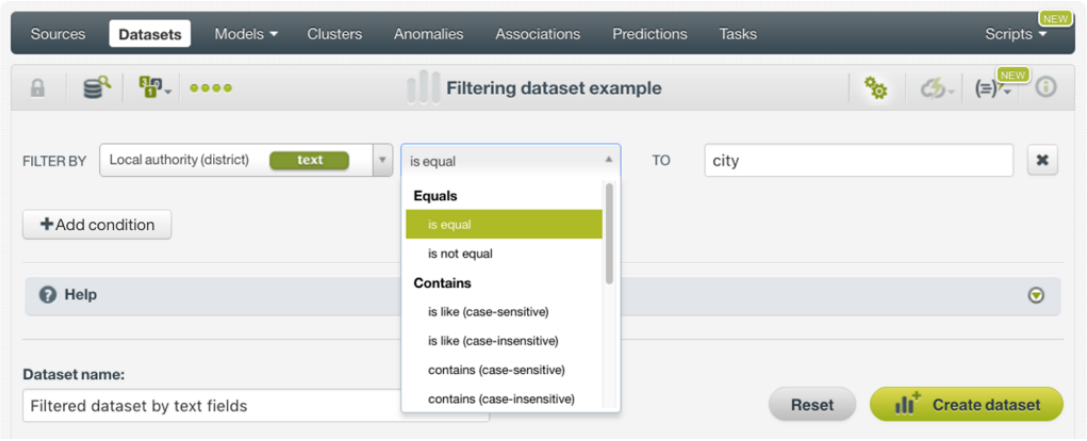

Figure 7.18: Filtering a dataset by a text field with **equals** operations

- **Contains** (See Figure 7.19.)
  - **Is like (case-sensitive)**: matches words containing at least part of the letters specified, taking into account lower and upper cases, e.g., “great” will also match a text containing the word “great” or “greatness,” but not “Great” or “Greatness”
  - **Is like (case-insensitive)**: matches words containing at least part of the letters specified, not taking into account lower and upper cases, e.g., “great” will also match a text containing the word “great”, “greatness”, “Great” or “Greatness”
  - **Contains (case-sensitive)**: matches texts containing the exact words specified, taking into account lower and upper cases, e.g., “great” will match a text containing the word “great”, but not “Great”
  - **Contains (case-insensitive)**: matches texts containing the exact words specified, not taking into account lower and upper cases, e.g., “great” will match a text containing the word “great” or “Great”

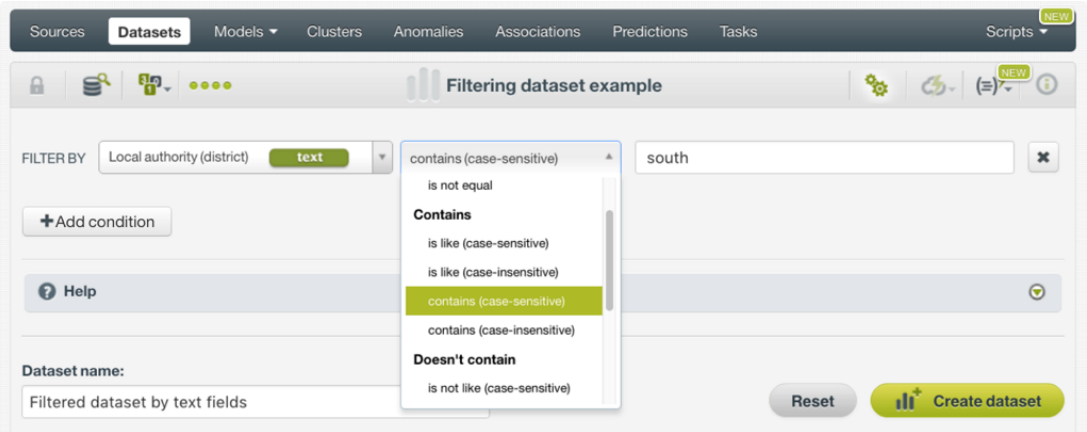

Figure 7.19: Filtering a dataset by a text field with **contains** operations

- **Doesn’t contain** (See Figure 7.20.)
  - **Is not like (case-sensitive)**: excludes instances with words containing at least part of the letters specified, taking into account lower and upper cases, e.g., “great” will also exclude a text containing the word “great” or “greatness”, but not “Great” or “Greatness”
  - **Is like (case-insensitive)**: excludes instances with words containing at least part of the letters specified, not taking into account lower and upper cases, e.g., “great” will also exclude a text containing the word “great”, “greatness”, “Great” or “Greatness”

  - **Not contains (case-sensitive)**: excludes texts containing the exact words specified, taking into account lower and upper cases, e.g., “great” will exclude a text containing the word “great”, but not “Great”
  - **Not contains (case-insensitive)**: excludes texts containing the exact words specified, not taking into account lower and upper cases, e.g., “great” will exclude a text containing the word “great” or “Great”

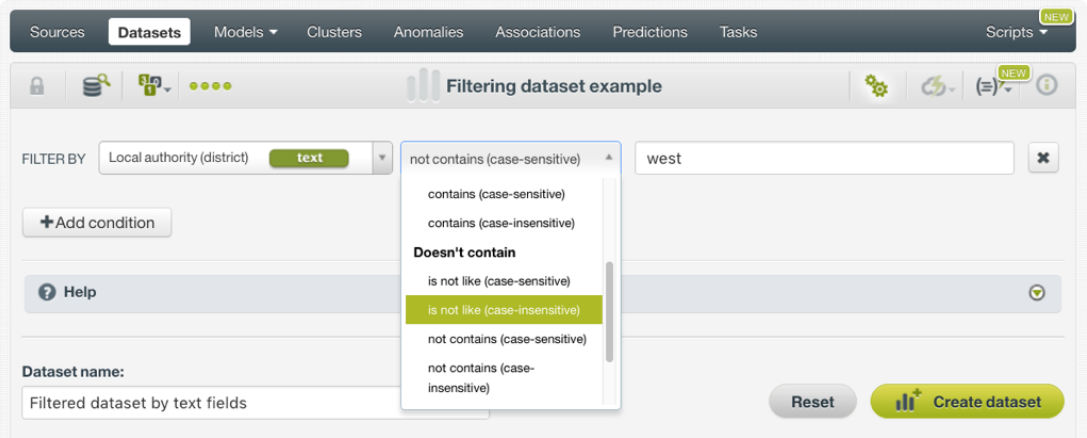

Figure 7.20: Filtering a dataset by a text field with **doesn’t contain** operations

- **Missing values** (See Figure 7.21.)
  - **If value is missing**: includes instances containing missing values for the selected field
  - **If value isn’t missing**: excludes instances containing missing values for the selected field

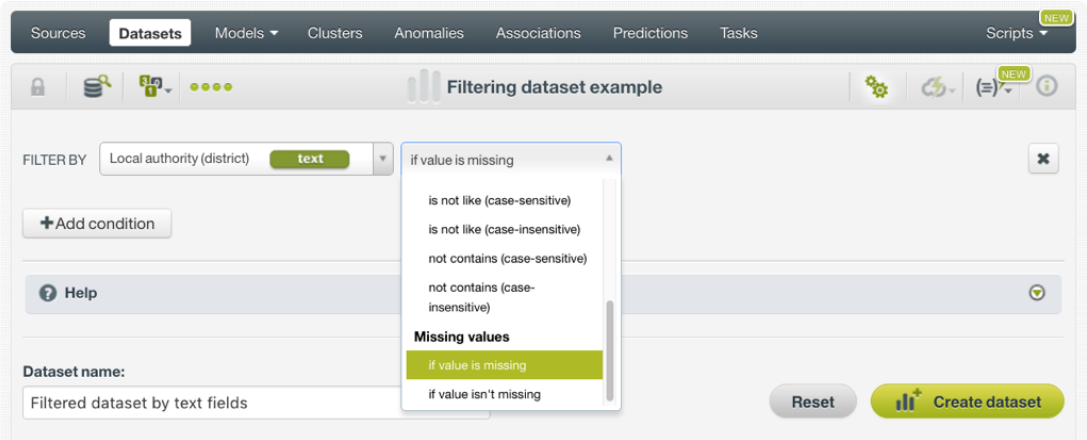

Figure 7.21: Filtering a dataset by a text field with **missing values** operations

7.3.4 Filtering By Items Fields

BigML offers the below **operations** for you to filter your dataset by items fields:

- **Equals** (See Figure 7.22.)
  - **Is equal**: includes instances containing the specified value/values
  - **Is not equal**: excludes instances containing the specified value/values

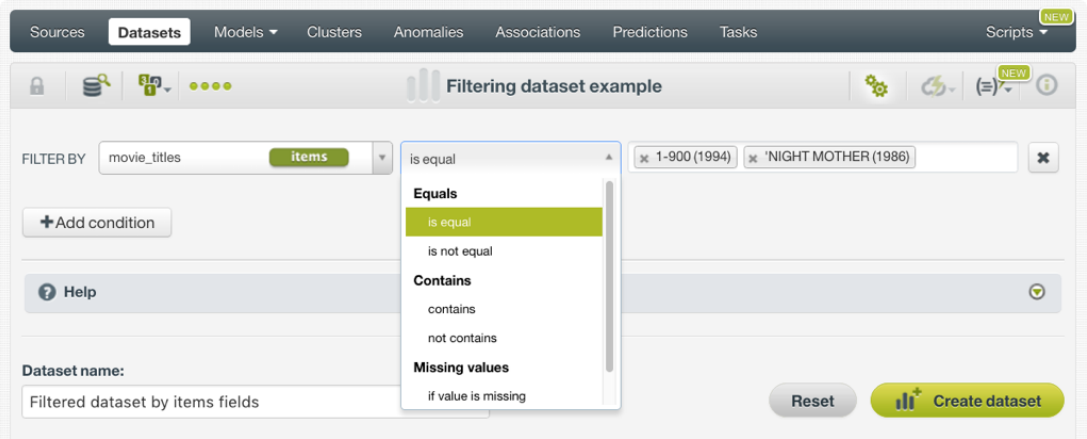

Figure 7.22: Filtering a dataset by an items field with **equals** operations

- **Contains** (See Figure 7.23.)
  - **Contains (case-insensitive)**: matches texts containing the exact words specified, not taking into account lower and upper cases, e.g., “great” will match a text containing the word “great” or “Great”
  - **Not contains (case-sensitive)**: excludes texts containing the exact words specified, taking into account lower and upper cases, e.g., “great” will exclude a text containing the word “great”, but not “Great”

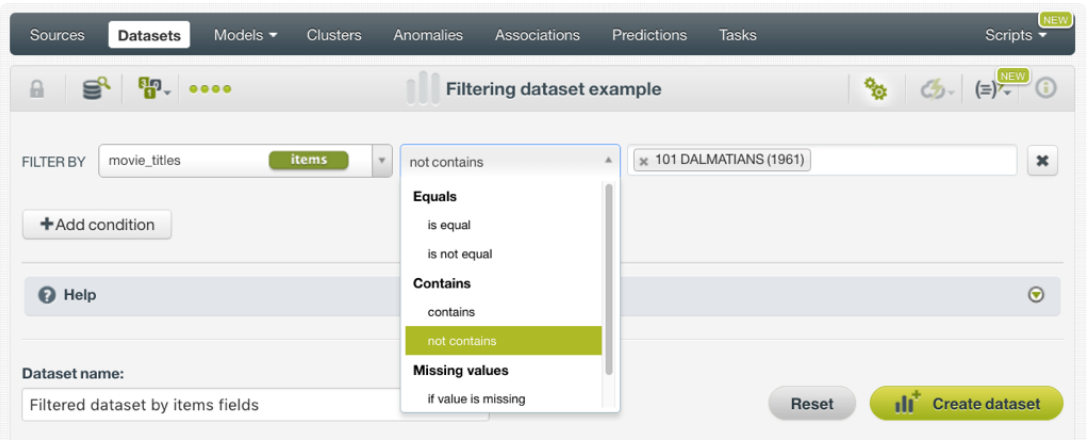

Figure 7.23: Filtering a dataset by an items field with **contains** operations

- **Missing values**: (See Figure 7.24)
  - **If value is missing**: includes instances containing missing values for the selected field
  - **If value isn’t missing**: excludes instances containing missing values for the selected field

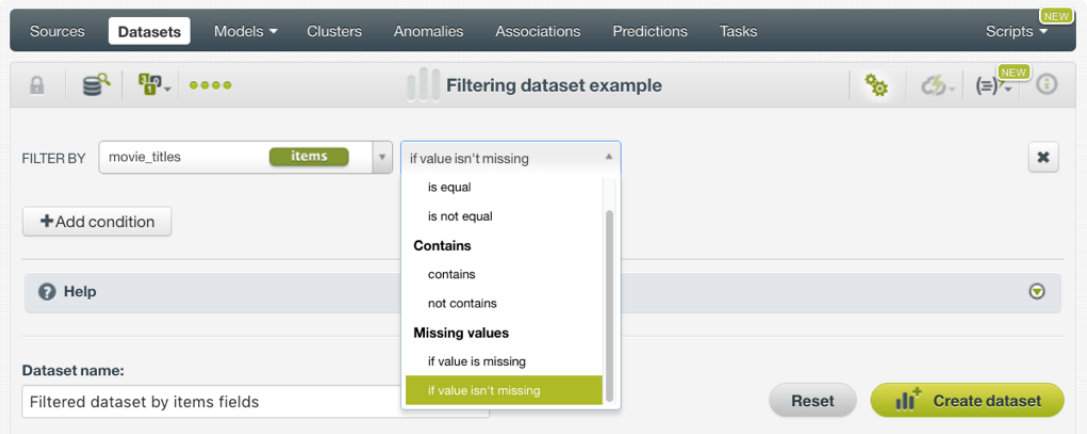

Figure 7.24: Filtering a dataset by an items field with **missing values** operations

7.3.5 Filtering By Date-Time Fields

To filter your dataset by date-time fields, BigML offers the **same operations as for the numeric fields** (see Subsection 7.3.1). The only difference is that you have to select the values in a calendar. (See Figure 7.25.)

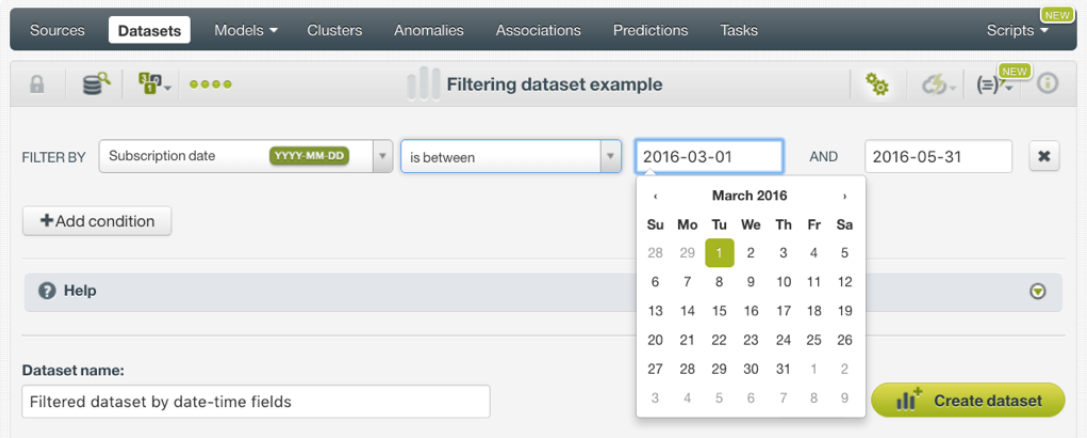

Figure 7.25: Filtering a dataset by a date-time field with **comparison** operations

7.3.6 Filtering Using Flatline Formulas

You can also filter your dataset by **writing a flatline formula**, either with Lisp syntax[3] or with JSON syntax[4]. BigML lets you easily type the formulas directly (see Figure 7.26 and Figure 7.27), or use the Flatline editor to create and validate your flatline formula. (See Subsection 7.3.7 for more details.)

[3] https://en.wikipedia.org/wiki/Lisp_(programming_language)
[4] https://en.wikipedia.org/wiki/JSON

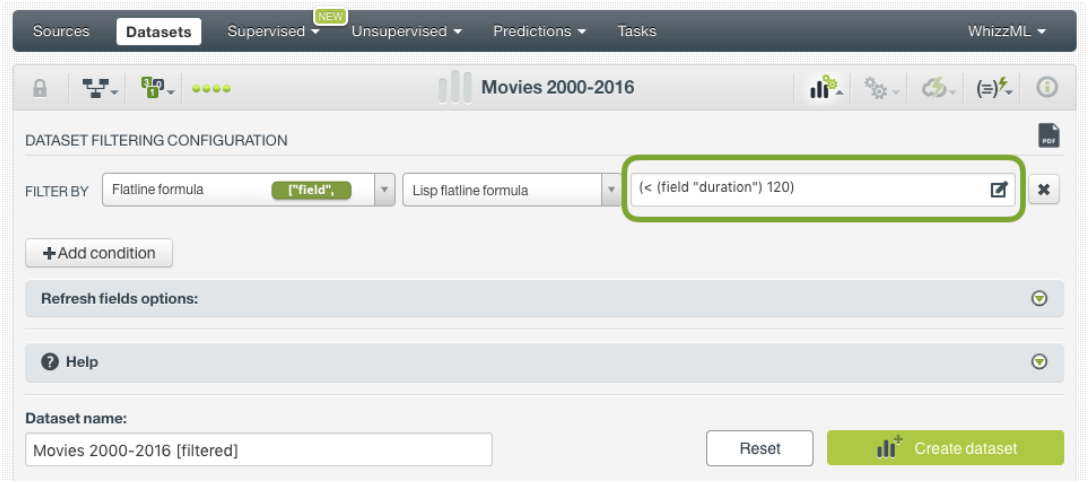

Figure 7.26: Filtering a dataset by a **Lisp flatline formula**

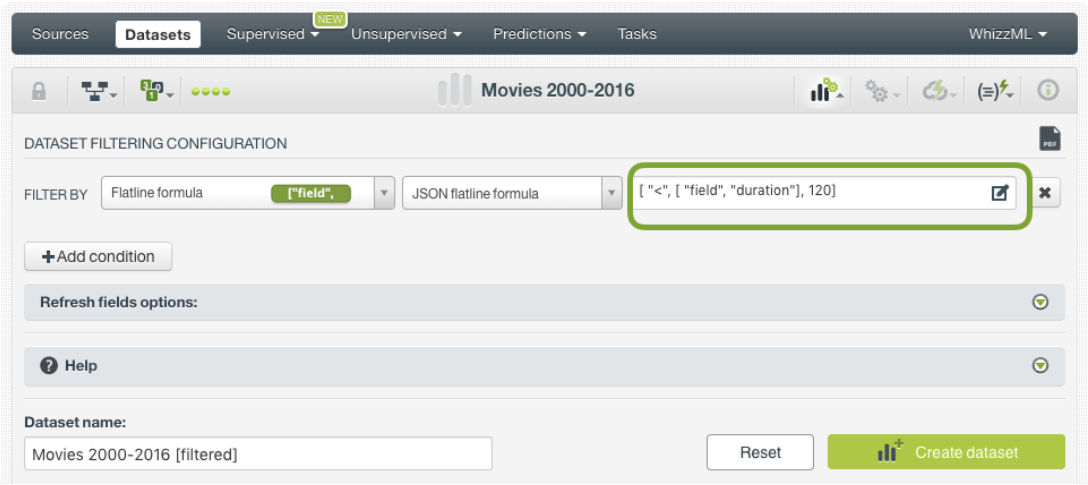

Figure 7.27: Filtering a dataset by a **JSON flatline formula**

7.3.7 Filtering Using the Flatline Editor

The **flatline language** can greatly help you filter your dataset in infinite ways to get higher quality predictors. Follow the steps below to edit your Lisp formula or your JSON formula. Select the desired syntax. The following example is a Lisp flatline formula:

1. Click the highlighted icon in Figure 7.28 to add a formula using the Flatline editor:

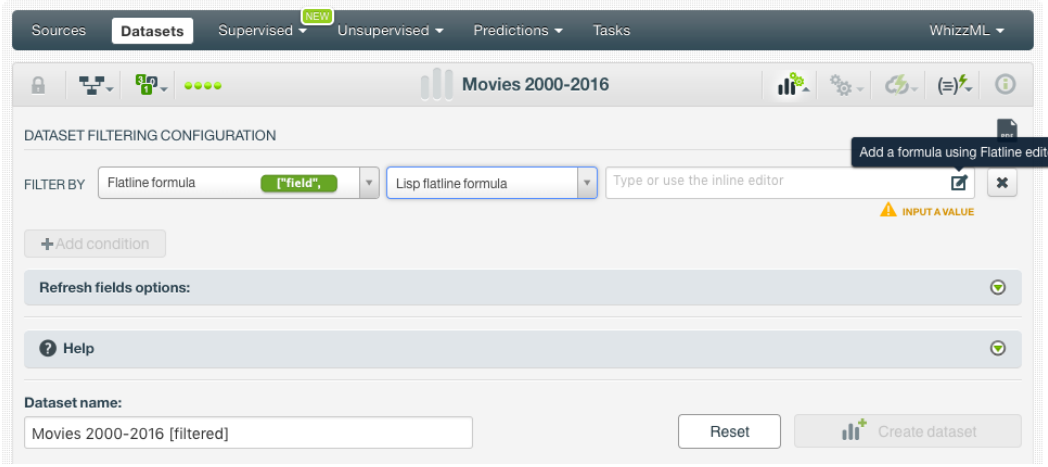

Figure 7.28: Access the Flatline editor

2. Next the Lisp expression is selected. Type your expression in the **editor panel** Figure 7.29. You can also use the help panel any time if you have doubts about the operation to compute (Figure 7.30).

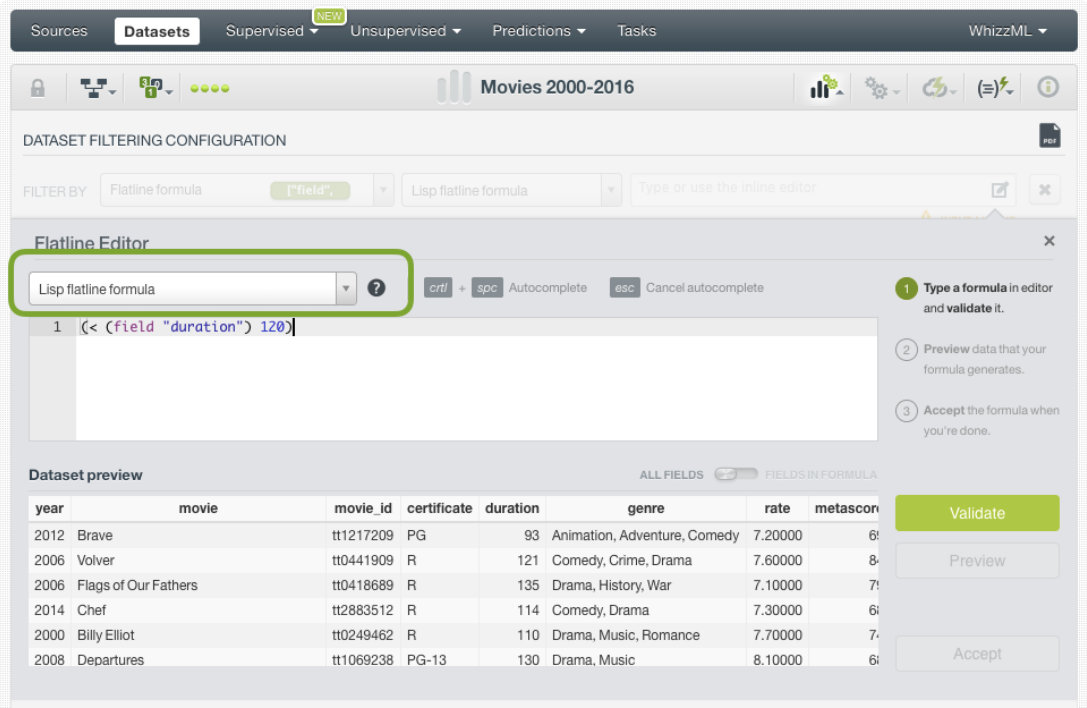

Figure 7.29: Edit your Lisp expression

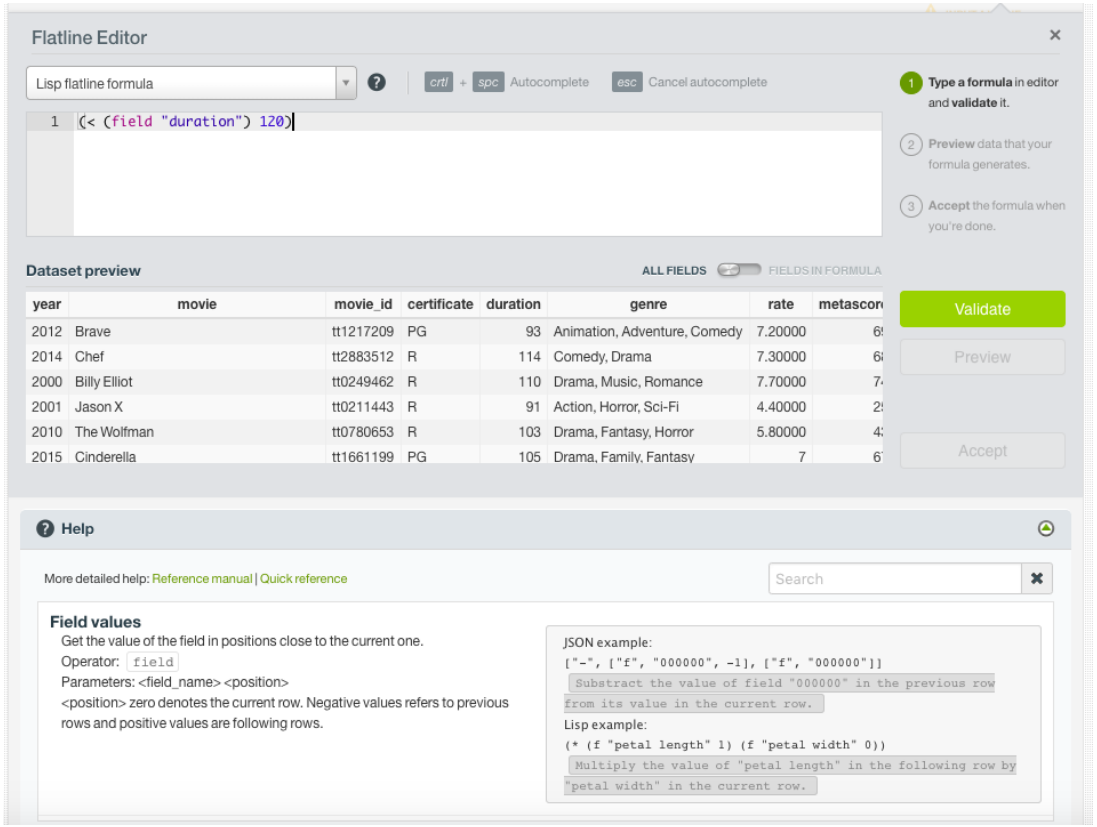

Figure 7.30: Help panel to learn more about the operations you can use to filter your dataset

3. Click the Validate button in Figure 7.30 to know whether the operation is valid. If it is valid (Figure 7.31), proceed with the following steps, but if it is not valid, BigML will display a message (Figure 7.32) letting you know the error.

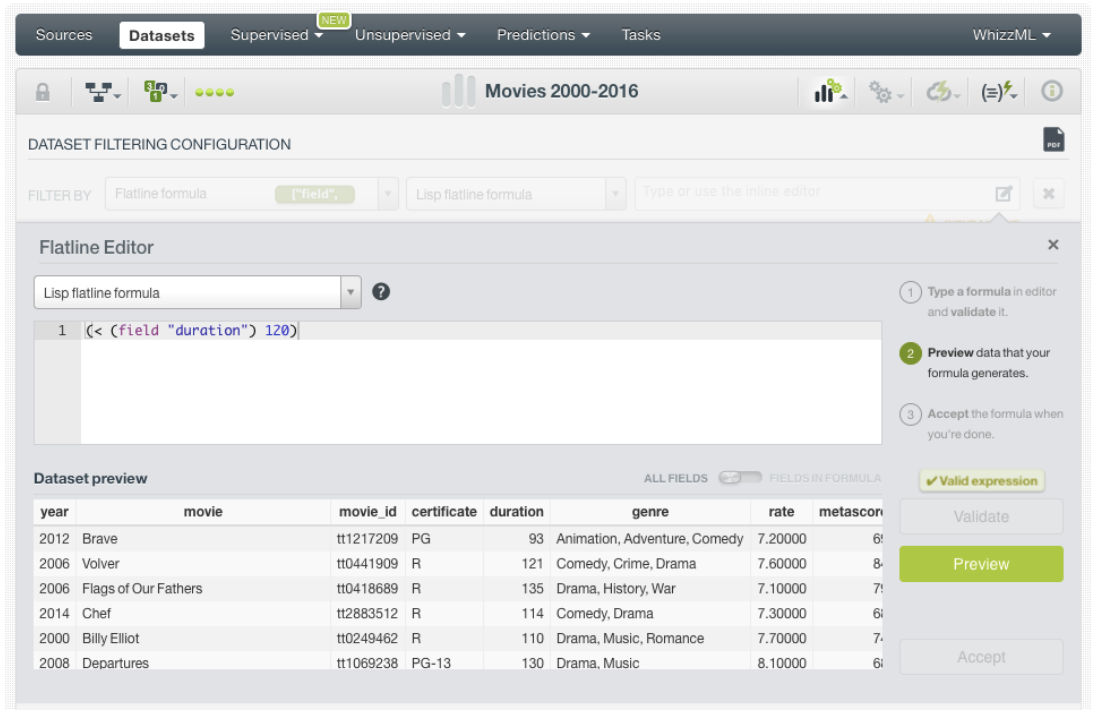

Figure 7.31: Example of a valid expression

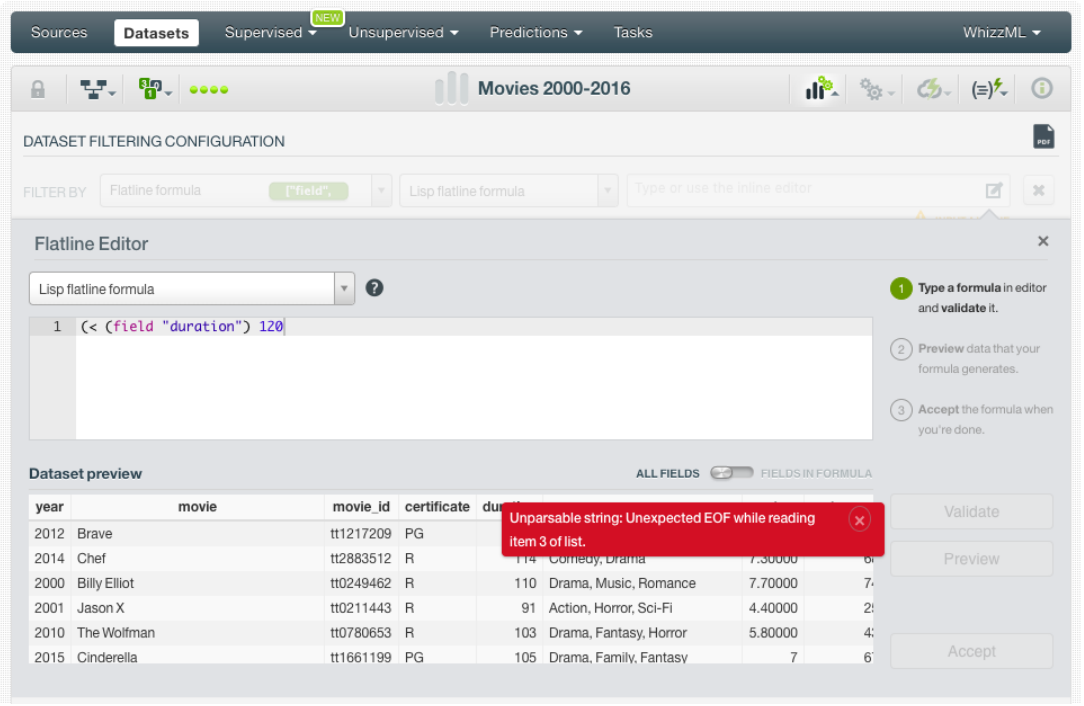

Figure 7.32: Example of an invalid expression

If you want to convert the Lisp expression into a JSON expression simply **switch** to JSON expression (Figure 7.33) so you do not lose it.

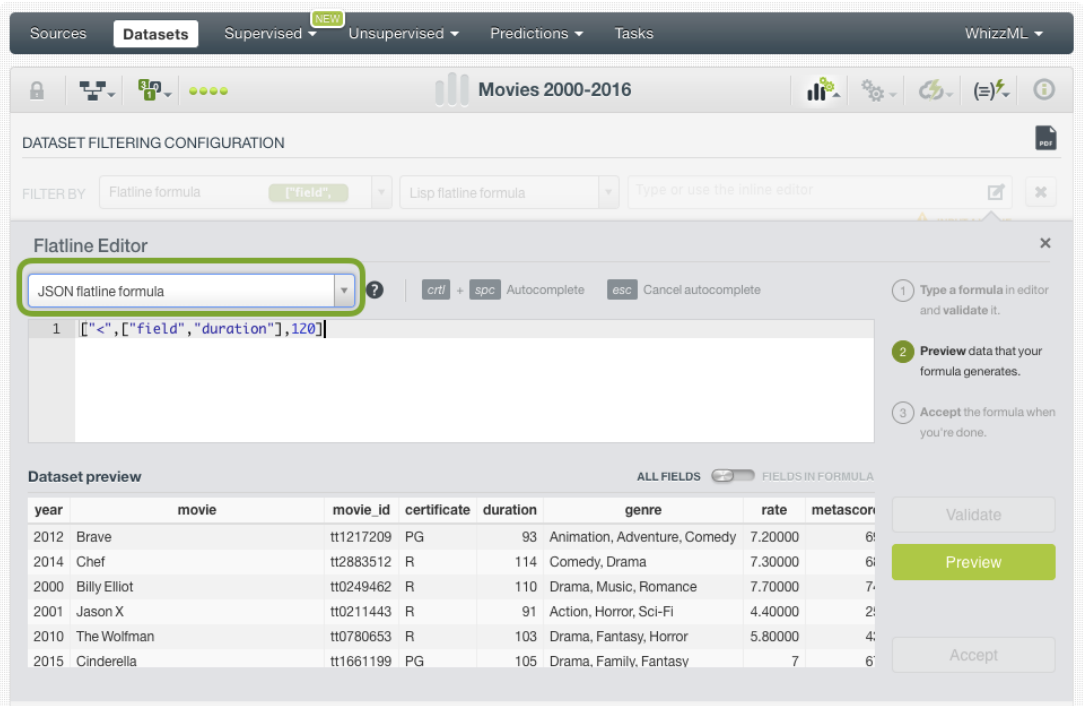

Figure 7.33: JSON expression

4. After validating your expression, click the Preview button (in Figure 7.31) to see the expression result shown in Figure 7.34. You can observe that, by default, only the fields involved in the formula are shown in the preview.

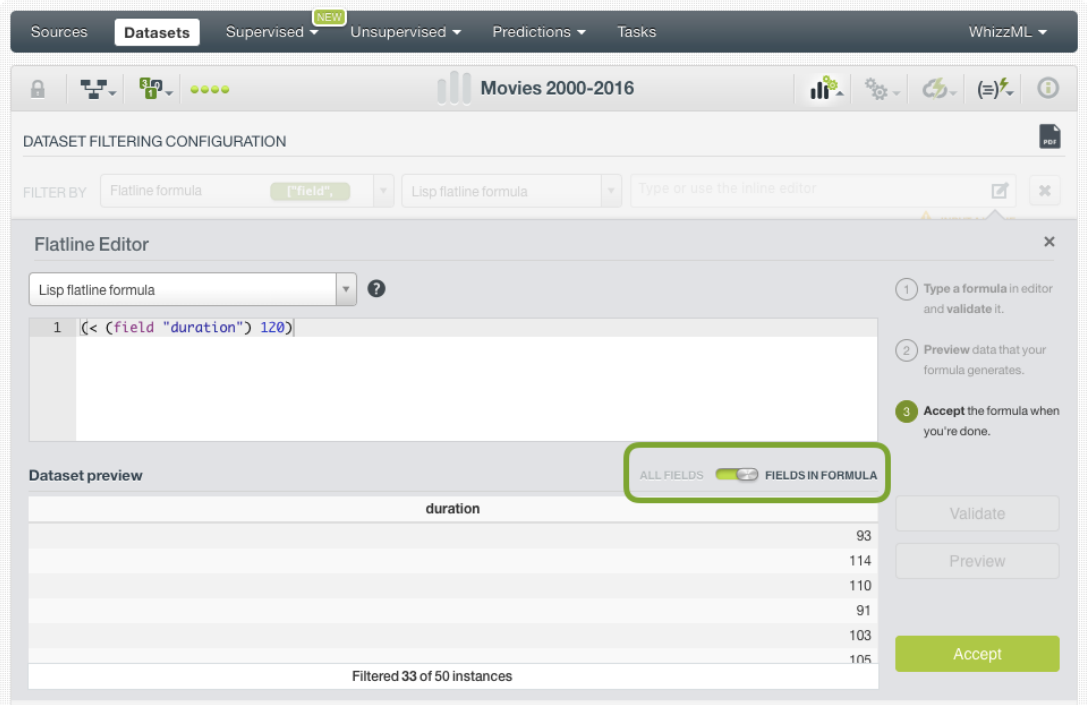

Figure 7.34: Preview of the expression result (only fields in formula)

You can change this, and display all the fields in the dataset by clicking in the switcher shown in Figure 7.35

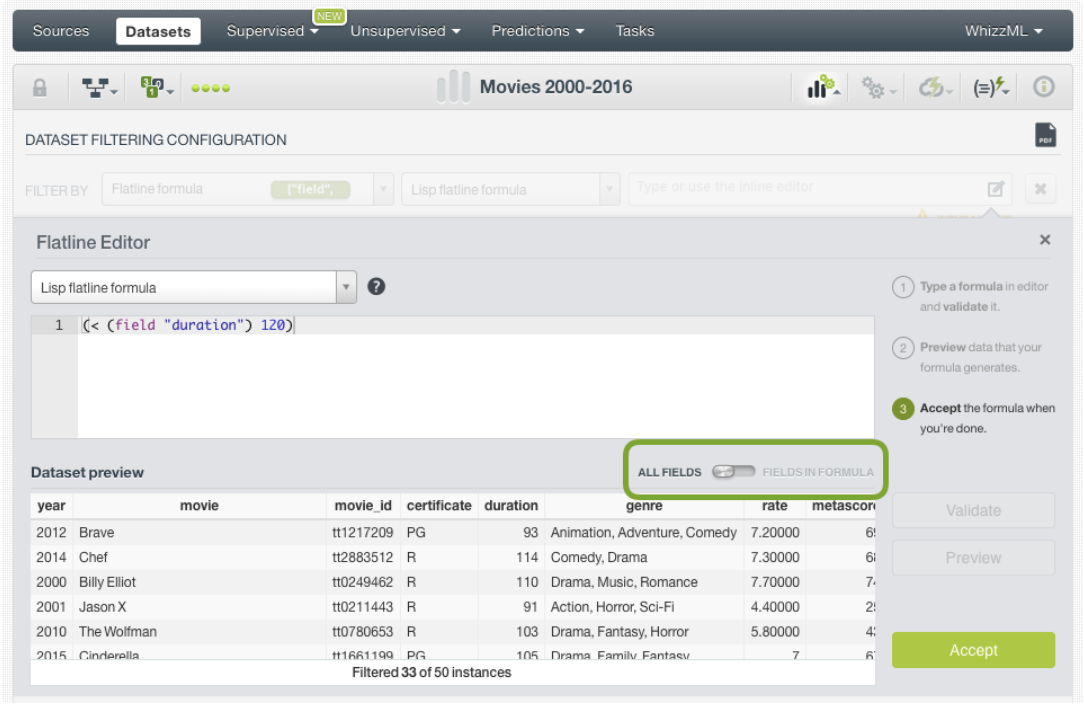

Figure 7.35: Preview of the expression result (all fields)

5. Then click the Accept button. (See Figure 7.34.) BigML will display the new Lisp expression in the same field where you can directly type the expression before opening the Flatline editor. (See Figure 7.36.) Press the Create dataset button to create the filtered dataset.

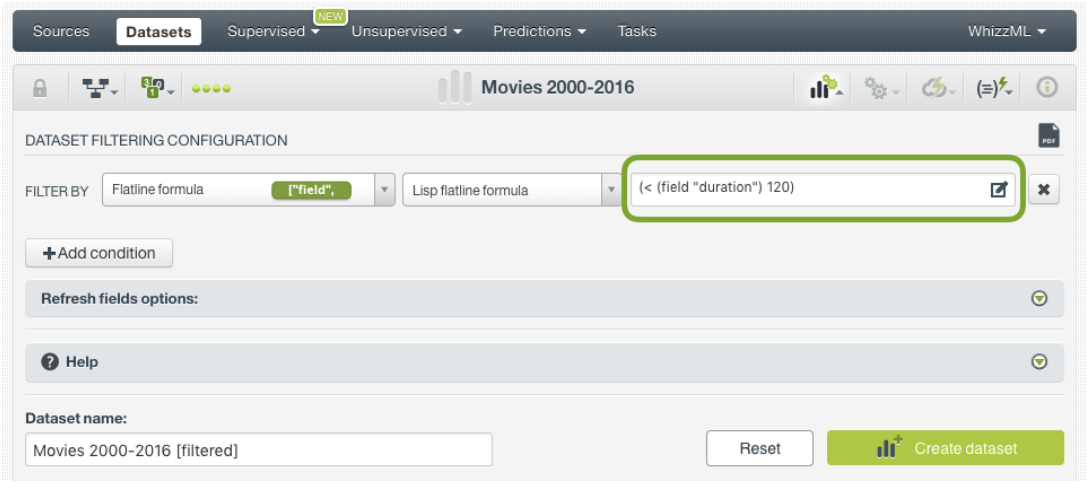

Figure 7.36: Lisp formula edited in the Flatline editor

Please visit the Flatline manual[5] for a full discussion about how to use the Flatline editor.

7.3.8 View and Reuse Filters

When you create the filtered dataset, you will be able to **view the filters** applied by clicking the option shown in Figure 7.37.

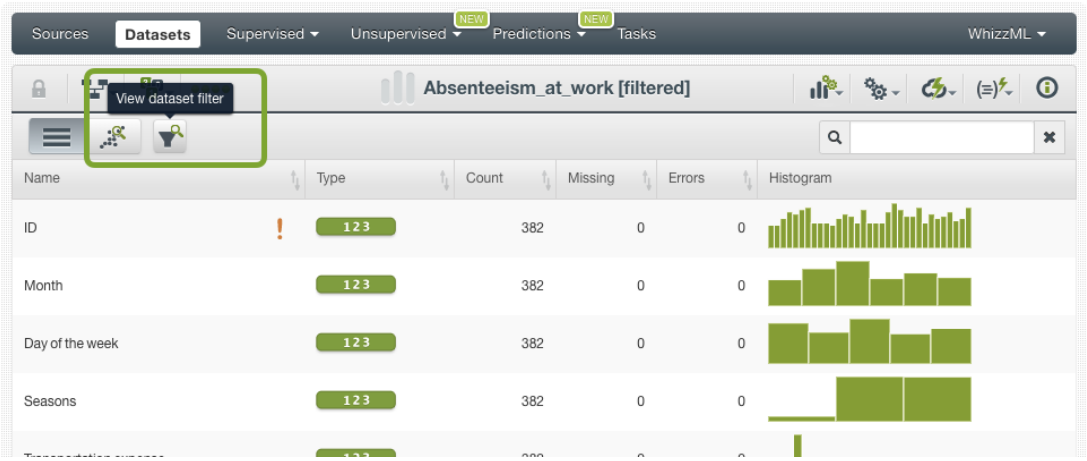

Figure 7.37: View the filters applied to a dataset

This option will display a window with the **Flatline formula** used to filter the dataset (see Figure 7.38). You can copy or download the formula (in Lisp and JSON formats) to apply this filter to another dataset.

[5] http://flatline.readthedocs.org/

Figure 7.38: Copy and download filters

This section described how to transform your data by filtering a dataset. The next section (Section 7.4) explains a different way of filtering your original dataset, by removing the duplicated instances.

7.4 Remove Duplicates

Duplicated instances in a dataset can be problematic when training Machine Learning models. For example, if you make a random split of your dataset and you take one subset for training and other for testing, it is likely that these duplicated instances appear in both subsets, which will give you an unrealistically good performance of your model. By removing the duplicated instances, you ensure each dataset has unique instances (see Figure 7.39).

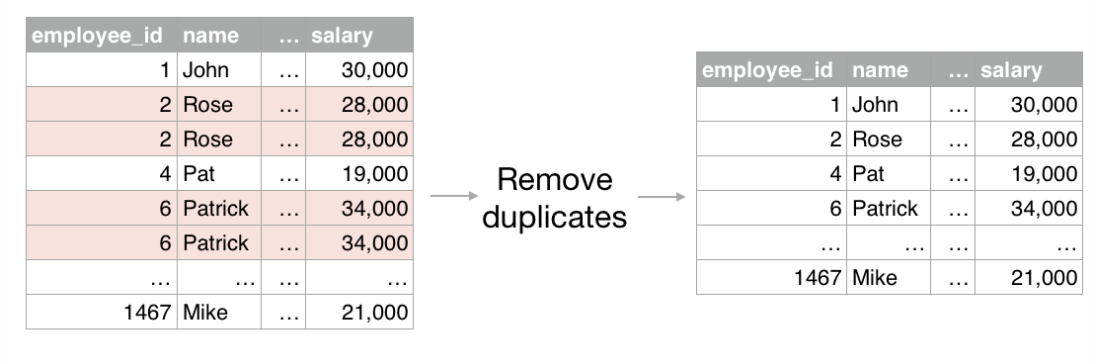

Figure 7.39: Remove duplicated instances example

With BigML you can easily remove the duplicated instances in your datasets following the steps below:

- Find the REMOVE DUPLICATES option in the dataset configuration menu as shown in Figure 7.40.

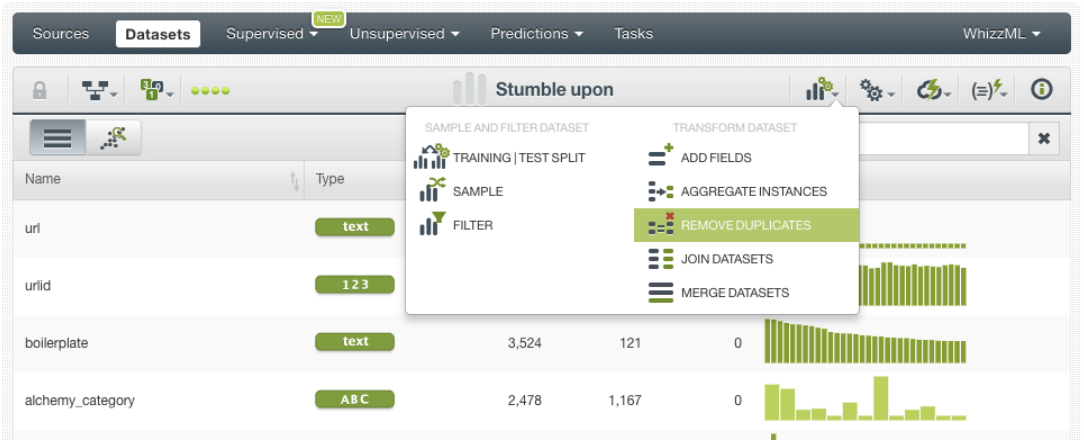

Figure 7.40: Remove duplicates option

- A configuration panel will be displayed where you have only one parameter, the new dataset name. Then click on the “Remove duplicates” button (see Figure 7.41).

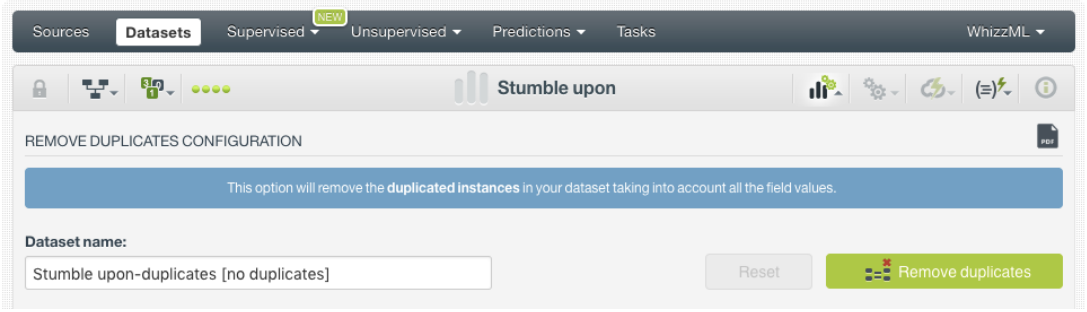

Figure 7.41: Remove duplicates

- When the process has finished, you will see an orange message on top of the dataset indicating how many duplicated instances have been removed (see Figure 7.42). If there were no duplicated instances to remove in your dataset, you will see it in the message too.

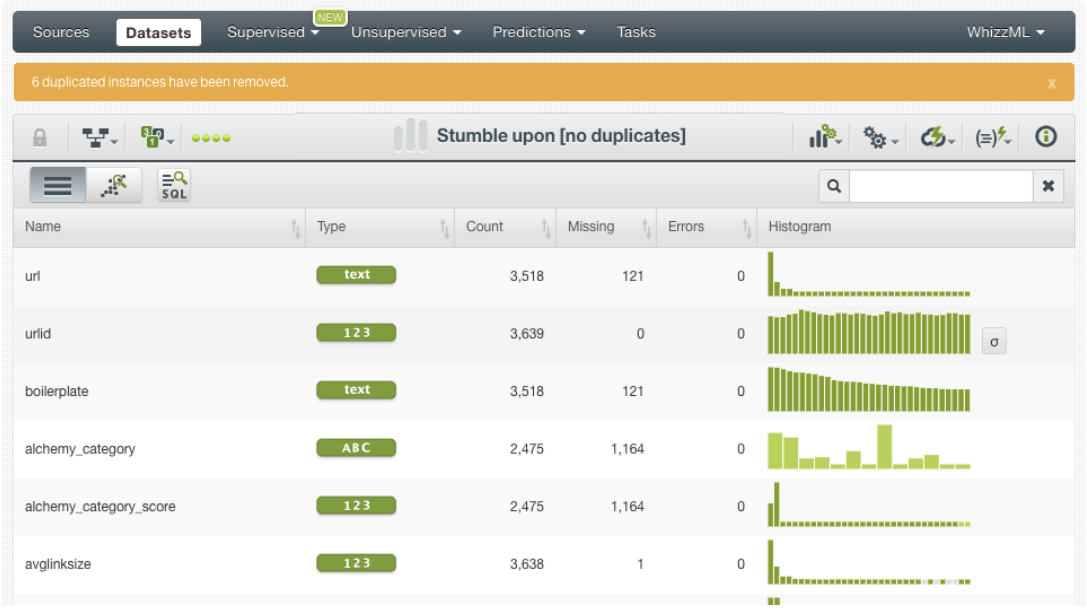

Figure 7.42: Number of duplicates removed

The remove duplicates option in the Dashboard uses an SQL query underneath. Therefore, when the new dataset is created, you can view the SQL query by clicking the option shown in Figure 7.43 below.

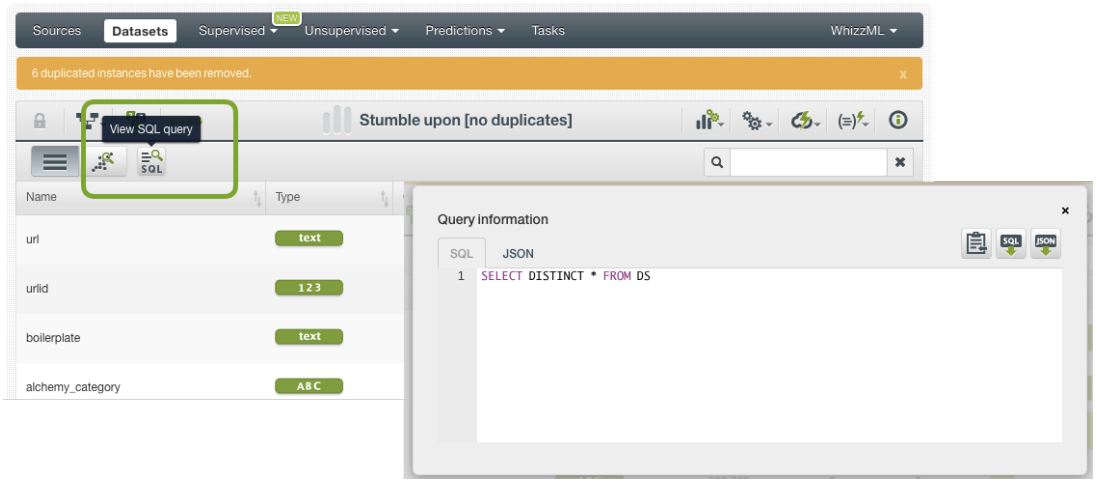

Figure 7.43: View the SQL query of the operation performed

CHAPTER 8

Transforming Datasets

Transforming your data is a key part of any Machine Learning process since the data does not usually have the correct format ready for a Machine Learning model. BigML provides some key functions that allow you to prepare a Machine Learning-ready dataset: **adding fields to a dataset** (feature engineering), **aggregating instances**, **joining, and merging datasets**. The following sections explain each of these transformations in detail.

You can find these options from the CONFIGURE DATASET menu as shown in Figure 8.1.

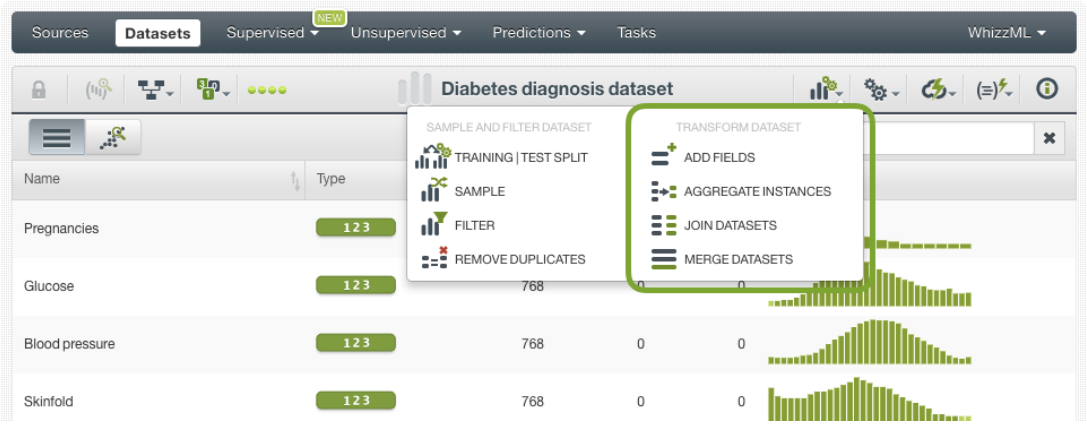

Figure 8.1: Transform dataset

8.1 Adding Fields to a Dataset

If you need to create new fields (i.e., feature engineering), BigML allows you to do it using **common operations** over your existing data, or writing custom operations with Flatline **formulas**. The following subsections describe how to add new fields to your dataset.

To start, access the **configuration option menu** and select ADD FIELDS TO DATASET. (See Figure 8.2.)

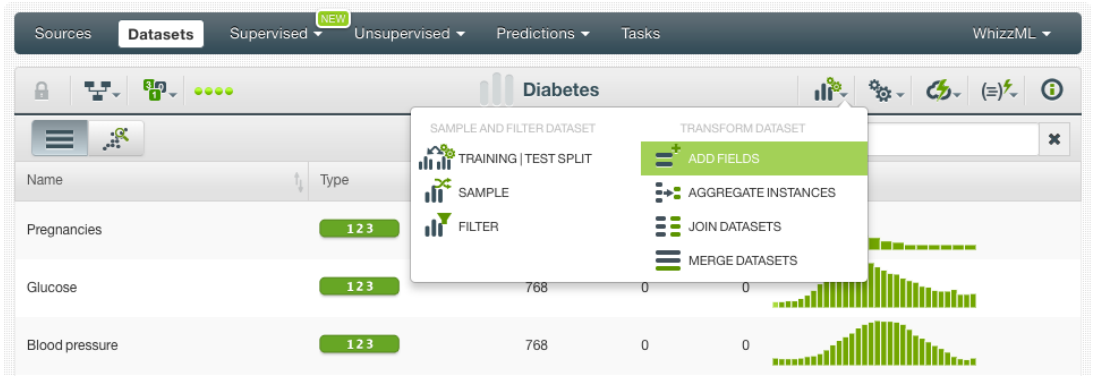

Figure 8.2: Access to add fields to your dataset

This leads you to a **configuration panel for adding fields**, where you can add a name for the new fields, decide which operation you wish to apply, and select the field you will use to generate the new one. (See Figure 8.3.) You can add up to ten new fields manually using BigML Dashboard, as well as writing a custom formula. This is explained in the following subsections.

BigML also provides a **help panel** with an explanation of each operation. This help panel may be useful when you want to quickly find the meaning of each operation. **Note: this is the same help panel as when filtering your dataset.**

Finally, you can also name your extended dataset differently before you click the Create dataset button.

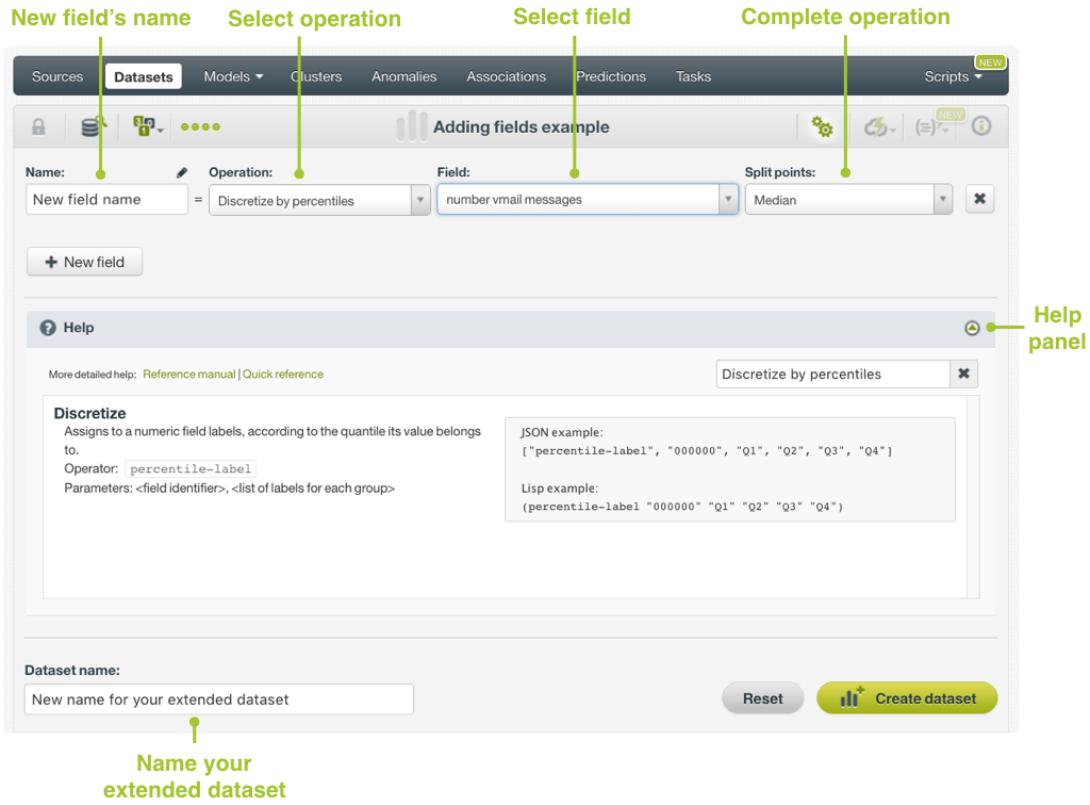

Figure 8.3: Configuration panel for adding fields

The following subsections define each of the operations you can apply to an existing field to create a new one.

8.1.1 Discretization

BigML offers three options to discretize your **numeric fields** to create new fields from them (See Figure 8.4):

- **Discretize by percentiles**: select a discretization value and BigML will split the field values into equal population segments (categories). Discretizing by percentiles will split the field values into 100 different categories, by quartiles into 4, by terciles into 3, etc.
- **Discretize by groups**: specify the number of groups and BigML will split the field values into equal width segments (categories), e.g., setting 3 groups for a field ranging from 0 to 6 will yield: category 1= [0,2], category 2= [2,4], category 3= [4,6].
- **Is within percentiles?**: specify a percentile range between 0 and 1 and you will get a boolean field with True or False values for each instance depending whether they belong to the specified range.

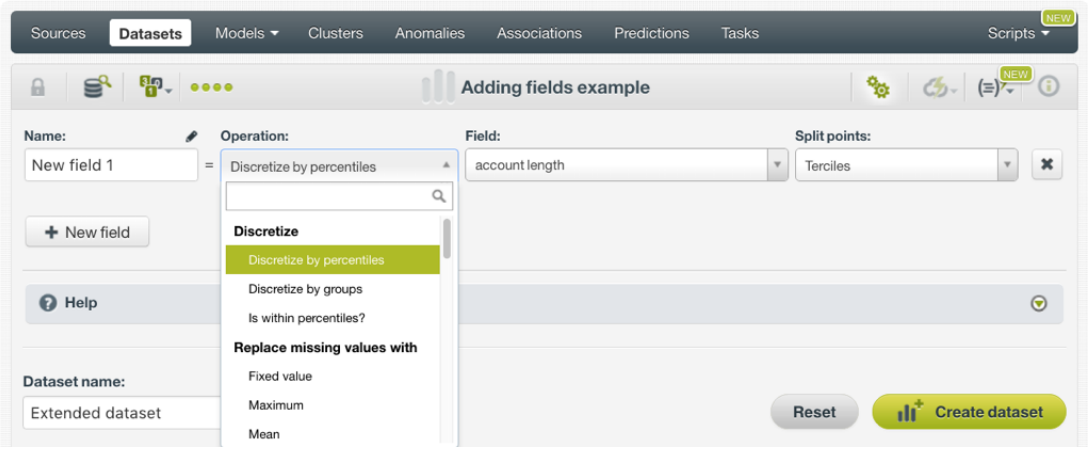

Figure 8.4: Adding new fields with discretization operations

8.1.2 Replacing Missing Values

Create new fields out of the selected **numeric field** by **replacing the missing values with** these operations (See Figure 8.5):

- **Fixed value**: all your field missing values will be replaced by the specified value. You can set a number or a string.
- **Maximum**: missing values will be replaced by the maximum value of the selected field.
- **Mean**: missing values will be replaced by the mean of the selected field.
- **Median**: missing values will be replaced by the median of the selected field.
- **Minimum**: missing values will be replaced by the minimum value of the selected field.
- **Population**: missing values will be replaced by the number of the total instances that have valid values for the selected field, e.g., for a field containing 54 instances with valid values, the missing values will be replaced by 54.
- **Random integer**: BigML creates a new field with a random value for each instance. You can set the maximum value you want for your random value generator.
- **Random value**: missing values will be replaced by a random value within your field range.
- **Random weighted value**: BigML sets a random value for your missing values within your field range but weighted by the population, so the population distribution for that field is used as a probability measure for the random value generator.

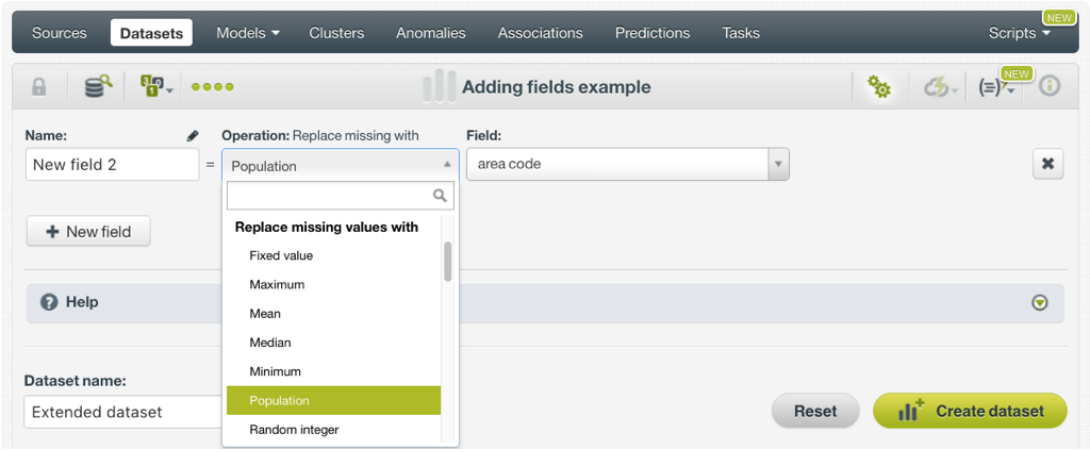

Figure 8.5: Adding new fields using replace missing values with operations

The operations **fixed value, random value, and random weighted value** are also available for **categorical fields**.

8.1.3 Normalizing

Create new fields out any **numeric fields** by **normalizing** them with the following operations (see Figure 8.6):

- **Normalize**: is a standardization of data distribution so your fields can be comparable. Select the range for which you want to normalize your field, which should be within the field range.
- **Z-score**: is a measure indicating the distance of the values from the mean.
- **Logarithmic normalization**: applies the z-score function to the logarithm of the values in the given field.

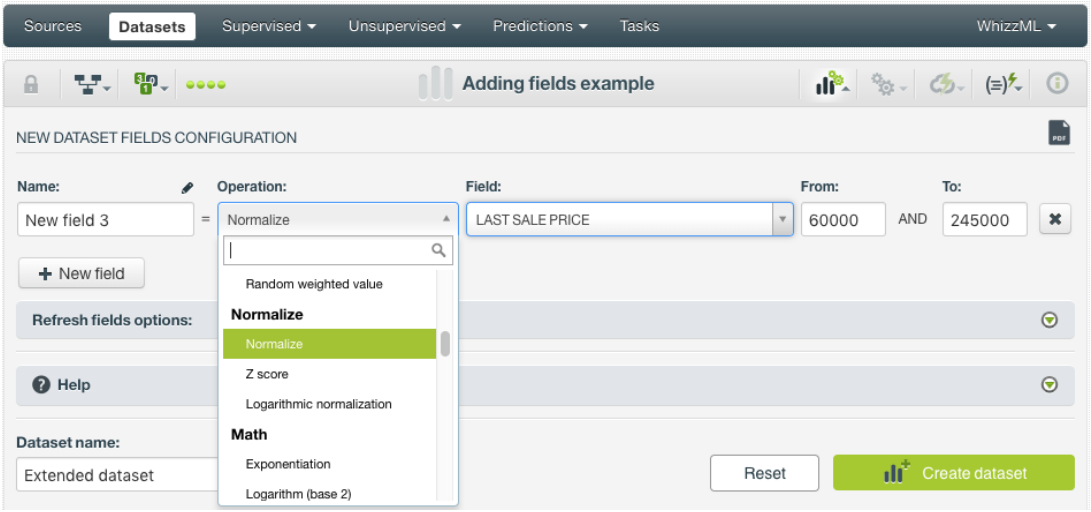

Figure 8.6: Adding new fields with normalizing operations

8.1.4 Math

You can also create new fields out of any **numeric fields** by applying any of the following **math** operations (see Figure 8.7):

- **Exponentiation**: computes e elevated to the field value: e^x.

- **Logarithm (base 2)**: converts fields into a logarithmic scale. This is useful for fields with a wide range of data (since it reduces the range to a more manageable scale) and to find exponential patterns in your data.
- **Logarithm (base 10)**: converts fields into a logarithmic scale.
- **Logarithm (natural)**: converts fields into a logarithmic scale.
- **Square**: elevates the value to the square: x^2.
- **Square root**: computes the square root of the value: $\sqrt{x}$.

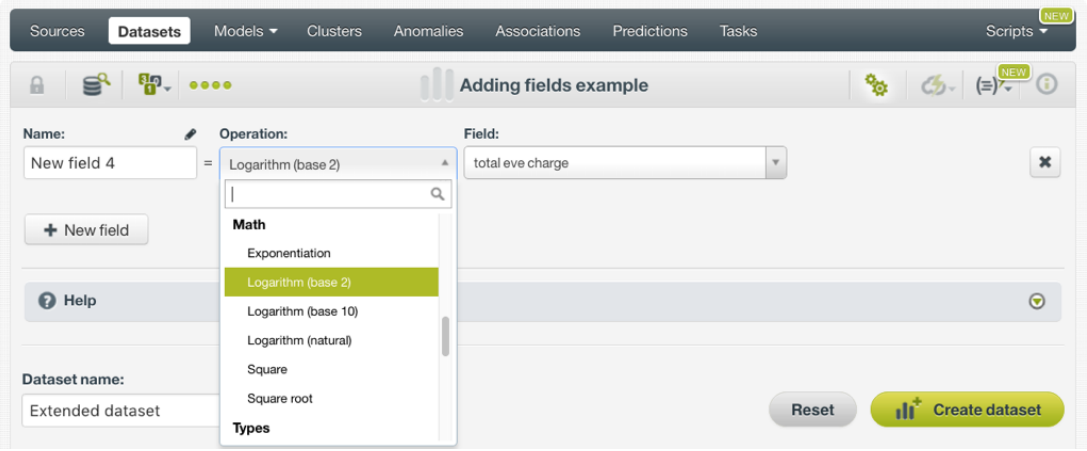

Figure 8.7: Adding new fields with math operations

8.1.5 Sliding Windows

Creating new features using **sliding windows** is one of the most common **feature engineering** techniques in Machine Learning. It is usually applied to frame **time series data** using previous data points as new input fields to predict the next time data points.

For example, imagine we have one year of sales data to **predict sales**. We have the daily sales (our objective field) and other information such as the holidays, the offers in the shop, etc. (our predictors). (See Figure 8.8). As domain experts, we know that past sales can be key predictors to predict today's sales. Therefore, we can use our objective field **"sales"** to create **additional input fields** that contain past data. We can create an infinite number of fields, last day sales, the average of last week sales, the difference between last month and this month sales, etc. However, we need to be very careful not to include today or future sales data in these new features; otherwise, we will be introducing leakage[1] in our model. For example, in the Figure 8.8 below, we are creating a new predictor that calculates the average sales of the last two days (see the field in green "avgSales_L2D"). This is a sliding window in which the window starts at -2 and it ends at -1.

[1] https://machinelearningmastery.com/data-leakage-machine-learning/

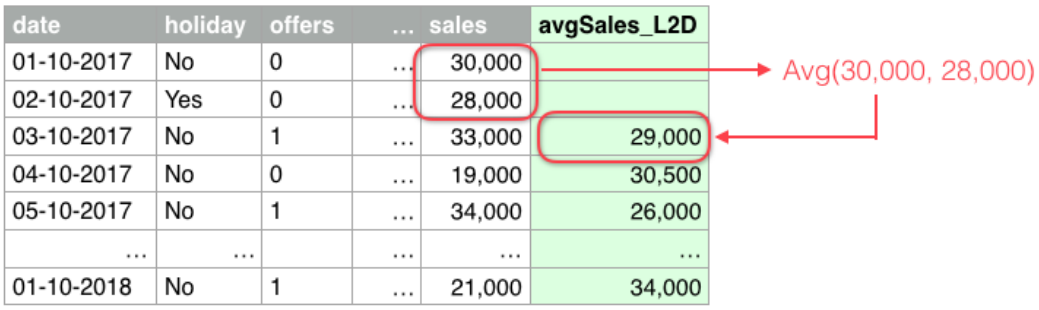

Figure 8.8: Example of sliding window that calculates the sales average of the last two days

In BigML, you can define the following operations and parameters to create sliding windows:

- **Operation**: select one of the below operations to be applied to the instances in the window (see Figure 8.9).
  - **Sum of instances**: sums consecutive instances by defining a window start and end. For example, for a sales dataset where each instance is a different day, we can get the sum of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Mean of instances**: calculates the mean of consecutive instances by defining a window start and end (negative values are previous instances and positive values next instances). For example, for a sales dataset where each instance is a different day, we can get the mean of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Median of instances**: calculates the median of consecutive instances by defining a window start and end (negative values are previous instances and positive values next instances). For example, for a sales dataset where each instance is a different day, we can get the median of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Minimum of instances**: calculates the minimum of consecutive instances by defining a window start and end (negative values are previous instances and positive values next instances). For example, for a sales dataset where each instance is a different day, we can get the minimum of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Maximum of instances**: calculates the maximum of consecutive instances by defining a window start and end (negative values are previous instances and positive values next instances). For example, for a sales dataset where each instance is a different day, we can get the maximum of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Product of instances**: calculates the product of consecutive instances by defining a window start and end (negative values are previous instances and positive values next instances). For example, for a sales dataset where each instance is a different day, we can get the product of sales of the previous 5 days (including today) by defining a window that starts at -5 and ends at 0 relative to each instance in the dataset.
  - **Difference from first**: calculates the difference between values associated with the start and end indices of the window, where the end index must be greater than the start index and the difference is calculated as end - start. For example, for a sales dataset where each instance is a different day, we can get the difference between yesterday and today's sales $[Sales(today) - Sales(yesterday)]$ by defining a window that starts at -1 and ends at 0.

  - **Difference from first (%)**: calculates the percentage difference between values associated with the start and end indices of the window, where the end index must be greater than the start index and the difference is calculated as end - start. For example, for a sales dataset where each instance is a different day, we can get the percentage difference between yesterday and today's sales $[Sales(today) - Sales(yesterday)]/Sales(yesterday)$ by defining a window that starts at -1 and ends at 0.
  - **Difference from last**: calculates the difference between values associated with the start and end indices of the window, where the end index must be greater than the start index and the difference is calculated as start - end. For example, for a sales dataset where each instance is a different day, we can get the difference between today and tomorrow's sales $[Sales(today) - Sales(tomorrow)]$ by defining a window that starts at 0 and ends at 1.
  - **Difference from last (%)**: calculates the percentage difference between values associated with the start and end indices of the window, where the end index must be greater than the start index and the difference is calculated as start - end. For example, for a sales dataset where each instance is a different day, we can get the percentage difference between today and tomorrow's sales $[Sales(today) - Sales(tomorrow)]/Sales(tomorrow)$ by defining a window that starts at 0 and ends at 1.

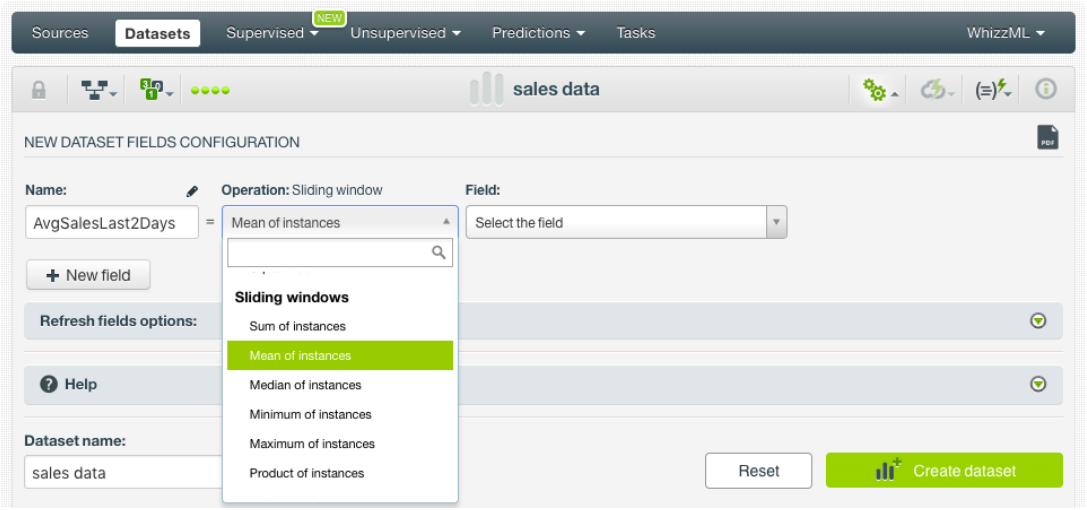

Figure 8.9: Select the operation for the instances in the sliding window

- **Field**: you can only select numeric fields to calculate sliding windows.

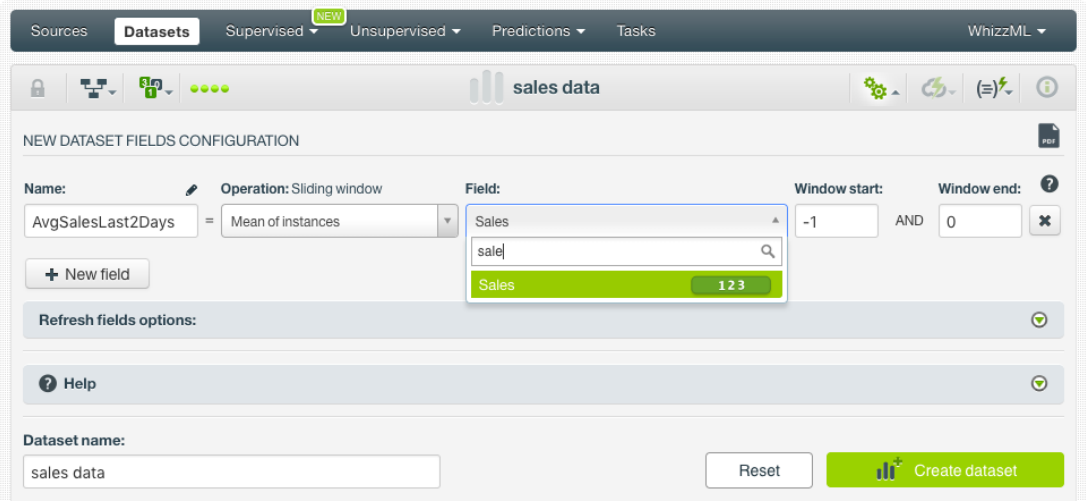

Figure 8.10: Select a field to calculate the sliding window

- **Window Start**: the start of the window defines the first instance to be considered for the defined calculation. Negative values are previous instances and positive values next instances. The 0 is the current instance.
- **Window End**: the end of the window defines the last instance to be considered for the defined calculation. Negative values are previous instances and positive values next instances. The 0 is the current instance.

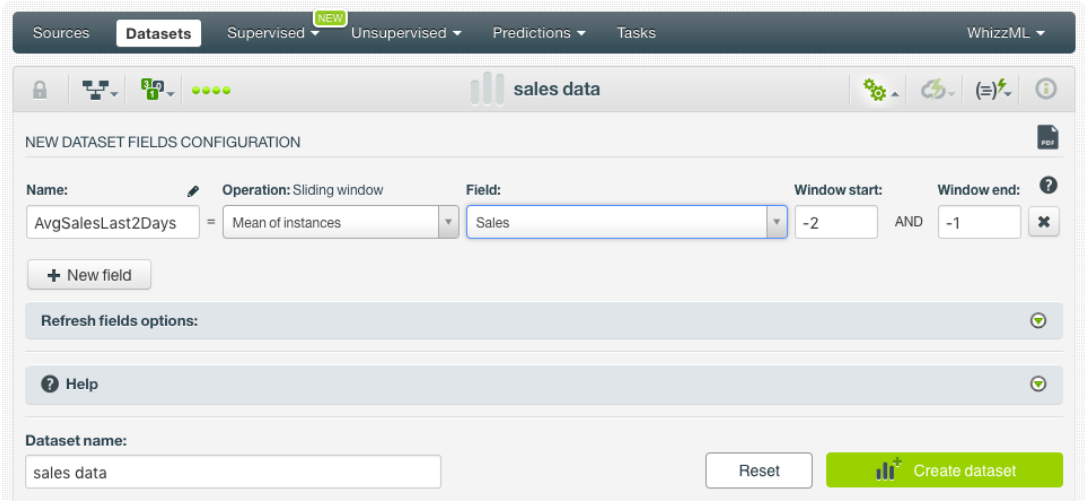

Figure 8.11: Set a window start and end

- Finally, click Create dataset and you will be able to see the new fields containing the sliding window calculations at the end of the new dataset.

8.1.6 Types

To create new fields from a **categorical, text, or items field**, use the **types** operations explained below (see Figure 8.12). **Note: only the categorical operation is available for numeric fields**:

- **Categorical**: coerce numeric field values into categorical values, e.g., the number 10 will become a string "10".
- **Integer**: coerce categorical values to integer values, e.g., the string "7.5 pounds" will become 7. Boolean values are assigned 0 (false) and 1 (true).
- **Real**: coerce categorical values to float values, e.g., the string "7.5 pounds" will become 7.5. Boolean values are assigned 0 and 1.

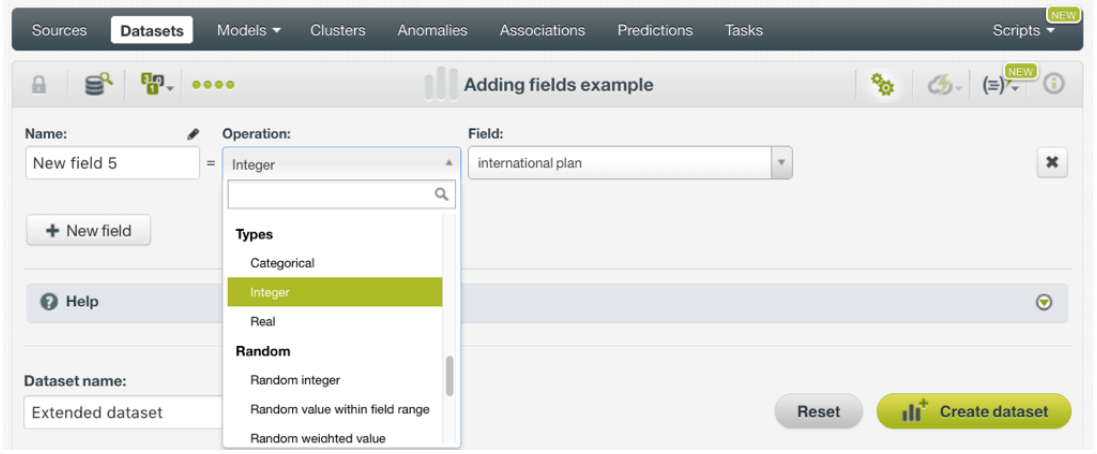

Figure 8.12: Adding new fields with types operations

8.1.7 Random

Random operations are available for **numeric and categorical fields**, except the first operation (random integer) which does not have any field type associated with it:

- **Random integer**: BigML creates a new field with a random value for each instance.
- **Random value within field range**: BigML sets a random value but takes your field range as the reference for minimum and maximum values.
- **Random weighted value**: BigML sets a random value within your field range weightened by the population, so the population distribution for that field is used as a probability measure for the random generator.

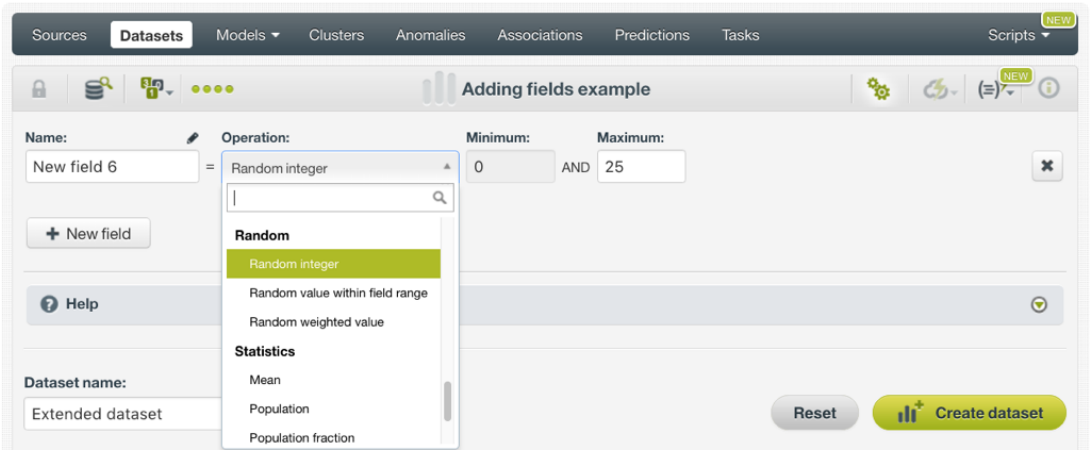

Figure 8.13: Adding new fields with random operations

8.1.8 Statistics

Another option to add new fields to your dataset based on your **numeric fields** is by applying **statistics** operations (see Figure 8.14):

- **Mean**: computes the field mean for all instances.
- **Population**: computes the count of total instances for that field.
- **Population fraction**: computes the number of instances whose values are below the specified value.

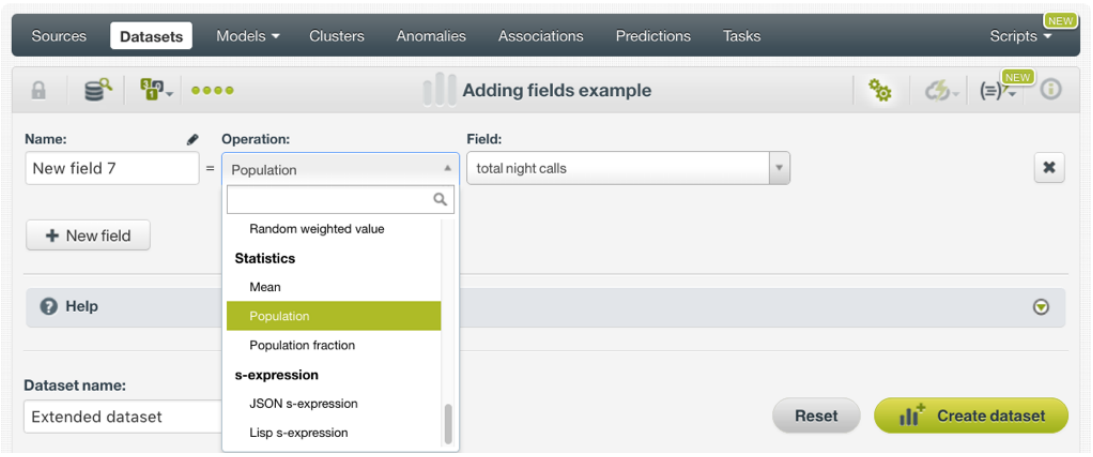

Figure 8.14: Adding new fields with statistics operations

8.1.9 Write Flatline Formula

In addition to all the operations explained the above subsections, BigML lets you perform any kind of operations with an flatline formula. Similar to filtering fields of your dataset, type the desired formula in either Lisp[2] or JSON[3] syntax.

Furthermore, use BigML Flatline editor, a powerful and flexible open-source lisp-like language, to create and validate your formulas before using them. To access the Flatline editor, first select the syntax you want to use, and click the highlighted icon in Figure 8.15, which leads to the Flatline editor.

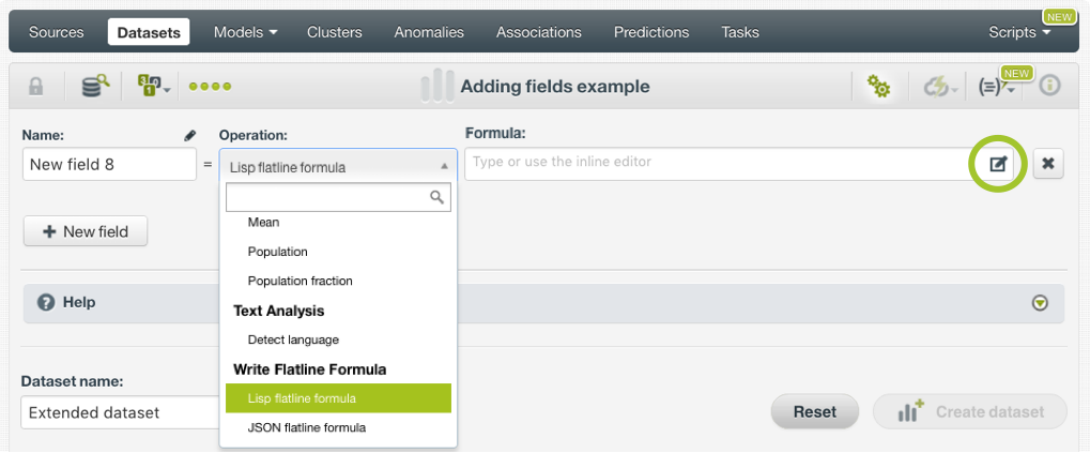

Figure 8.15: Adding new fields writing custom formulas

Note: the Flatline editor can be used to add new fields to your dataset following the same procedure as when filtering your dataset. (See Subsection 7.3.7.)

8.1.10 View and Reuse New Fields' Formulas

When you add new fields to a dataset, you will be able to **view the formulas** used to create them by clicking the option shown in Figure 8.16.

[2] https://en.wikipedia.org/wiki/Lisp_(programming_language)
[3] https://en.wikipedia.org/wiki/JSON

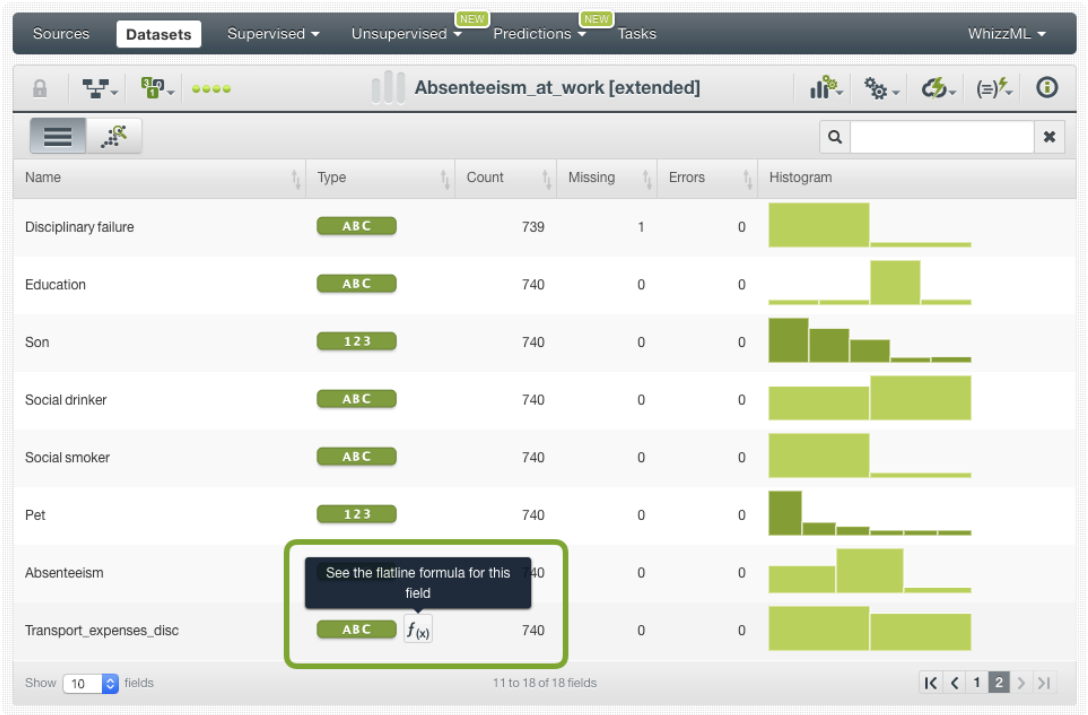

Figure 8.16: View the formulas used to create new fields

This option will display a window with the **Flatline formula** that is found underneath any new field in a dataset (see Figure 8.17). You can copy or download the formula (in Lisp and JSON formats) to create the same field using other datasets.

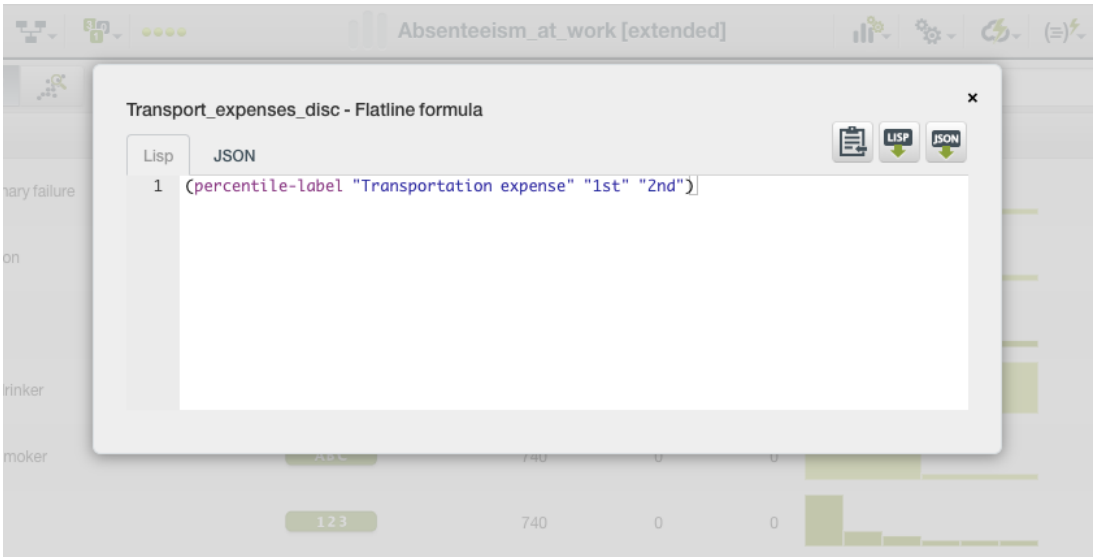

Figure 8.17: Copy and download formula

8.2 Aggregating Instances

The **aggregating instances** option in BigML allows you to group the rows of a dataset by a given field. This is a very common transformation to prepare your data for Machine Learning models. For example, imagine you have customer data stored in a dataset where each purchase is a different row. If you want to use this dataset to train models to analyze customers' purchase behaviors, you need a dataset where each row is a customer instead of a purchase. This is the case of the dataset in Figure 8.18 where

we can aggregate the instances by the field “customerID” to get a row per unique customer. Apart from grouping the instances by customer, we also need to add the purchase information per customer. We can do this by defining some aggregation functions on top of the former fields per purchase. For example, in the image below you can find the total purchases per customer (“Count_customerID”), the total units purchased (“Sum_Quantity”), the first purchase date (“Min_Date”) and the average price per unit spent per customer (“Avg_UnitPrice”).

InvoiceNo	Description	...	Quantity	InvoiceDate	UnitPrice	CustomerID
536365	WHITE HANGING HEART T-LIGHT HOLDER	...	6	12/1/18	2.55	17850
536366	WHITE METAL LANTERN	...	6	23/1/18	3.39	17850
536367	CREAM CUPID HEARTS COAT HANGER	...	8	18/2/18	2.75	17850
536368	KNITTED UNION FLAG HOT WATER BOTTLE	...	6	15/1/18	3.39	12583
536369	RED WOOLLY HOTTIE WHITE HEART.	...	6	10/3/18	3.39	12583
...	...	...	...		...	...
739411	JAM MAKING SET WITH JARS	...	4	25/2/18	4.29	13047

Aggregate instances by customer ID

CustomerID	Count_CustomerID	...	Sum_Quantity	Min_Date	Avg_UnitPrice
17850	3	...	20	12/1/18	2.90
12583	2		12	15/1/18	3.39
...	...	...	...	...	...
13047	1	...	4	25/2/18	4.29

Figure 8.18: Aggregate instances by customer ID example

The example above can be easily executed in the BigML Dashboard by following these steps:

- Find the AGGREGATE INSTANCES option in the dataset configuration menu (see Figure 8.19).

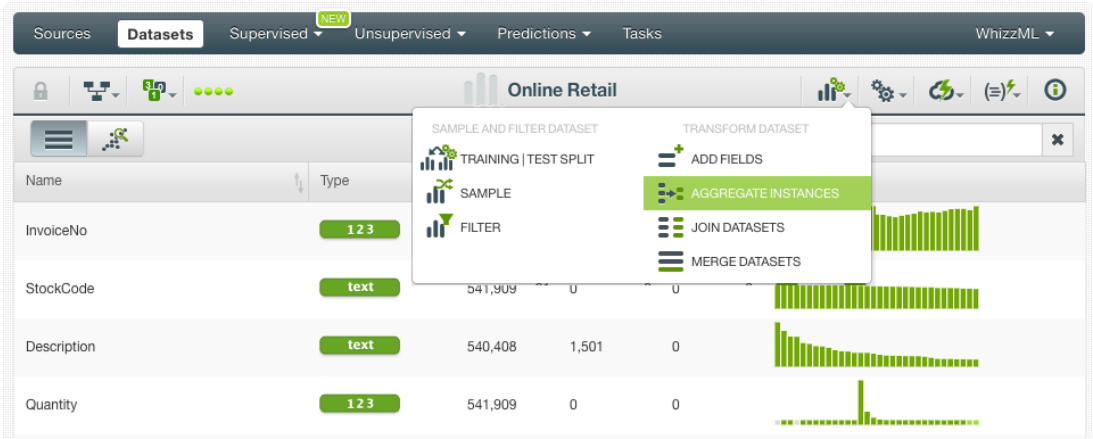

Figure 8.19: Select the option to aggregate the instances

- When the configuration panel has been displayed, **select a field** to aggregate your instances. You can select any type of field (numeric, categorical, text or datetime fields) and your instances will be grouped by the unique values of this field. In this case, we select “CustomerID” because we want a dataset with one row per customer (see Figure 8.20).

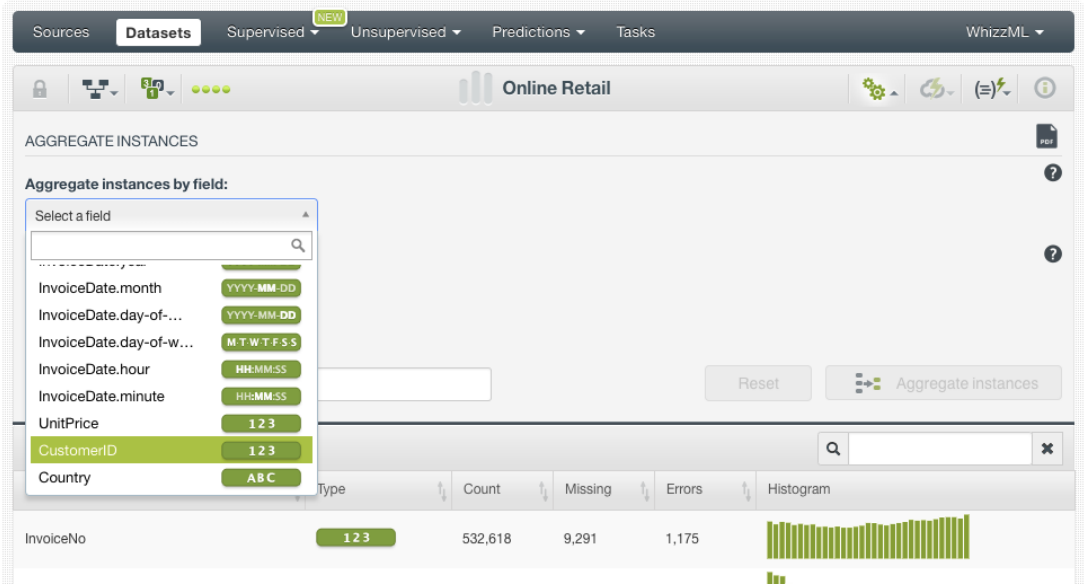

Figure 8.20: Select a field to aggregate the instances

You can optionally **add more aggregation fields** by clicking on the option shown in Figure 8.21. You can add up to **five fields** from the Dashboard, if you need to aggregate more fields you can use the API[4]. This option is very useful when you need to aggregate fields in a nested format, e.g. you may want that each row represents a customer per day.

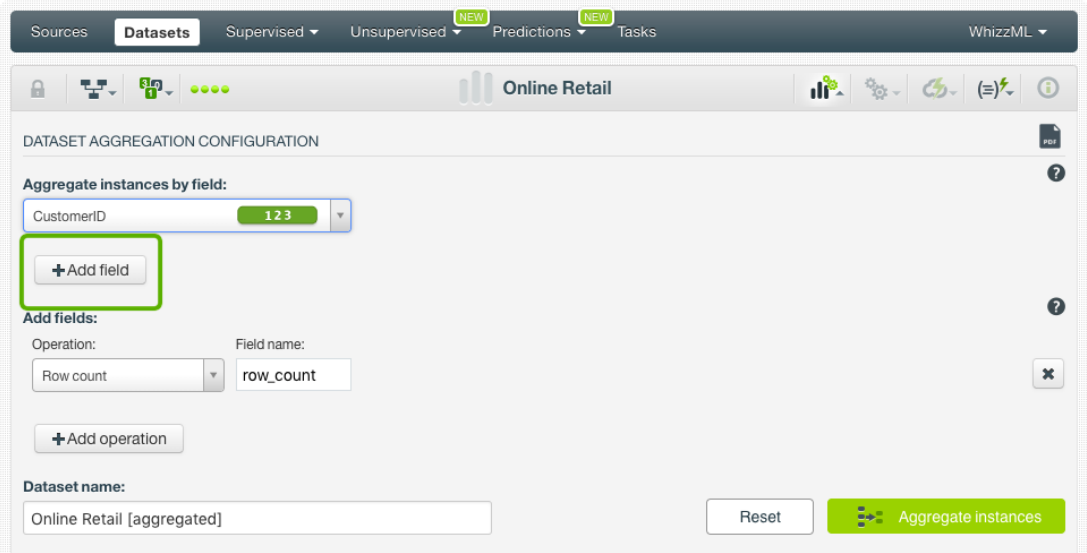

Figure 8.21: Add more aggregation fields

- When you select the aggregating field, you can see that BigML automatically displays an operation which is the **row count** to aggregate the instances. This operation calculates the number of rows per value for the aggregating field. In the example below (see Figure 8.22), the count operation on top of the "CustomerID" allows us to know the total purchases per customer. You can also remove this operation if you are not insterested in it by clicking the remove icon on the right-hand side.

[4] https://bigml.com/api/datasets

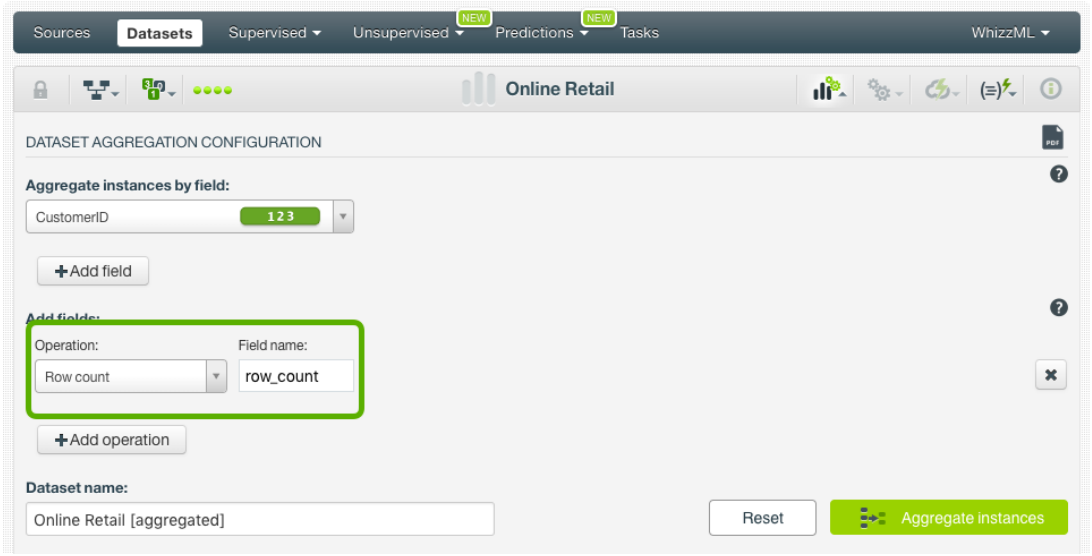

Figure 8.22: Row count operation by default

At this point, we can go ahead and create a new dataset that only contains two fields, the "CustomerID" and the "row_count". However, this new dataset with only two fields will not be very useful to train any Machine Learning model. We want to add more fields to the resulting dataset that gather as much information as we can about each customer's purchase behavior.

- You can add more fields to the dataset by defining additional **aggregation operations**. For example, imagine we want to know the total units purchased per customer, we can select **Sum** in the operation selector (see Figure 8.23):

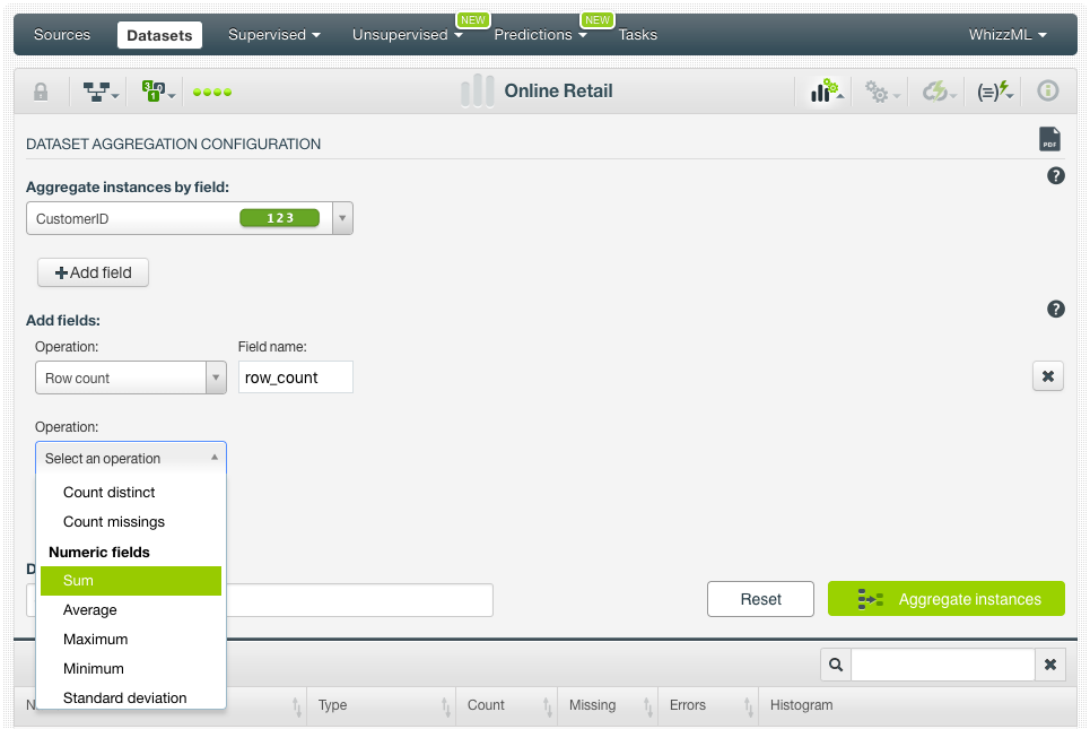

Figure 8.23: Add more operations

And then select the field "Quantity" as shown in Figure 8.24:

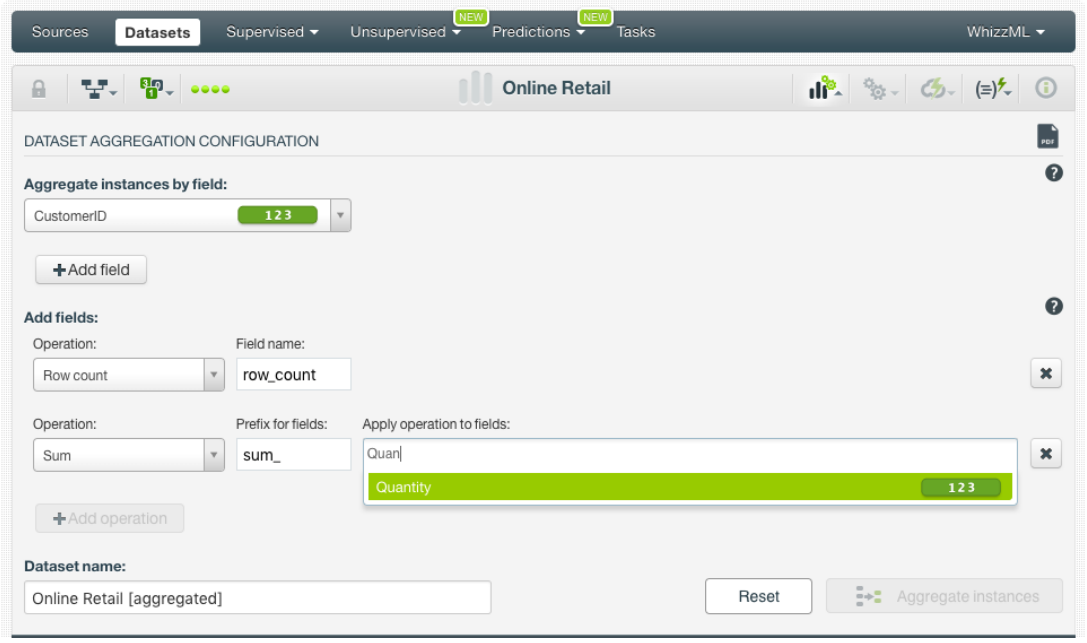

Figure 8.24: Select the field for the chosen operation

All operations have a **prefix for fields** defined. In the resulting dataset, all the fields that have a given operation applied will be renamed with the prefix before their actual names. This allows you to know the operation applied to a given field. You can edit this prefix name or remove it.

You can select the following **operations** depending on the field type:

- **Count**: counts the total rows per unique value of the aggregating field. It can be applied to all field types.
- **Count distinct**: counts the rows that have distinct values per unique value of the aggregating field. It can be applied to all field types.
- **Count missings**: counts the rows that have missing values per unique value of the aggregating field. It can be applied to all field types.
- **Sum**: sums the values of the aggregated instances. Only for numeric fields.
- **Average**: averages the values of the aggregated instances. Only for numeric fields.
- **Maximum**: takes the maximum value of the aggregated instances. Only for numeric fields.
- **Minimum**: takes the minimum value of the aggregated instances. Only for numeric fields.
- **Standard deviation**: takes the standard deviation of the aggregated instances. Only for numeric fields.
- **Variance**: takes the variance of the aggregated instances. Only for numeric fields.
- **Concatenate values**: concatenates the values of the aggregated instances. Only for categorical, text, and items fields. You can also define the separator and the final field type (text, categorical or items field).
- **Concatenate distinct values**: concatenates the distinct values of the aggregated instances. Only for categorical, text, and items fields. You can also define the separator and the final field type (text, categorical or items field).

Note: the fields in the original dataset that do not have an operation defined will be dropped from the final dataset.

For our example, we are defining more operations such as the total units purchased, the total price spent per customer, the average price per purchase per customer, and the concatenation of the purchased products descriptions (see Figure 8.25).

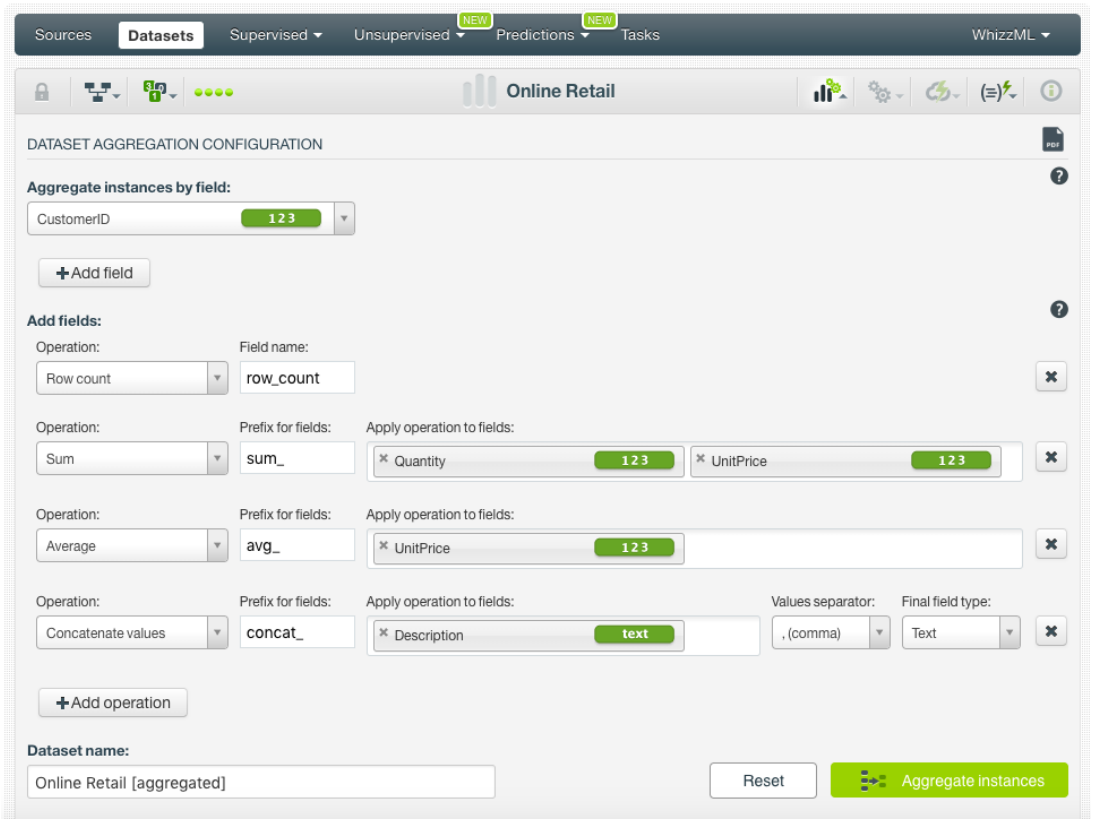

Figure 8.25: Define all the operations you want for the dataset fields

- Finally, click Aggregate instances and a new dataset with the new aggregated instances and field calculations will be created.

The aggregation option in the Dashboard uses an SQL query underneath. Therefore, when the new dataset is created, you can view the SQL query by clicking the option shown in Figure 8.26 below.

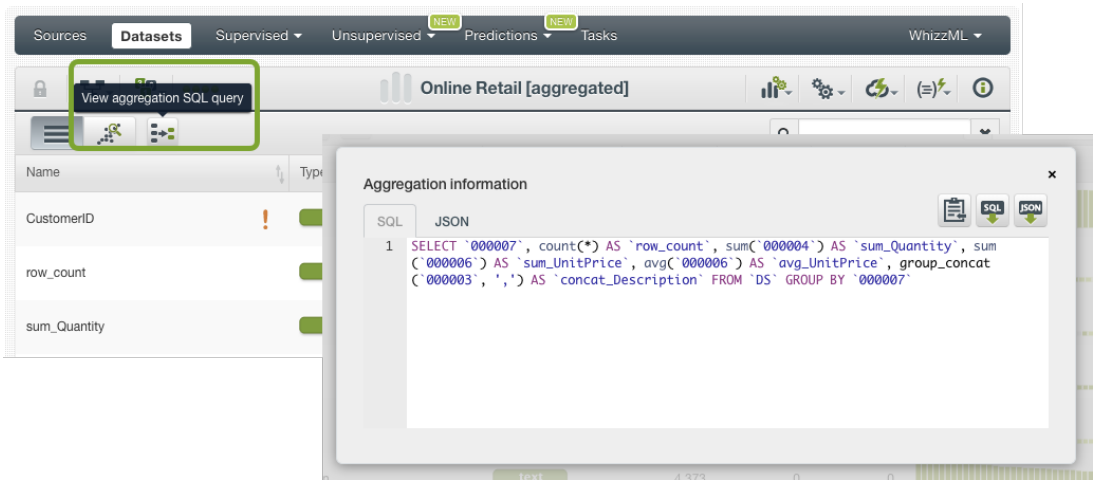

Figure 8.26: View the SQL query of the aggregation performed

8.3 Joining Datasets

It is very common to have the data scattered in two or more different datasets. BigML allows you to **join several datasets** to combine their **fields** and **instances** based on one or more related fields between them. For example, imagine we want to predict employee performance and we have two

different sources of data: a dataset containing employees' data (employee name, salary, age, etc.) and another dataset containing departments data (department name, budget, etc.). (See Figure 8.27). If we want to include the department data as additional predictor for our employees analysis, we can use a common field in both datasets (department_id) to add the department characteristics to the employee dataset.

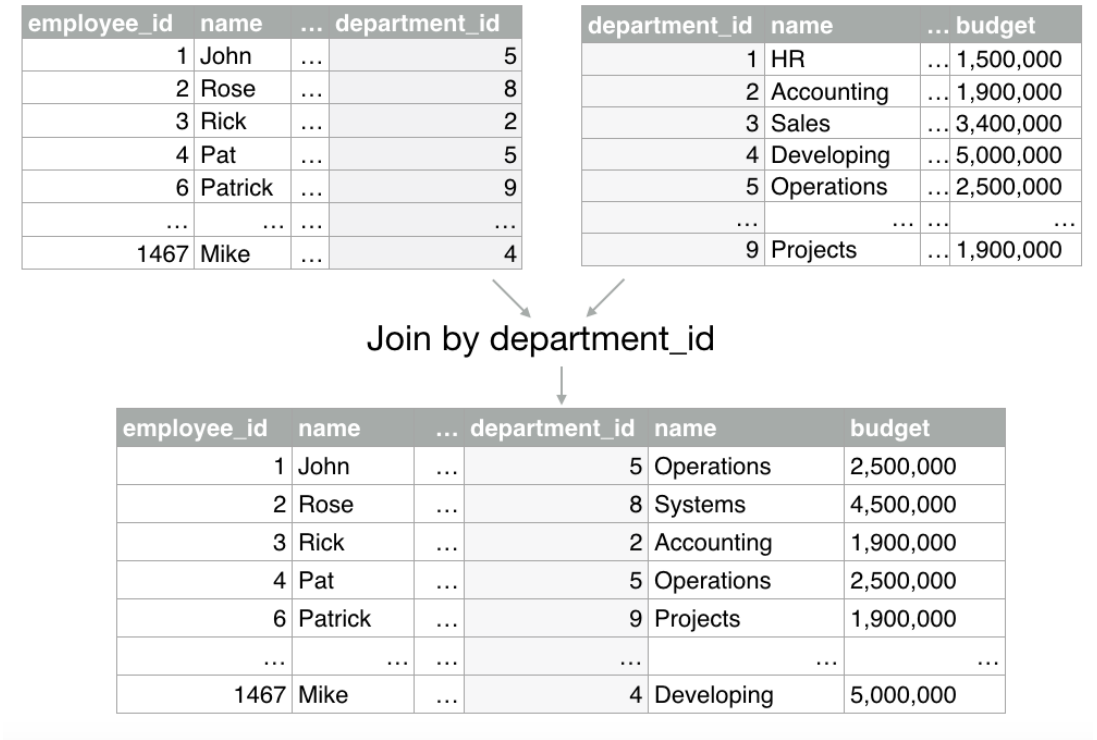

Figure 8.27: Join example

The example above can be easily executed in the BigML Dashboard by following these steps:

- First of all, you need to upload both sources to BigML and create a datasets from each source (see Chapter 3).
- When the datasets are created, find the JOIN DATASETS option in the employees dataset configuration menu as shown in Figure 8.28. We use the employees dataset and not the departments dataset because our ultimate goal is to analyze employee performance, hence we need to use employee data to train a Machine Learning model.

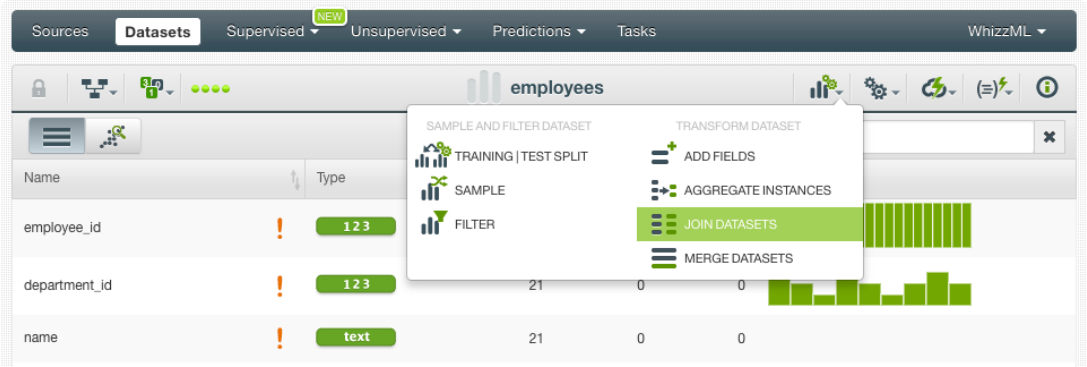

Figure 8.28: Join datasets

- This option will display the join configuration panel in which you need to input the following param-

eters:

- **Type of join**: you can perform four different types of join:
  - **Left join**: returns all the instances from the current (left) dataset, the employees dataset, and the matched instances from the selected (right) dataset, the departments dataset. If there are instances in the current dataset that do not have a matching instance in the selected dataset, the field values will be missing.
  - **Right join**: returns all the instances from the selected (right) dataset, and the matched instances from the current (left) dataset, the departments dataset. If there are instances in the selected dataset that do not have a matching instance in the current dataset, the field values will be missing.
  - **Full join**: returns the matched and unmatched instances in both datasets.
  - **Inner join**: returns the instances that have matching values in both datasets, the rest of instances will be droped.

  For our example we are performing a left join since we are interested in having all the employees data to make our predictive model and it is not so important if a given employee does not have a department assigned. In this case, the department information will be missing for that employee and the models in BigML can handle missing values afterward.

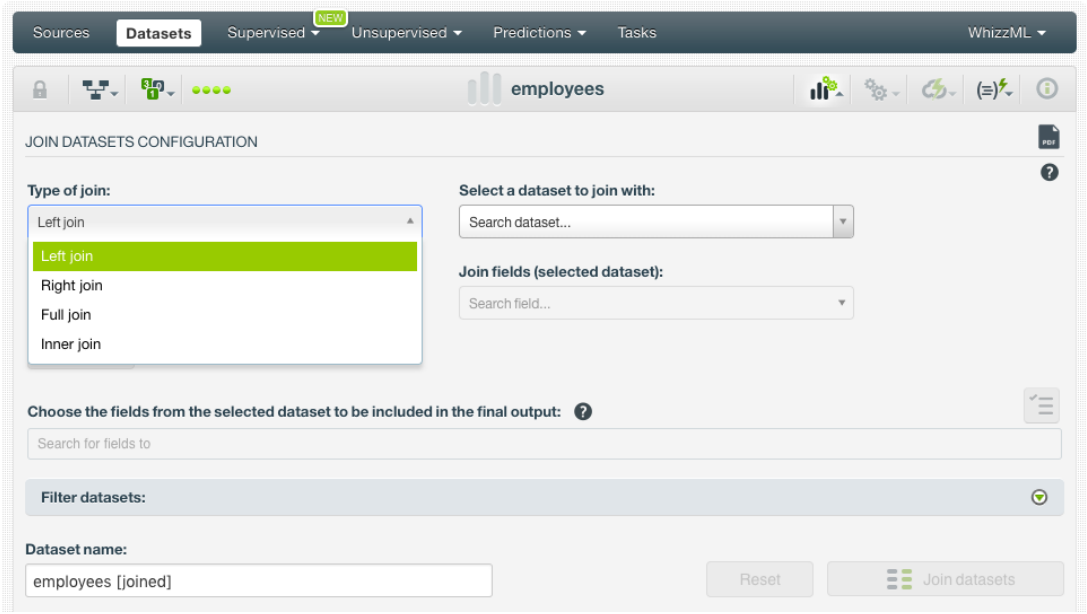

Figure 8.29: Select the type of join

- **Select a dataset to join with**: this is the dataset you want to join with the current dataset. Select a dataset that contains at least one field in common with the current dataset to perform the match between the instances.

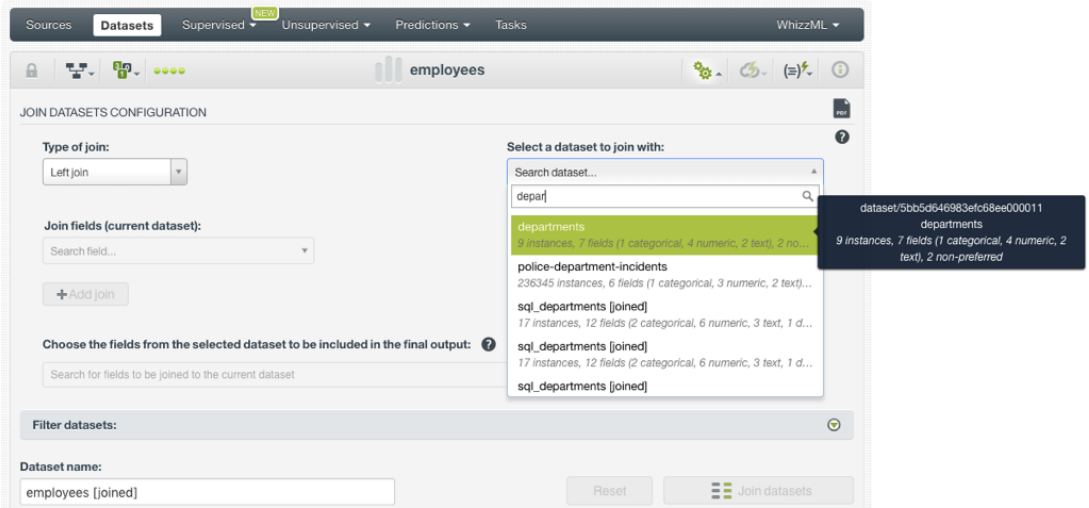

Figure 8.30: Select the dataset to make the join

– **Join fields (current dataset)**: select one or more fields from the current dataset (the employees dataset) to match the instances with the selected dataset (the departments dataset). These fields should have the same values in both datasets so the instances can be matched. Usually a field with unique values per instance such as an ID field is used here.

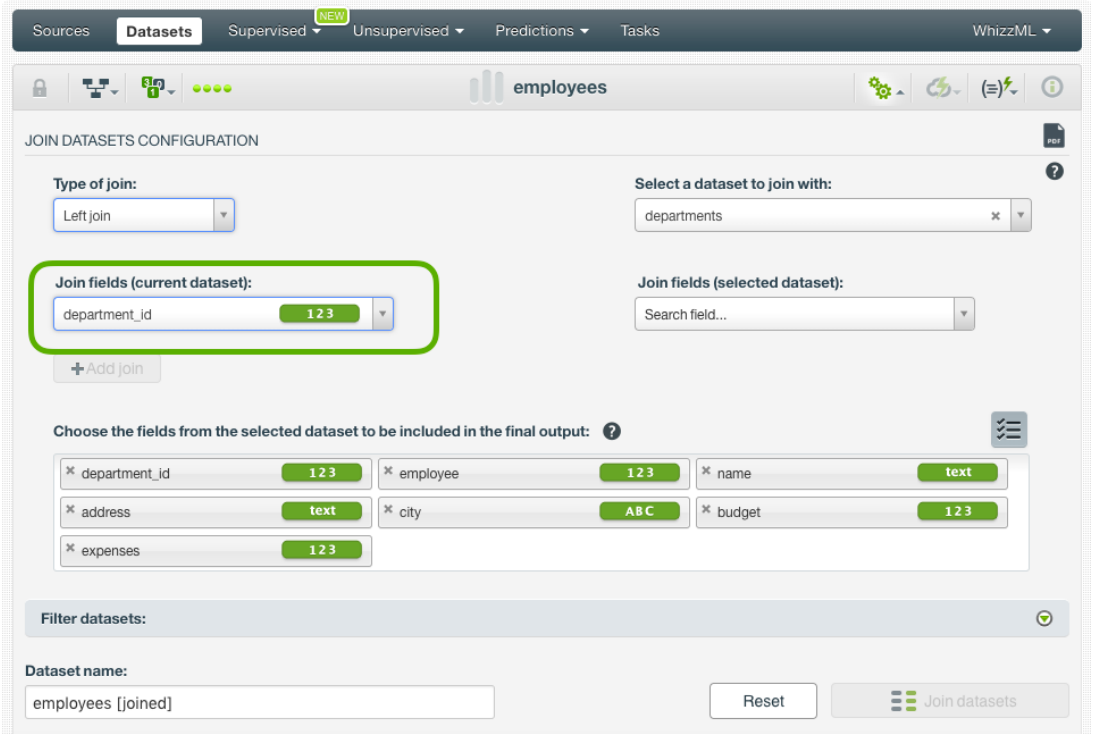

Figure 8.31: Select the join field from the current dataset

– **Join fields (selected dataset)**: select one or more fields from the selected dataset (the departments dataset) to match the instances with the current dataset (the employees dataset). These fields should have the same values in both datasets so the instances can be matched. Usually a field with unique values per instance such as an ID field is used here.

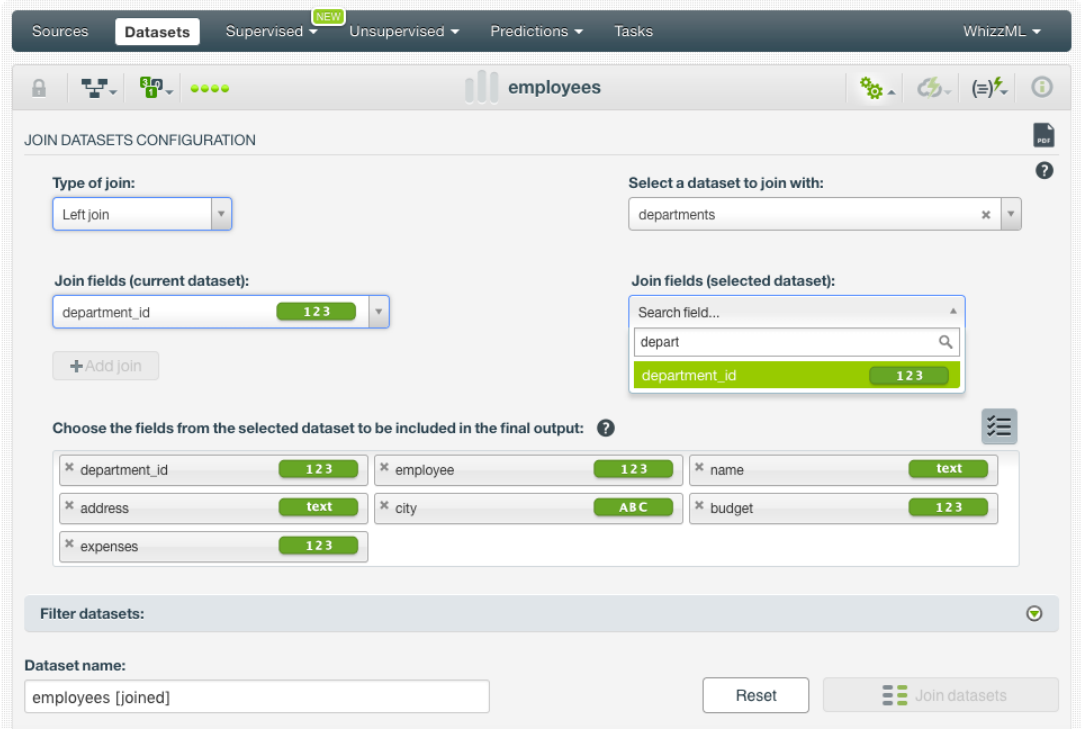

Figure 8.32: Select the join field from the selected dataset

– **Choose the fields from the selected dataset to be included in the final output**: you can choose to include all the fields from the selected dataset or select a subset of them.

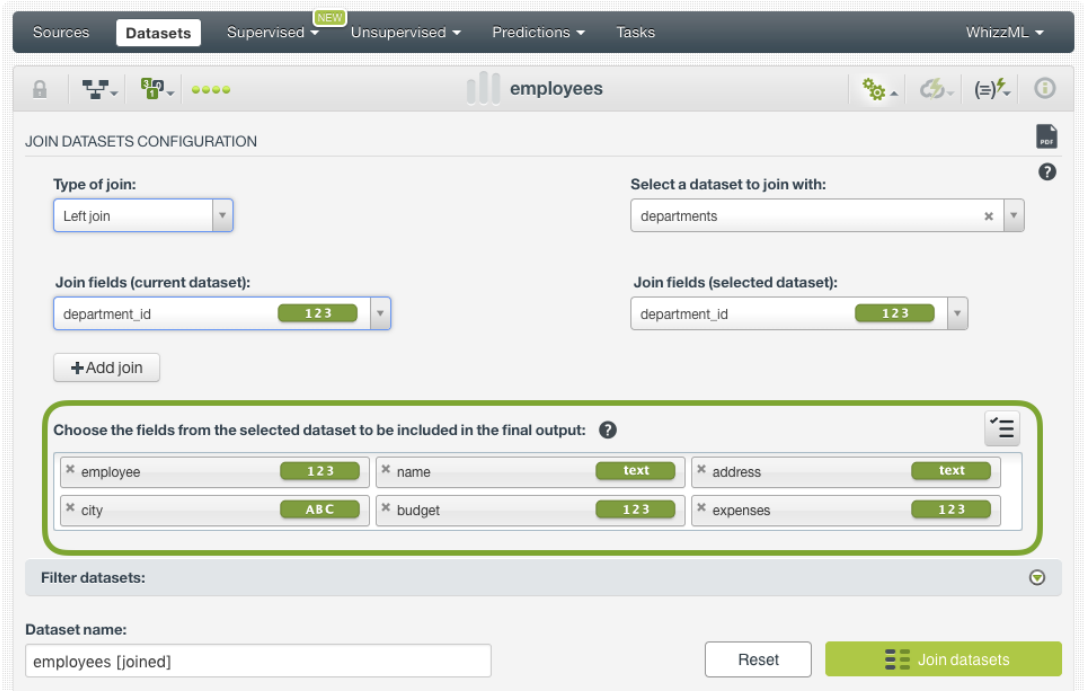

Figure 8.33: Choose the fields from the selected dataset

- Optionally, you can **filter** the **current** and/or the **selected dataset** before creating the new joined dataset. You can add up to six different filters. You can filter any type of field except full date-time fields. Please read more about filtering datasets in Section 7.3.

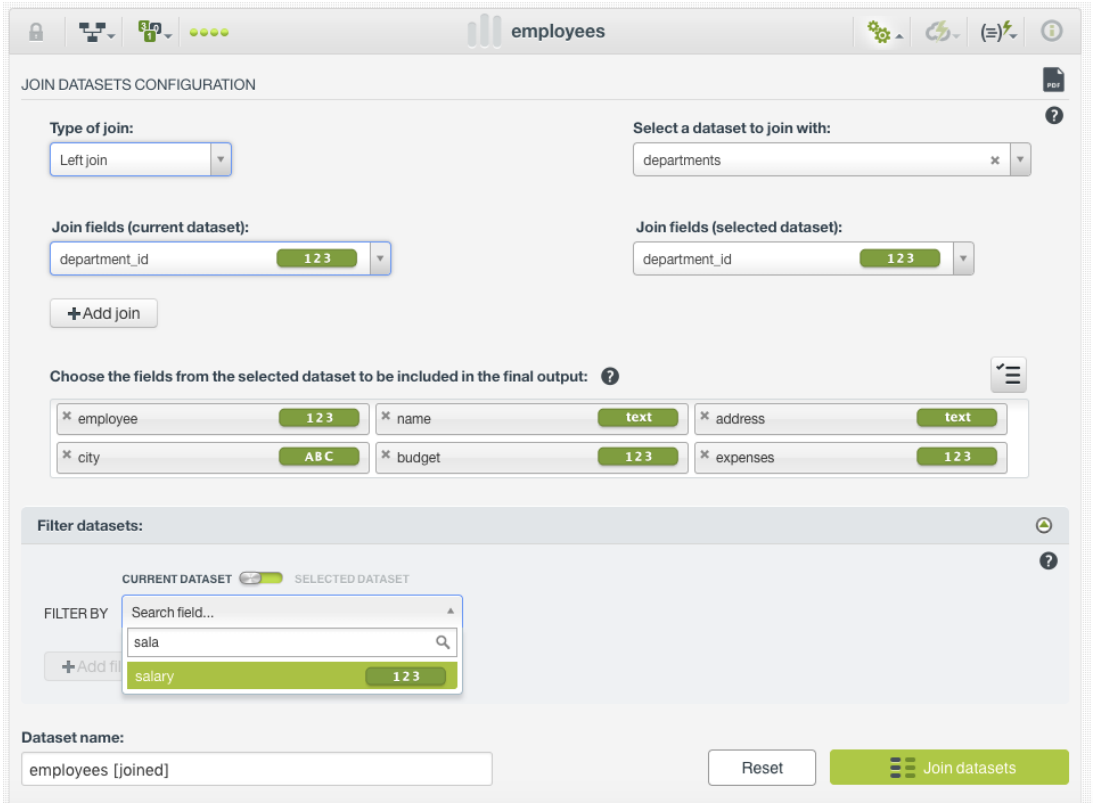

Figure 8.34: Filter one or more fields from the current and/or the selected dataset

- Click the Join datasets button.

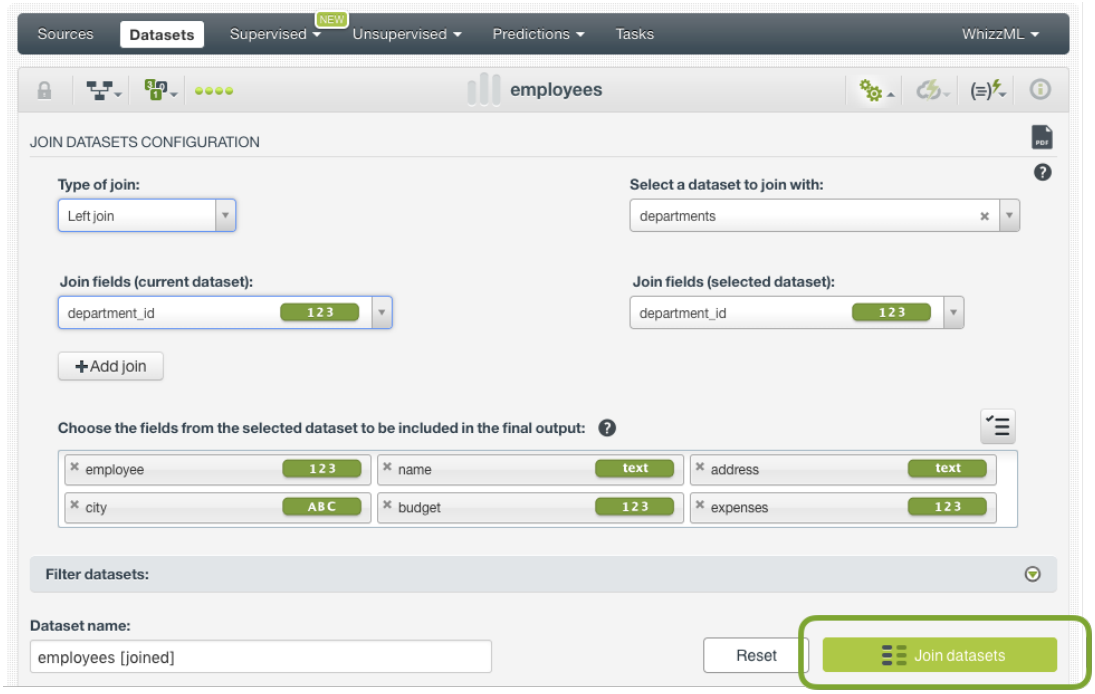

Figure 8.35: Join datasets

- A **new dataset** with the matched instances and the new fields will be created.

  Note: if each instance has one single match in both datasets, i.e., the join fields have unique

values per instance, the resulting dataset will have a maximum number of instances equal to the dataset with most instances. However, if the instances in one or more datasets have repeated values for the join fields, each instance in a given dataset will be matched as many times as it finds the same matching value in the other dataset. Therefore, the final number of instances may be much larger than the number of instances in both original datasets.

The join option in the Dashboard uses an SQL query underneath. Therefore, when the joined dataset is created, you can view the SQL query by clicking the option shown in Figure 8.36 below.

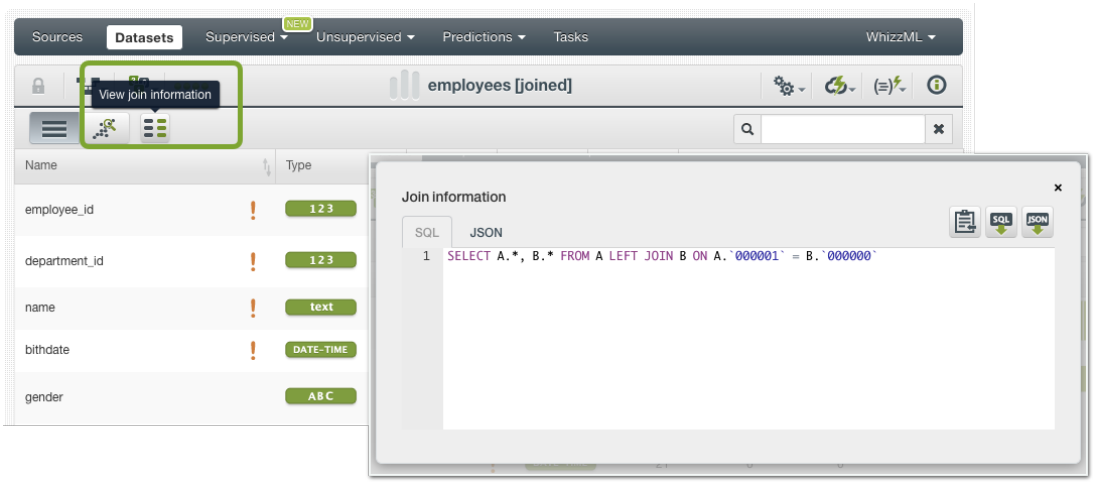

Figure 8.36: View join SQL query

8.4 Merging Datasets

In case you have instances in different datasets and you want to merge them all into one single dataset, you can do it using the **merging datasets** option in BigML. This functionality can be very useful when you use multiple sources of data. Imagine, for example, that you collect data on an hourly basis and want to create a dataset aggregating data collected over the whole day. You only need to send the new data generated each hour to BigML, create a source and a dataset for each one and then merge all the individual datasets into one at the end of the day.

For example, imagine we have employees data in two different datasets and we want to merge them into one dataset (see Figure 8.37).

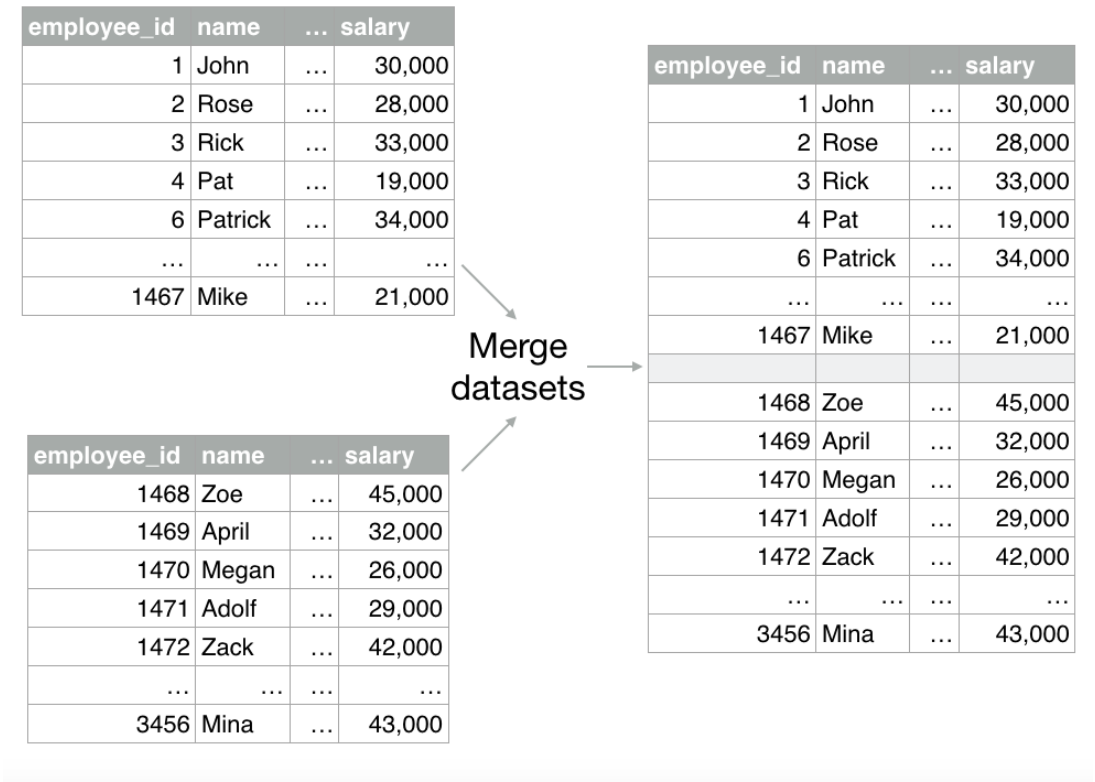

Figure 8.37: Merge datasets example

You can easily **merge datasets** in the BigML Dashboard by following these steps:

- From one of the datasets, open the CONFIGURE DATASET menu (see Figure 8.38). By convention, this first dataset defines the final dataset fields. All datasets should have the same field names and IDs. If this first dataset has fields not found in the other datasets, the merge will give an error. However, if the other datasets have some fields that are not found in the first dataset, you can still excute the merge and these fields will be dropped from the final dataset. You can map the fields from different datasets using the merging option from the API[5] for the moment.

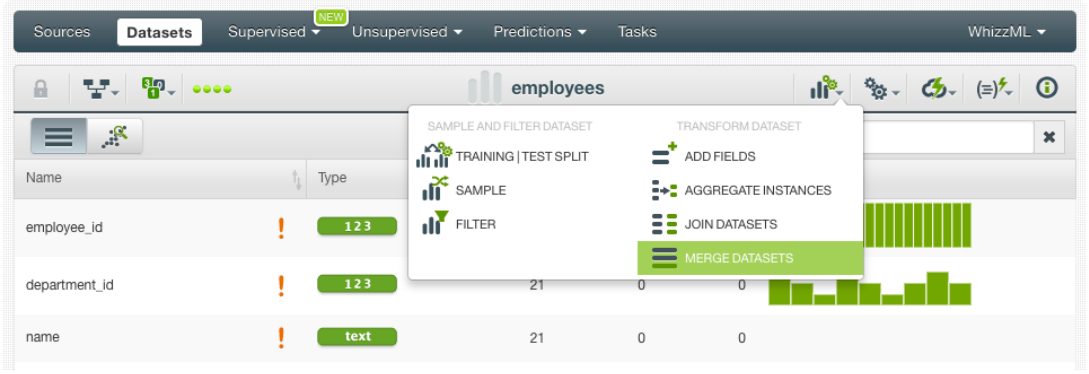

Figure 8.38: Select the merge datasets option

- Select the datasets you want to merge (see Figure 8.39).

[5] https://bigml.com/api/datasets#ds_multi_datasets

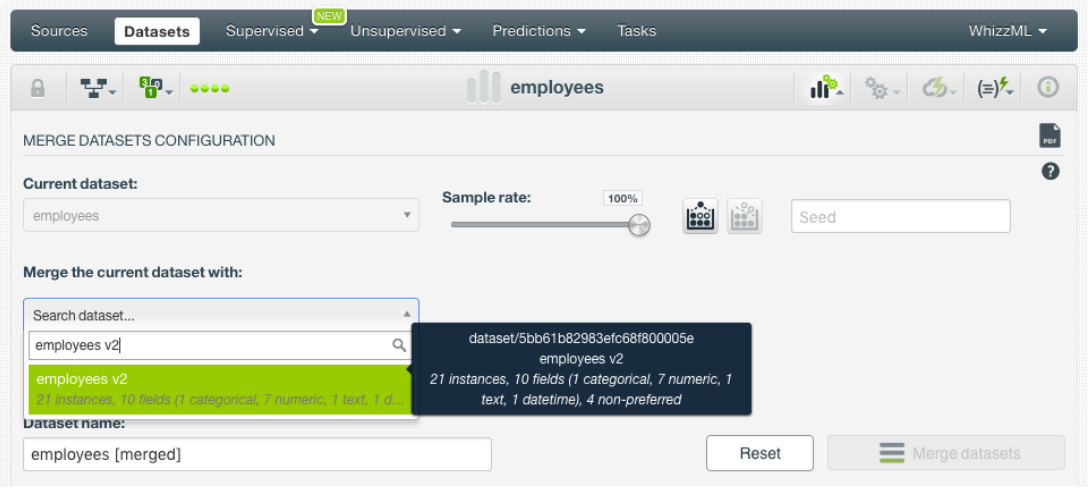

Figure 8.39: Select the datasets you want to merge

You can select up to **32 datasets** (see Figure 8.40).

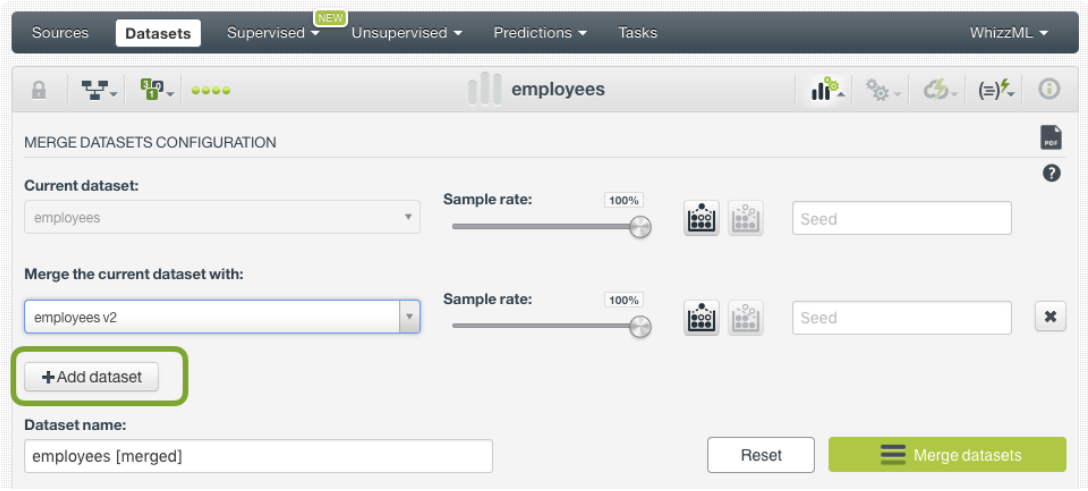

Figure 8.40: Add more datasets to merge

- You can **sample** each one of the selected datasets (see Section 7.2 to find an explanation for each sampling option).

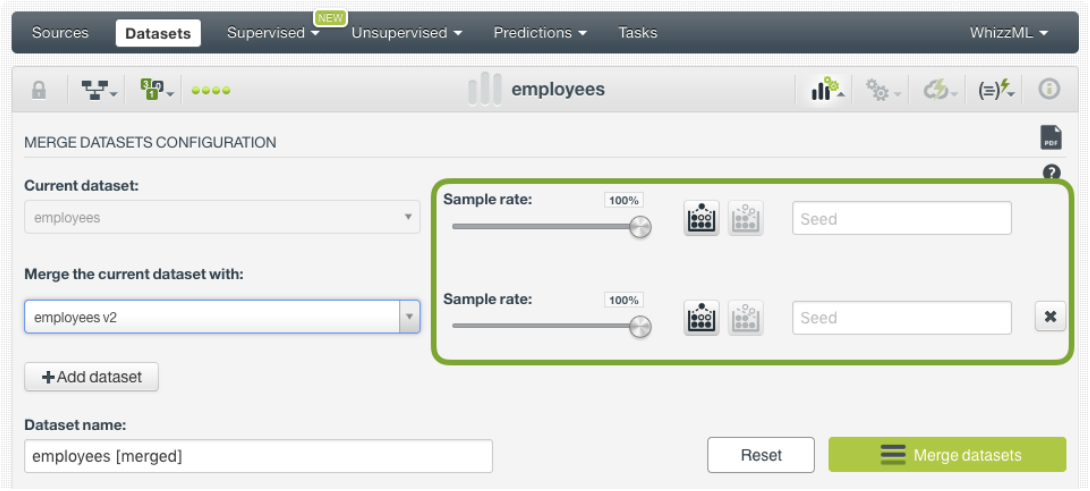

Figure 8.41: Sample the datasets to merge

- Click Merge datasets to create a new dataset with all the merged instances.

From the resulting dataset you can click the option shown in Figure 8.42 to see the merge configuration of each dataset.

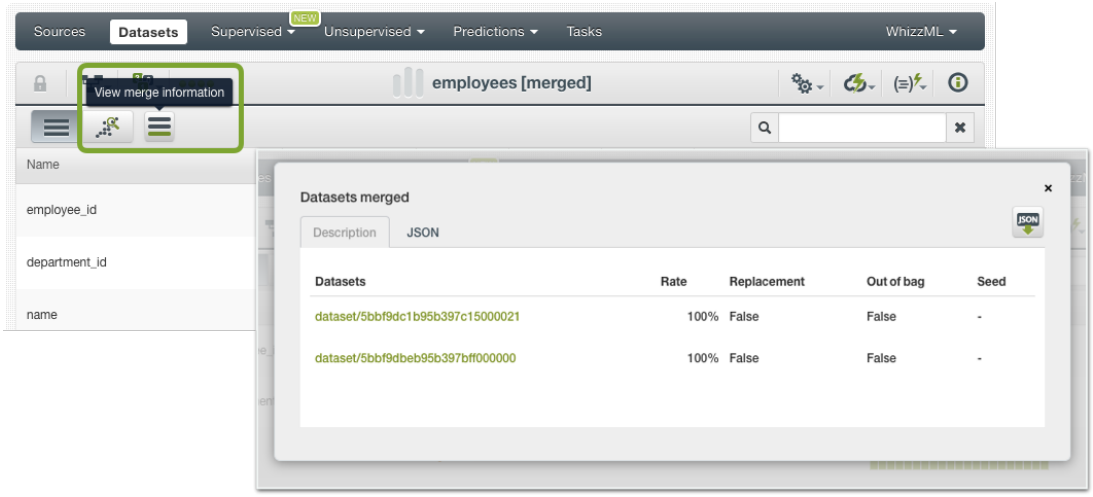

Figure 8.42: Merge configuration information

Note: the merging option is the only transformation option that does not use SQL query behind the scenes.

8.5 Ordering Instances

The **ordering instances** option in BigML allows you to sort the rows of a dataset by one or more selected fields in ascending or descending order. The instances will be sorted first by the first selected field, then by the second field, and so on. You can select up to 8 different sorting fields.This option is very useful for time series, when you have a dataset containing a date field and you need to sort your instances chronologically.

For example, imagine we have a dataset containing the monthly minimum temperatures in Melbourne (Australia) and they are not chronologically sorted (see left-side table in Figure 8.43). If we want to create a time series model using this dataset, we first need to sort instances in ascending order by date as you can see in the right-side table of the Figure 8.43 below.

Figure 8.43: Select the option to order instances

We can easily do this in BigML by following these steps:

- From the dataset view, click on the ORDER INSTANCES menu option (see Figure 8.44).

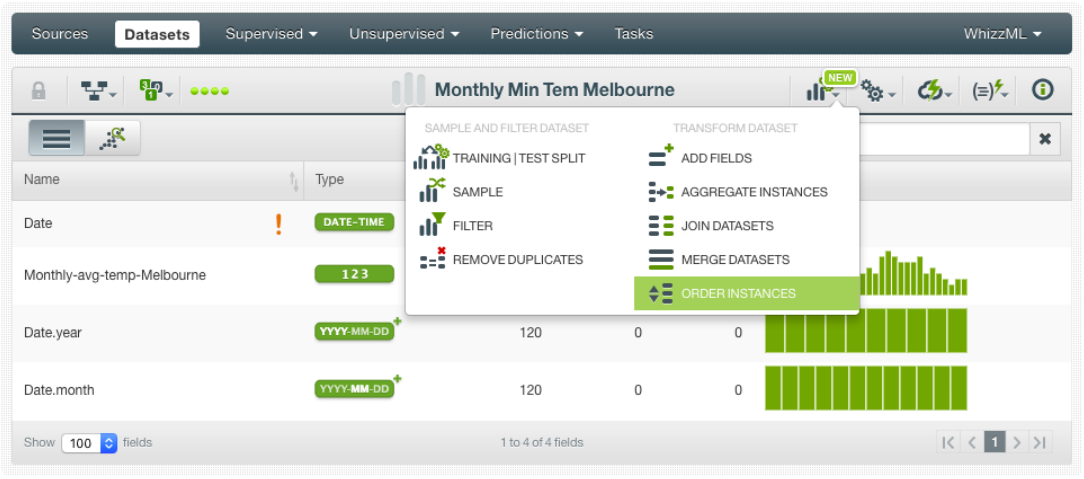

Figure 8.44: Select the option to order instances

- You cannot select the full date-time field to sort instances, but you can select the expanded fields (year, month, day of month, etc.) to do so. Remember when you select multiple fields to sort your intances, the first field is the one that decides the final order first, then the second field (keeping the order of the first field) and so on. That's why we need to select first the larger date unit, in this case the year, and then the next date unit, the month in this case (see Figure 8.45). Then click on the order instances button.

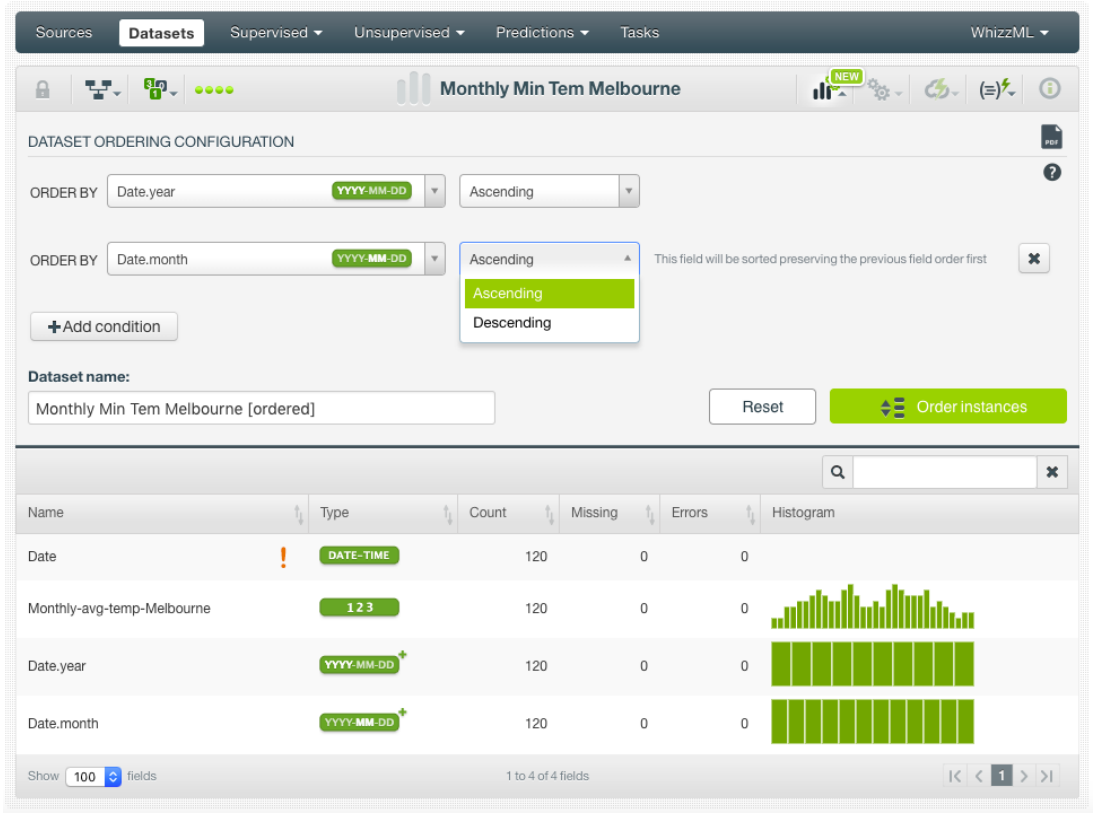

Figure 8.45: Select the fields you want to sort by

- A new dataset will be created with the sorted instances. You can see the confirmation message on top of the dataset view in blue color (see Figure 8.46).

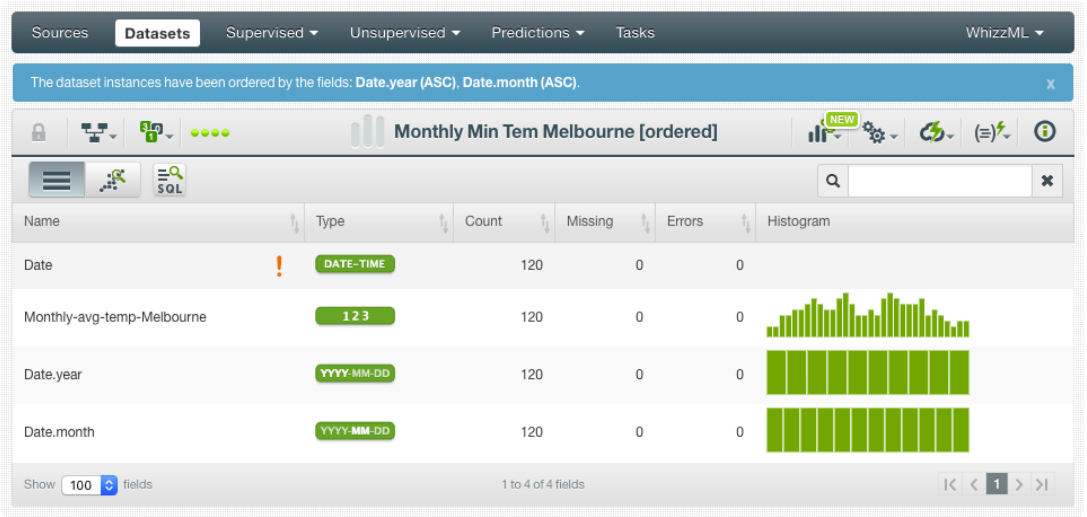

Figure 8.46: See the notification message

The ordering option in the Dashboard uses an SQL query underneath. Therefore, when the dataset is created, you can view the SQL query by clicking the option shown in Figure 8.47 below.

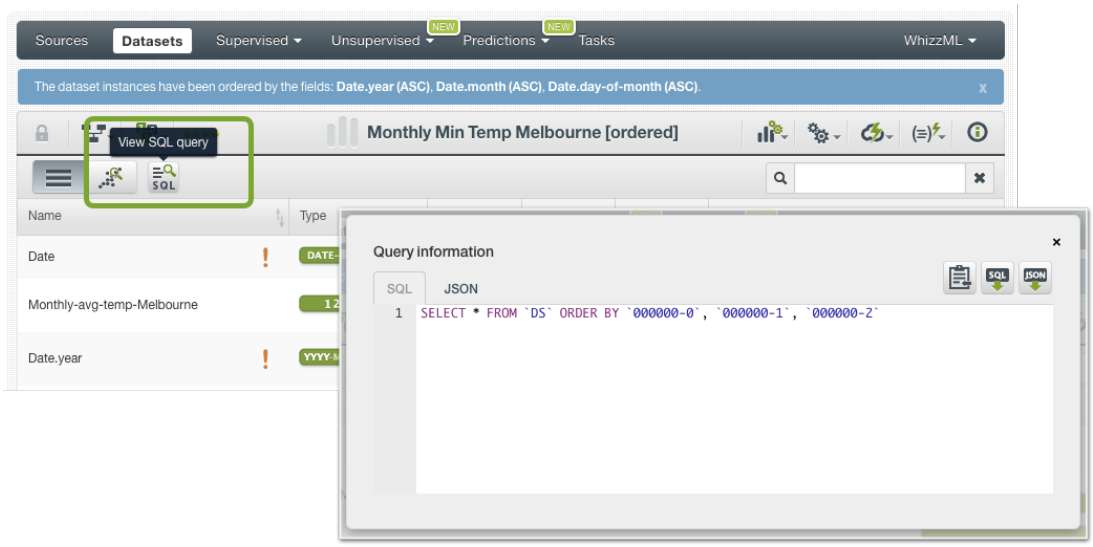

Figure 8.47: View SQL query to order instances

CHAPTER 9

Consuming Datasets

BigML offers you several options to use your dataset out of the BigML Dashboard. When your data is ready, you can **export** it and **download** it to the Comma-Separated Values (SCV) format, and use it in your local environment, or download it to the Tableau Data Extract (TDE) format, so you can use it in Tableau[1], or use it programmatically via the **BigML API and bindings**. This section explains these three options.

9.1 Exporting and Downloading Datasets to CSV

To download your dataset from the BigML Dashboard:

1. From the **dataset view**, click the **1-click action menu**, and select REQUEST EXPORT (CSV). (See Figure 9.1.)

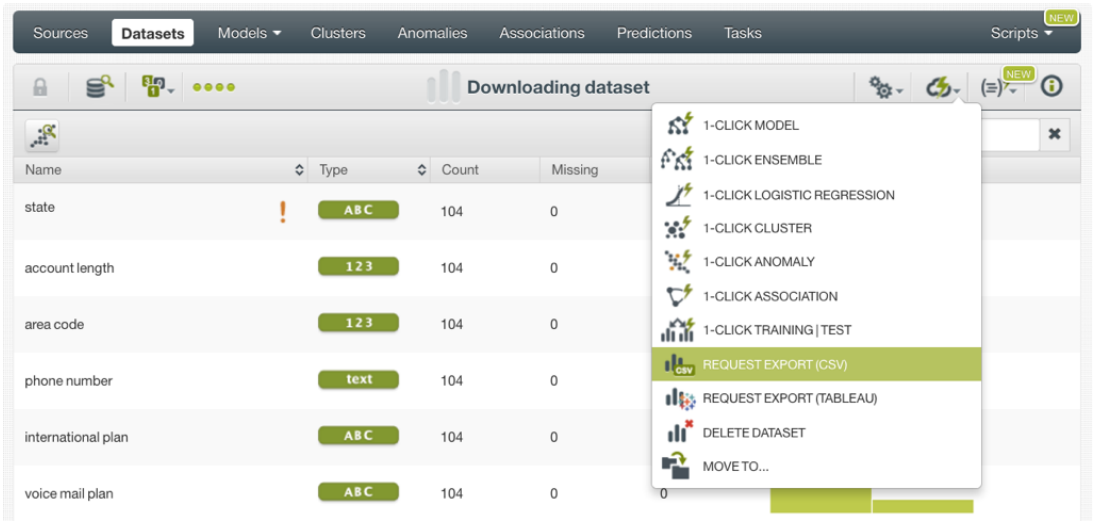

Figure 9.1: Request export to CSV format

2. BigML processes your request. **Note: the process may take a few minutes**. The larger the dataset size, the longer it will take.
3. Once the dataset is ready, select DOWNLOAD DATASET (CSV) from the same 1-click action menu, and save the dataset to your local environment (see Figure 9.2).

[1] http://www.tableau.com/

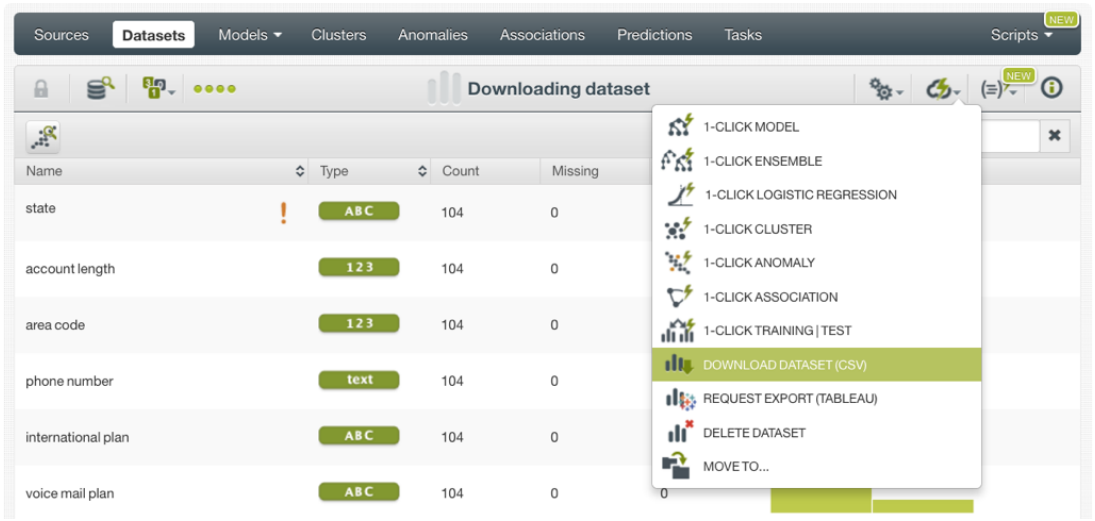

Figure 9.2: Download your dataset in CSV format

9.2 Exporting and Downloading Datasets to Tableau

To export your dataset from the BigML Dashboard to the **Tableau** format (TDE):

1. From the **dataset view**, click the **1-click action menu**, and select REQUEST EXPORT (TABLEAU). (See Figure 9.3.)

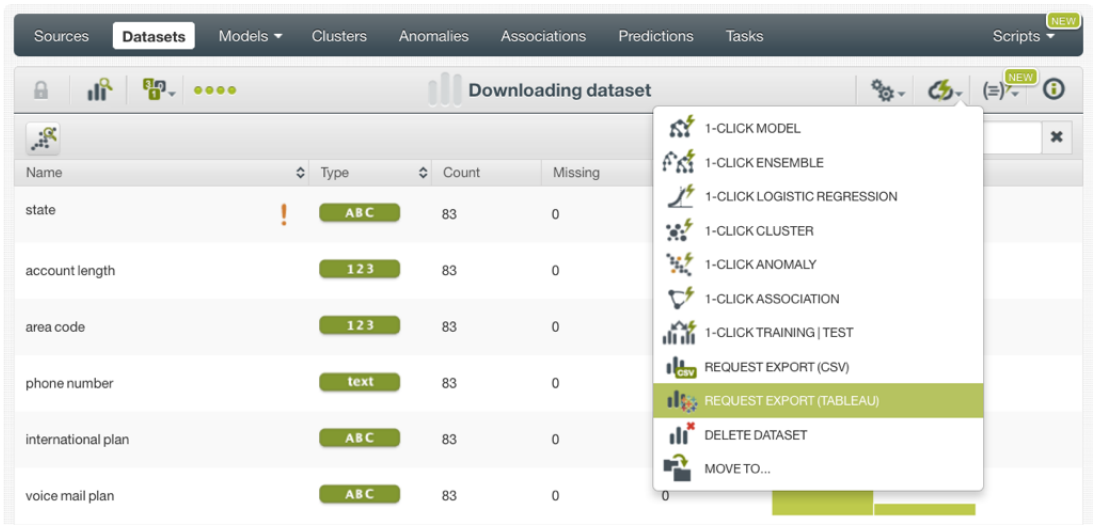

Figure 9.3: Request export to TDE format

2. BigML processes your request. **Note: the process may take a few minutes. The larger the dataset size, the longer it will take**.
3. Once the dataset is ready, select DOWNLOAD DATASET (TABLEAU) from the same 1-click action menu (see Figure 9.4), and save the TDE file in your local environment. This file is ready to be used in the Tableau platform.

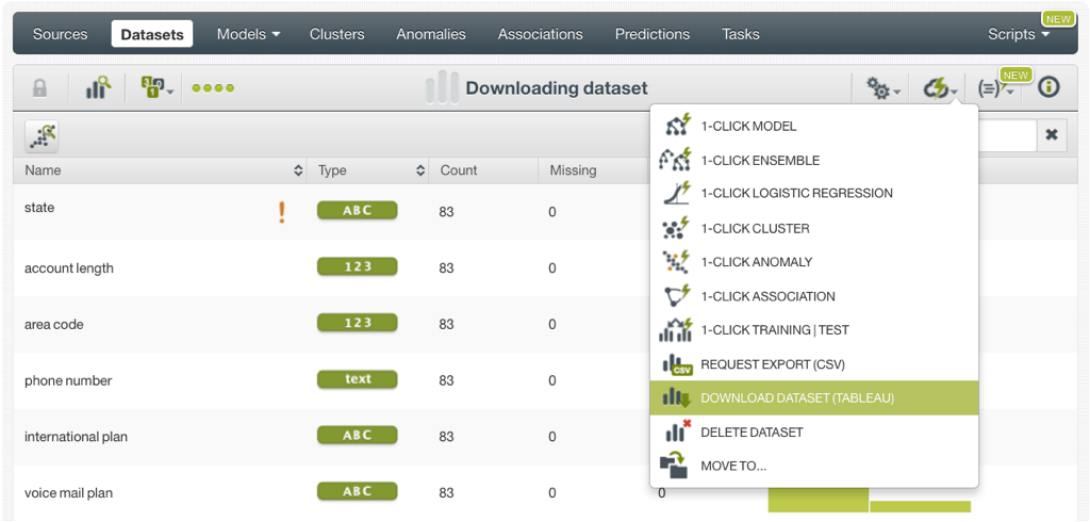

Figure 9.4: Download your dataset in TDE format

9.3 Using Datasets Via the BigML API

Datasets as well as all BigML resources can also be used through the BigML API, which allows you to programmatically create, update, list, delete, and use your dataset for models creation, and later make predictions with them. Create a dataset with custom values for a few available options, after properly setting the BIGML_AUTH environment variable to contain your authentication credentials:

```
curl "https://bigml.io/dataset?$BIGML_AUTH" \
     -X POST \
     -H 'content-type: application/json' \
     -d '{"source":"source/50a4527b3c1920186d000041", "name": "my dataset"}'
```

For more information on using datasets through the BigML API, please refer to the dataset REST API documentation[2].

9.4 Using Datasets Via the BigML Bindings

You can also create and use datasets via the **BigML bindings**, which are libraries aimed to make it easier to use the BigML REST API from your language of choice. BigML offers bindings for a number of languages, including: Python, Node.js, Java, Swift or Objective-C. You can find an example to create a dataset with the Python bindings below:

```
from bigml.api import BigML
api = BigML ()
dataset = api.create_dataset('source/50a4527b3c1920186d000041')
```

For more information on using the BigML Python bindings, please refer to the BigML Python bindings documentation[3].

[2] https://bigml.com/api/datasets
[3] http://bigml.readthedocs.io/en/latest/#creating-datasets

CHAPTER 10

Dataset Limits

Before creating your dataset you should consider the following limitations:

- **Fields**: there is no enforced limit to the number of fields that can be present in a dataset.
- **Instances**: there is no enforced limit to the number of instances that can be present in a dataset.
- **Classes**: a maximum number of 1,000 distinct classes per field is allowed.
- **Terms**: BigML can handle up to 1,000 terms total. If multiple text fields are defined, then the term limit per field is divided by the number of text fields, , e.g., a dataset with two text fields would result in 500 terms per text field. BigML selects those terms with most significant frequency, discarding both those that appear either too often or too infrequently. A maximum of 256 characters per term is allowed.
- **Items**: a maximum of 10,000 items per field is allowed.

If you need to exceed these limits, please contact the Support Team at BigML[1] and request your BigML Private Deployment[2].

[1] support@bigml.com
[2] https://bigml.com/private-deployments

CHAPTER 11

Descriptive Information

Each dataset has an associated **name**, **description**, **category**, and **tags**. A brief description follows for each concept. The MORE INFO menu option lets you edit this information. (See Figure 11.1.)

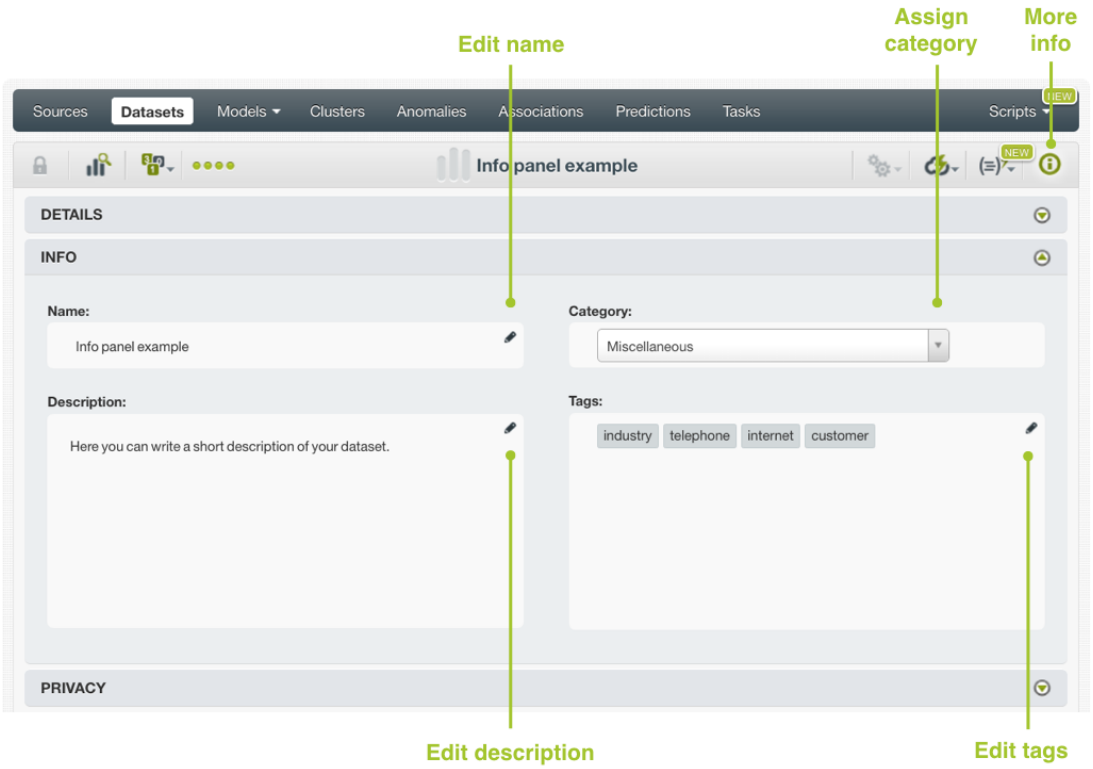

Figure 11.1: Panel to edit a dataset name, description, category and tags

11.1 Dataset Name

Each dataset has an associated **name** that is displayed on the listing and also on the header of a **dataset view**. Dataset name is indexed to be used in searches. When you create a dataset, it gets a default name. Change it using the MORE INFO menu option on the right corner of the **dataset view** (see Figure 11.1). The name of a dataset cannot be longer than **256** characters. There is no restriction on the characters that can be used in a dataset name. More than one dataset can have the same name, even within the same project. They will always have different identifiers.

11.2 Description

Each dataset also has a **description** that is very useful for documenting your Machine Learning projects. Descriptions can be written using plain text and also markdown[1]. BigML provides a simple markdown editor. (See Figure 11.2.)

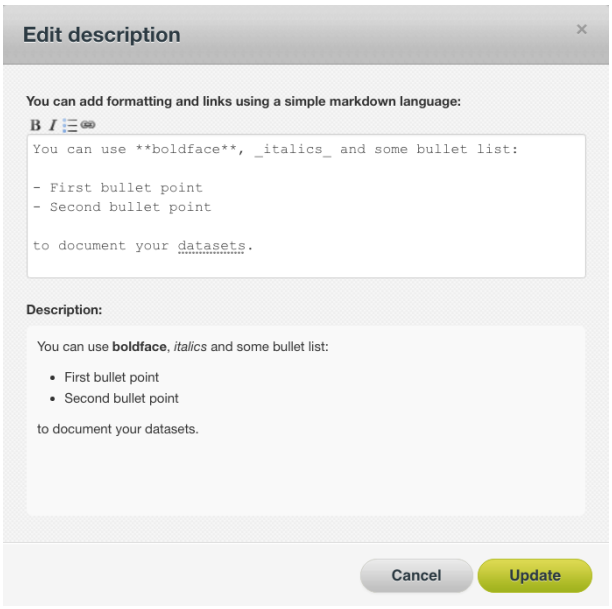

Figure 11.2: Markdown editor for datasets descriptions

Descriptions cannot be longer than **8192** characters and can use many charsets.

11.3 Category

Each dataset is associated with a **category**. Categories are useful to classify datasets according to the domain which your data comes from. This is useful when you use BigML to solve problems across industries or with multiple customers. A dataset category must be one of the categories listed on Table 11.1.

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

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

Table 11.1: Categories used to classify datasets by BigML

11.4 Tags

A dataset can also have a number of **tags** associated with it which can help retrieve the dataset via the BigML API or provide datasets with some extra information. Each tag is limited to a maximum of **128** characters. Each dataset can have up to **32** different tags.

11.5 Counters

For each dataset, BigML also stores a number of **counters** to track the number of other resources that have been created using it as a starting point. In the **dataset view** you can see a menu option that displays these counters (see Figure 11.3). It also allows you to quickly jump to all the resources of one type that have been created with this dataset.

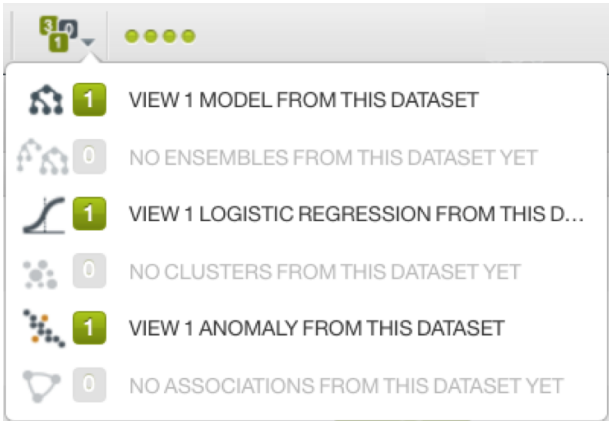

Figure 11.3: Menu option to quickly access to resources created with a dataset

CHAPTER 12

Dataset Privacy

Privacy options for a dataset can be defined in the MORE INFO menu option, displayed in Figure 12.1. There are three **levels of privacy** for BigML datasets:

- **Private**: only accessible by authorized users (the owner and those who have been granted access by him or her).
- **Shared**: accessible by any user with whom the owner shares a secret link.
- **Public**: accessible and clonable as private resources by any user. Public resources are listed in the BigML Gallery. If you want to let other BigML users make use of your dataset, please follow the steps in Section 13.2.

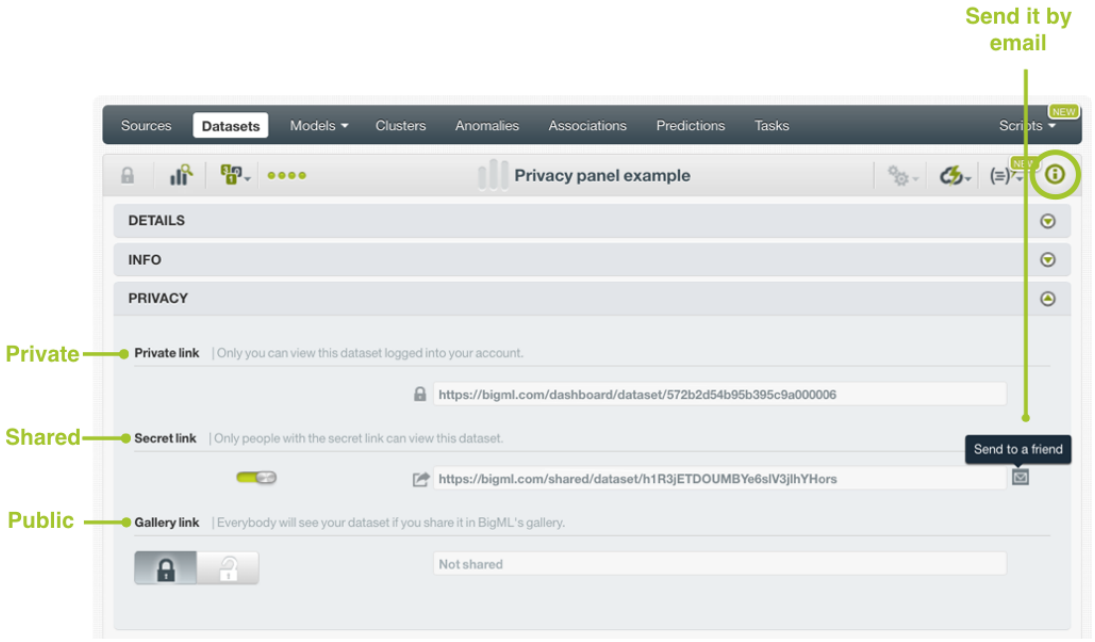

Figure 12.1: Datasets privacy options

CHAPTER 13

The BigML Gallery

The BigML Gallery[1] is a section of BigML to share, sell, buy or clone datasets, models, and scripts. The following subsections cover the dataset parts. Please read the terms of service[2] before you buy or sell any of these three resources.

13.1 Cloning a Dataset From the BigML Gallery

BigML lets you use datasets that are public in BigML Gallery. These datasets are public and available because other users have shared them. Some of the datasets available are free of charge and others have a specific cost. The owner of the dataset decides its cost. (See Section 13.2 for more details on how to share or sell your dataset.)

1. To import a dataset from the gallery into your Dashboard, first you need to clone it. The link that gives you access to BigML public Gallery is on the very top menu on the left. (See Figure 13.1.)

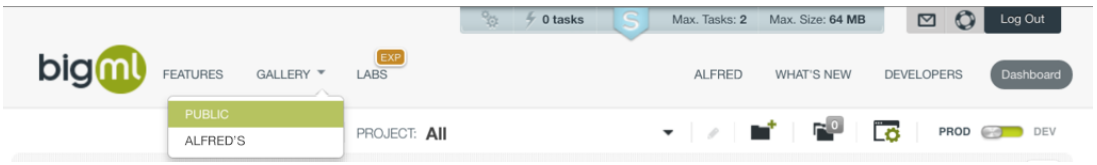

Figure 13.1: Access to BigML Gallery

2. Select "Datasets" on the top menu. Then click the dataset you are interested in. (See Figure 13.2.) Clone it by clicking the Buy label. If the dataset is free of charge, click the Free label, which changes to Buy once you mouse over it, but actually BigML will not charge you anything.

[1] https://bigml.com/gallery
[2] https://bigml.com/tos

Figure 13.2: BigML public gallery

3. A modal window (see Figure 13.3) will be displayed asking you for confirmation. Click the Clone button to confirm.

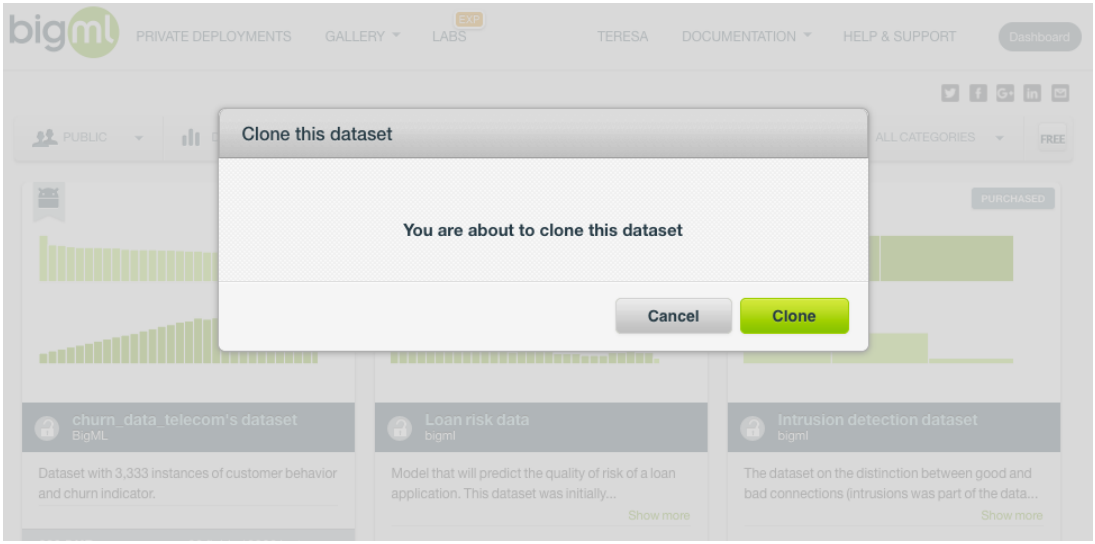

Figure 13.3: Modal window to confirm you want to clone this dataset

4. Your new dataset goes directly to your Dashboard. Notice that any task performed from a cloned dataset from the BigML Gallery (except predictions and evaluations) is free of charge, no matter the size of the task to be performed.

13.2 Publishing a Dataset in the BigML Gallery

You can make your dataset public for other BigML users. To accomplish this, put your dataset in the BigML Gallery:

1. We recommend you assign a proper name, category, and tags to the dataset you want to make public in the BigML Gallery. However, a description is mandatory. (See Chapter 11 to learn how to update your dataset descriptive information.)
2. By default your dataset will be private, since you haven not shared it yet. To share it, click the White Box icon. (See Figure 13.4.)

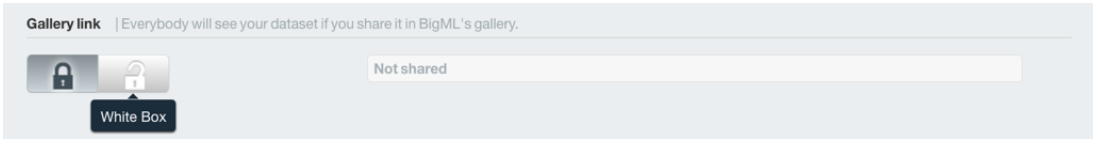

Figure 13.4: White Box lets you make your dataset public in the BigML Gallery

3. A modal window will automatically appear asking for confirmation. Decide whether to share your dataset for free or sell it. Set the price you consider appropriate by just moving the dataset price slider. (See Figure 13.5.)

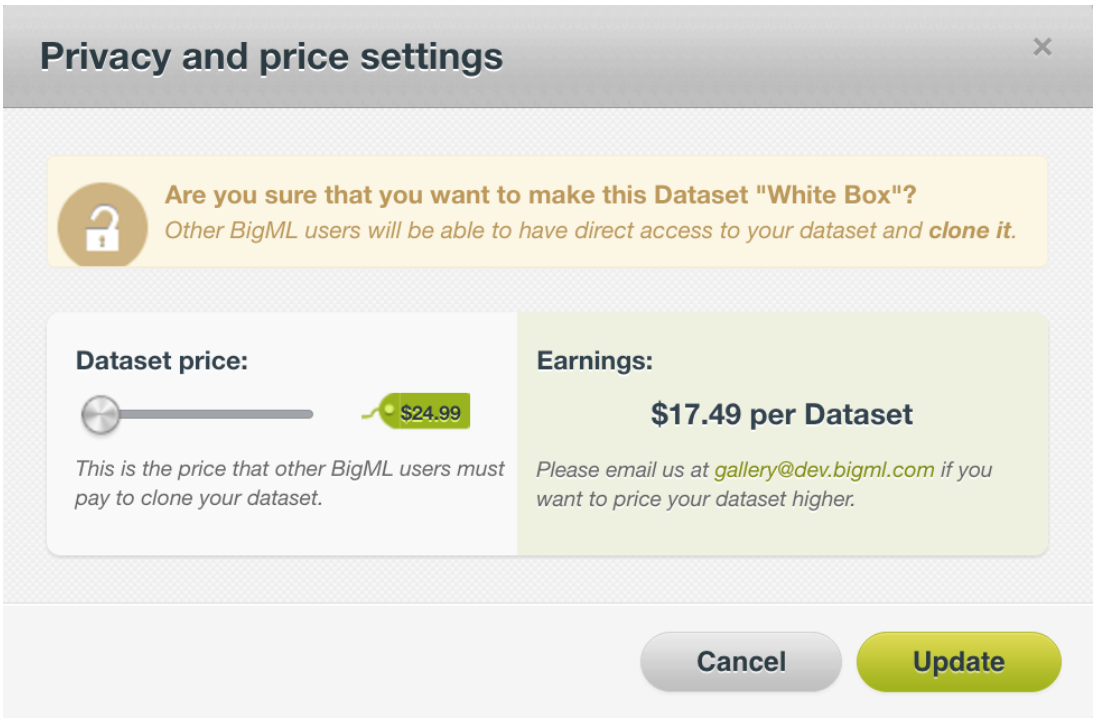

Figure 13.5: Make your dataset public for free or with earnings

4. Then, the gallery link automatically appears in the **privacy panel** and the status changes from "Private" to "White Box." You can change the set price anytime by clicking the edit icon. (See Figure 13.6.)

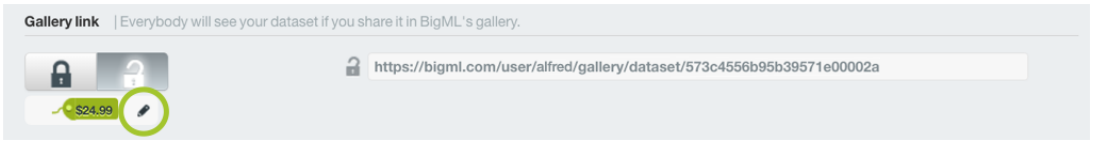

Figure 13.6: Public status changed to White Box and the gallery link is available

You can only share your own datasets. If you are using a dataset previously cloned, BigML will display a modal window (see Figure 13.7) stating you cannot share that dataset or sell it.

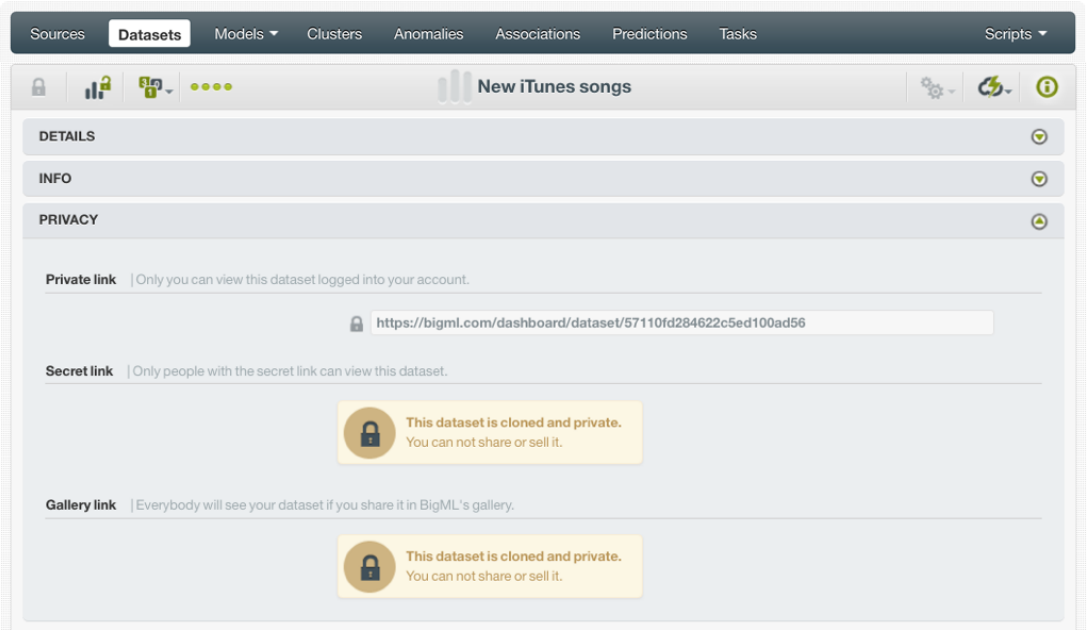

Figure 13.7: A cloned dataset cannot be shared or sold

CHAPTER 14

Moving a Dataset to Another Project

By default, when you create a dataset, it will be assigned to the project indicated on the project selector bar. (See Figure 14.1.) This dataset will be assigned to the same project where your source is (if your source is in a project). If you did not assign any project to the source you used to create your dataset, the new dataset will not be assigned to any project, and it will be shown when the project selector bar shows "**All**."

Figure 14.1: Project bar

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

1. In the **dataset list view**, within the **1-click action menu** for each dataset. (See Figure 14.2.)

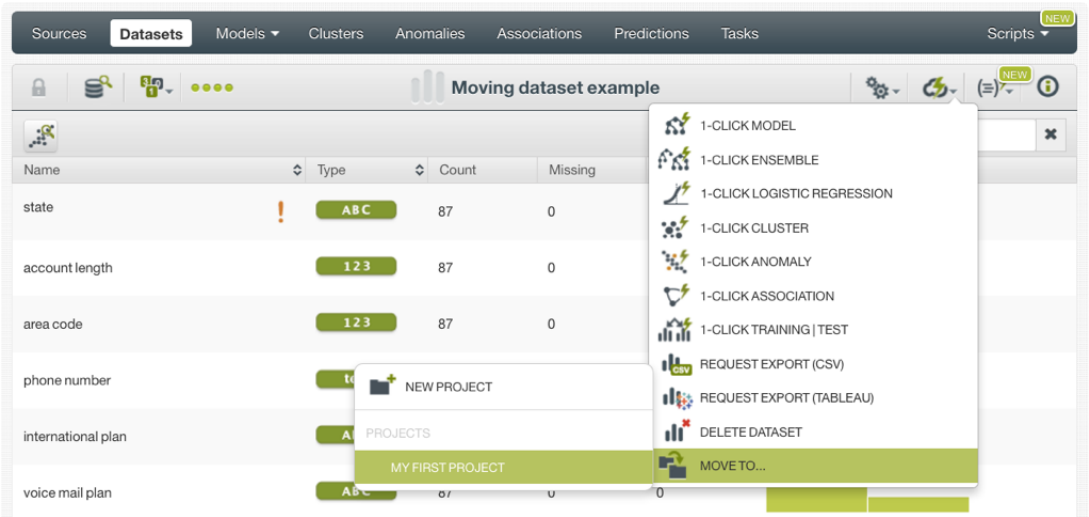

Figure 14.2: Menu option to move datasets

2. Within the **pop up menu** of a dataset in the **dataset list view**. (See Figure 14.3.)

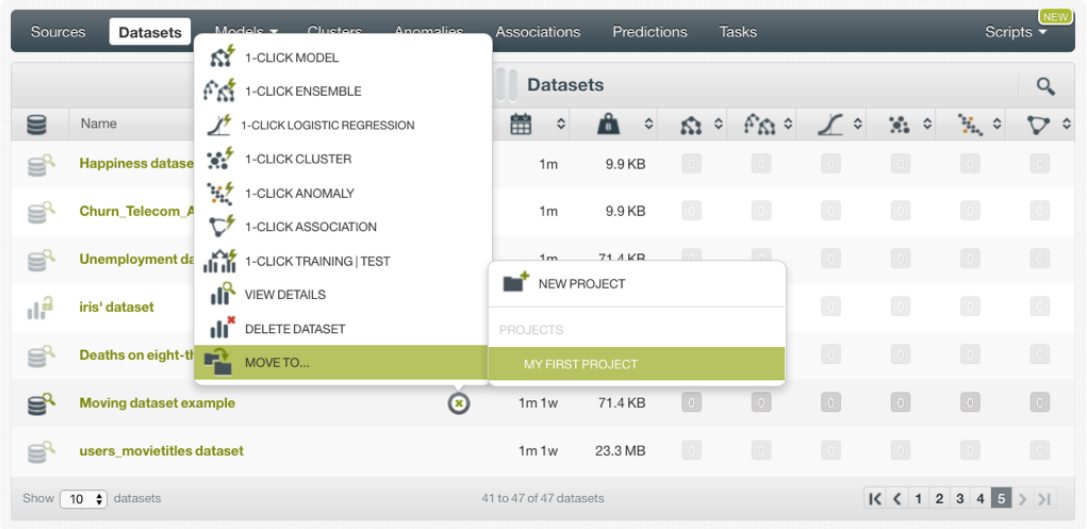

Figure 14.3: Menu option to move datasets with the pop up menu from the dataset list view

You can switch the working mode anytime by moving the slider displayed in Figure 14.4.

Figure 14.4: Top menu with switching modes slider

CHAPTER 15

Stopping Dataset Creation

BigML lets you stop a dataset creation before the task is finished. You can do this in two ways:

1. Select DELETE DATASET from the **1-click action menu** while BigML is processing your request. (See Figure 15.1.)

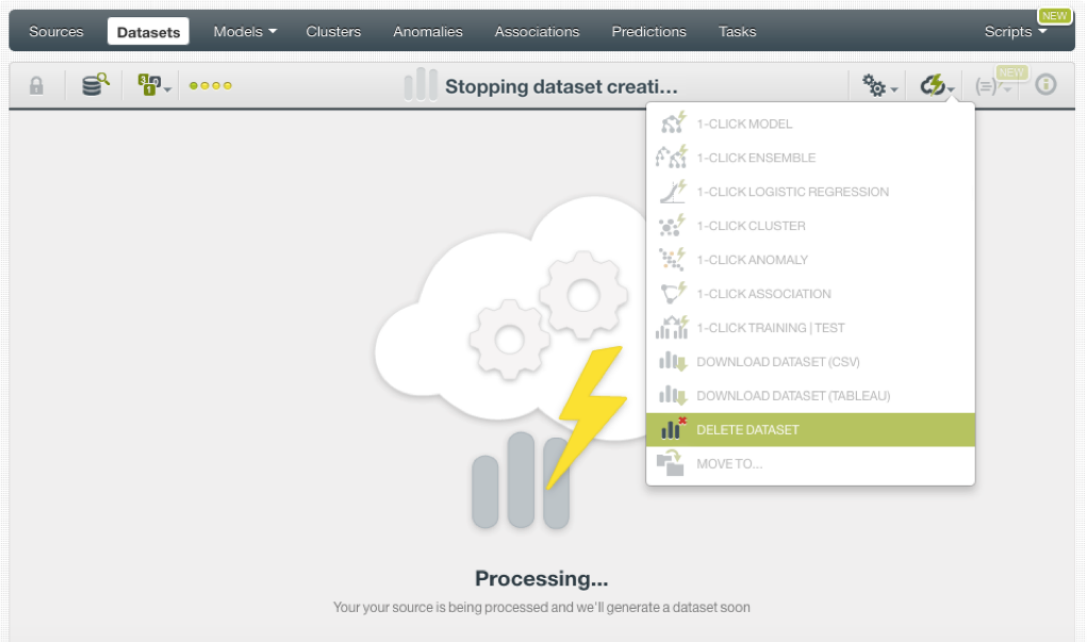

Figure 15.1: Stop your request from the 1-click action menu options

2. Or select DELETE DATASET from the **pop up menu** on the **dataset list view**. (See Figure 15.2.)

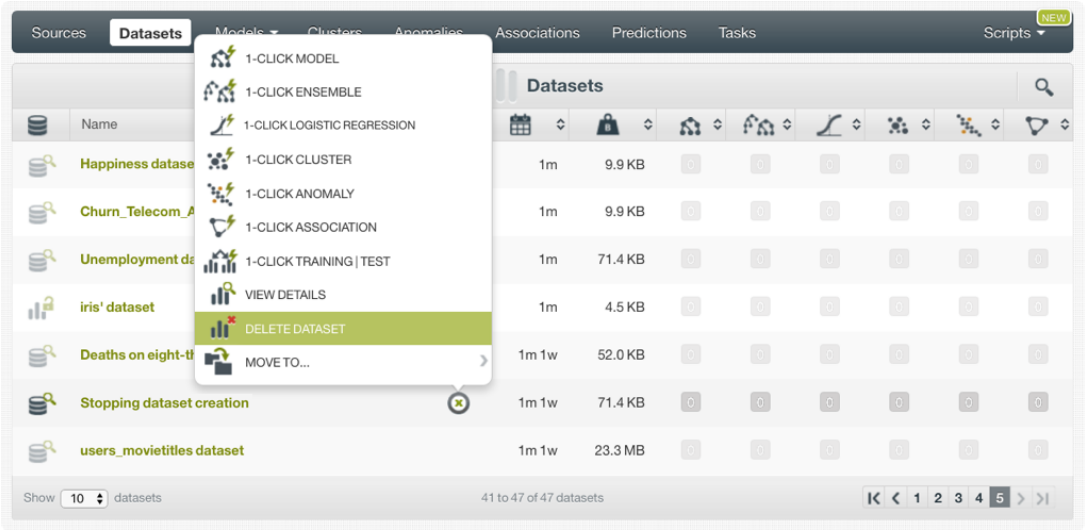

Figure 15.2: Stop your request from the pop up menu

In both cases, a modal window (see Figure 15.3) will be displayed asking you for confirmation.

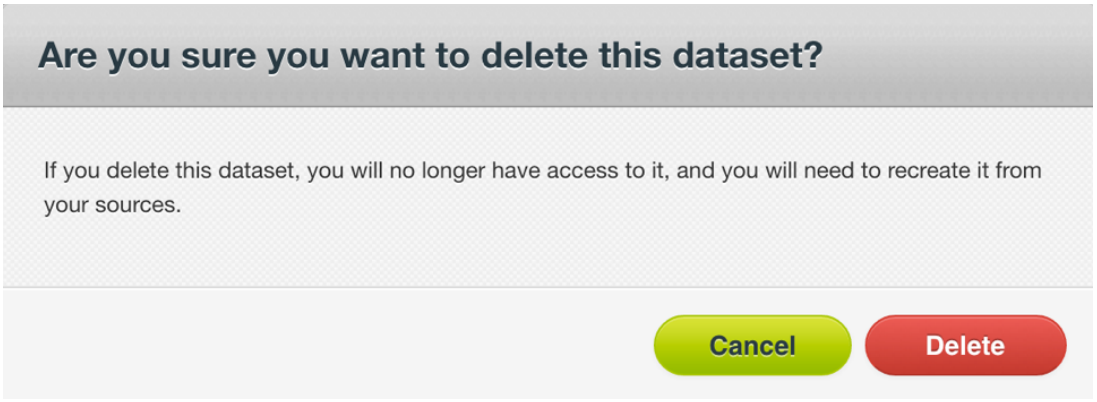

Figure 15.3: Confirmation window to stop the dataset creation process

The next section describes how to delete datasets once they have been created.

CHAPTER 16

Deleting Datasets

If you no longer need a dataset, BigML lets you delete it. You can delete your datasets in two ways:

1. Select DELETE DATASET from the **1-click action menu**. (See Figure 16.1.)

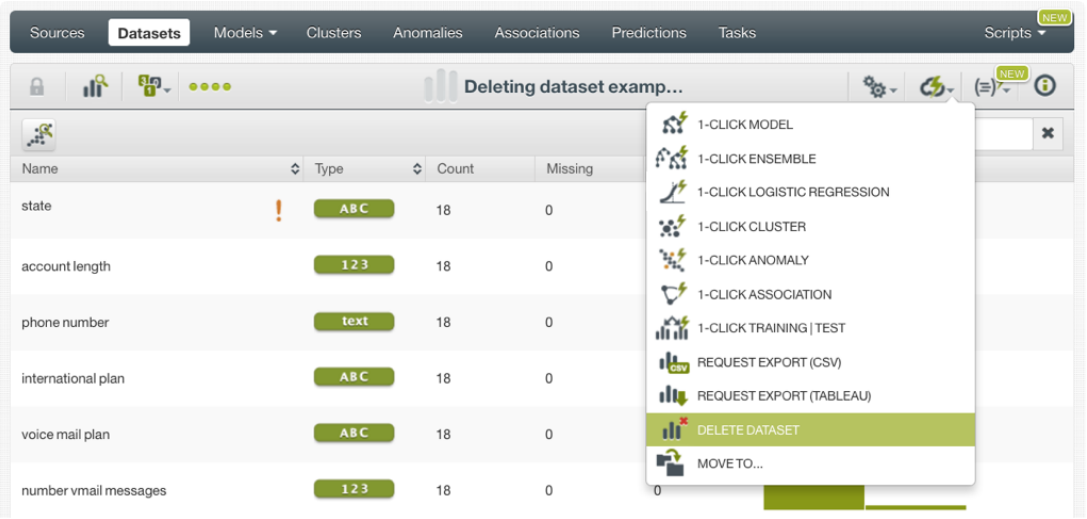

Figure 16.1: Menu option to delete a dataset

2. Or select DELETE DATASET from the **pop up menu** on the **dataset list view**. (See Figure 16.2.)

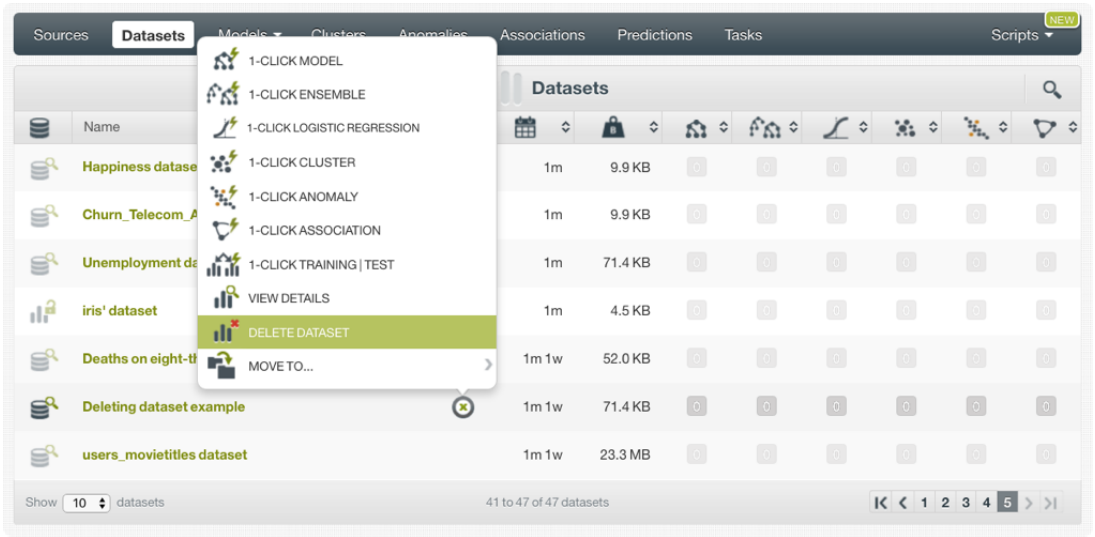

Figure 16.2: Delete dataset pop up menu option

In both cases, a modal window (see Figure 15.3) will be displayed asking you for confirmation. After you delete a dataset, it is deleted permanently, and there is no way you (or even the IT folks at BigML) can retrieve it.

Note: you cannot delete a dataset that is being used. BigML will display a modal window with the error message shown in Figure 16.3.

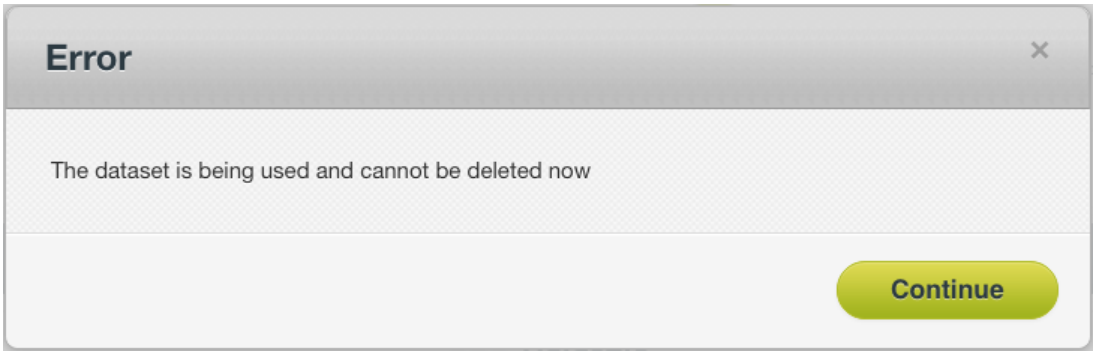

Figure 16.3: Modal window informing that you cannot delete a dataset that is being used

CHAPTER 17

Takeaways

This document explains datasets in detail. We finish it with a list of key points:

- A dataset is a structured version of your data. On the one hand, BigML computes some general statistics, and on the other hand, BigML computes

  statistics for each one of the fields.
- A dataset is created processing your source. BigML computes basic statistics per each field.
- You can create datasets from sources that have previously been uploaded to BigML.
- Datasets are the input to create a models, ensembles, logistic regressions, evaluations, clusters, anomalies and associations. (See Figure 17.1.)
- A model, a cluster, an anomaly, an association, a batch prediction (using models, ensembles, or logistic regressions), a batch centroid, or a batch anomaly score can produce a dataset as an output. (See Figure 17.2.)
- It is not required for a dataset be entirely loaded into memory for it to be processed.
- Often the transformations required for a dataset to optimally solve a given problem can be long, complex, and easy to get lost in. With BigML datasets, you do not risk losing track of the sequence of transformations you apply to your data.
- You can easily update field types after the dataset creation. You need to configure the source of your dataset and update the changes.
- You can create a dataset with just 1-click or select the size and the fields you want to include.
- You can transform your original dataset and create a new one by splitting your dataset in two different subsets, sampling it, filtering it, and adding new fields to your dataset. (See Figure 17.3.)
- The non-preferred fields and the objective field are inherited when you split your dataset in two subsets, when you sample it, filter it, or add new fields to your dataset. Also when you clone it from the BigML Gallery.
- You can use the Flatline editor to perform powerful transformations with your dataset.
- You can export and download your dataset to CSV format to use it in your local environment.
- You can export and download your dataset to TDE format to use it in Tableau platform.
- You can programmatically create, list, delete, and use your dataset for models creation, and later make predictions with them through the BigML API and the BigML bindings.
- You can furnish your dataset with descriptive information (name, description, tags, and category) and also every individual field (name, label, and description).
- There are three levels of privacy for BigML datasets: private, shared and public.
- You can clone an existing dataset from BigML Gallery.

- You can share your dataset in the BigML Gallery, either for free or with earnings.
- You can only assign a dataset to a specific project.
- You can move a dataset between projects.
- You can stop the dataset creation.
- You can permanently delete a dataset.

Figure 17.1: Using a dataset as the input to create your resources

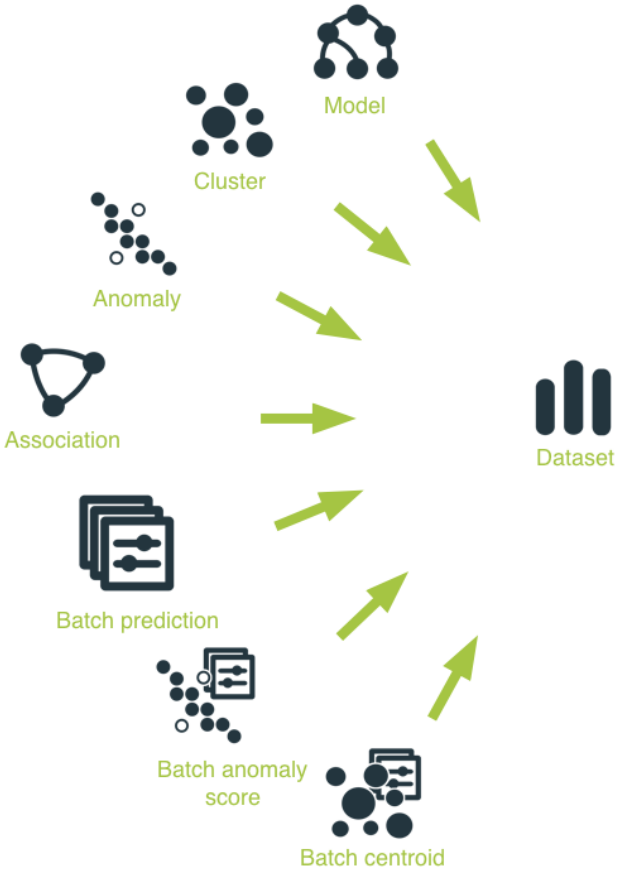

Figure 17.2: Resources that produce a dataset as output

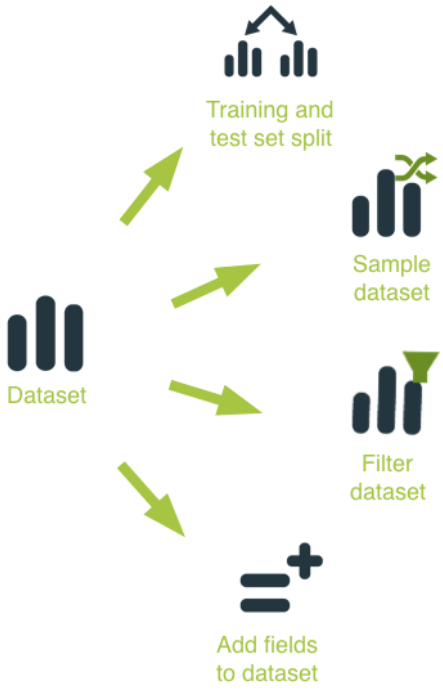

Figure 17.3: Creating new datasets from your original dataset

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

List of Figures

List of Tables

Glossary

Anomaly Detection an unsupervised Machine Learning task which identifies instances in a dataset that do not conform to a regular pattern. ii, 96

Association Discovery an unsupervised Machine Learning task to find out relationships between values in high-dimensional datasets. It is commonly used for market basket analysis. ii, 96

BigML Gallery a section of BigML to share, buy or sell datasets, models, and scripts. Go to Gallery. 1, 19, 86, 87, 96

Classification a modeling task whose objective field (i.e., the field being predicted) is categorical and predicts classes. ii, 27

Clustering an unsupervised Machine Learning task in which dataset instances are grouped into geometrically related subsets. ii, 96

Dashboard The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. ii, 1, 86

Data Wrangling the process of converting or mapping data from one “raw” form into another format that allows a more convenient use of the data. 1

Dataset the structured version of a BigML source. It is used as input to build your predictive models. For each field in your dataset a number of basic statistics (min, max, mean, etc.) are parsed and produced as output. ii, 1, 65, 70, 86, 96

Discretization the process of transforming a numeric field into a categorical field. 51

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. 96

Evaluation a resource representing an assessment of the performance of a predictive model. 96

Feature Engineering the process of generating new features for a dataset so that Machine Learning algorithms will be more effective on that data. The features can either be transformations of existing features or entirely new information. 49

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). 96

Flatline a domain-specific lisp-like language that allows you to perform an infinite number of operations to create new fields or filter your BigML datasets. Furthermore, with the Flatline Editor you will be able to validate your Flatline expressions and preview the results from your Dashboard. 31, 49, 58, 96

Histogram a bar chart-style visualization of a collection of values, in which the range of the values is broken up into a collection of ranges, and the height of a given bar increases as more points fall into the range associated with that bar. 3

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. 96

Model a single decision tree-like model when we refer to it in particular, and a predictive model when we refer to it in general. 86, 96

Non-preferred fields fields that, for a number of possible reasons, are by default not included in the modeling process. One example of this is fields that contain the same value for every instance; in general, constant fields add no information to the modeling process. 19, 96

Objective Field the field that a regression or classification model will predict (also known as target). 19, 96

Predictive Model a machine-learned model that has been created using statistical learning. It can help describe or infer some statistical properties of an entity using the instances provided by a dataset. ii

Regression a modeling task whose objective field (i.e., the field being predicted) is numeric. ii, 27

Resource any of the Machine Learning objects provided by BigML that can be used as a building block in the workflows needed to solve Machine Learning problems. 18, 79, 86

Script a compiled source code written in WhizzML for automating Machine Learning workflows and implementing high-level algorithms. 86

Source the BigML resource that represents the data source to which you wish to apply Machine Learning. A data source stores an arbitrarily-large collection of instances. A BigML source helps you ensure that your data is parsed correctly. The BigML preferred format for data sources is tabular data in which each row is used to represent one of the instances, and each column is used to represent a field of each instance. 1, 65, 70

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

Tag cloud a visualization of a text field in which each term is sized according to the number of instances in which it appeared in that field. 5

Task the process of creating a BigML resource, such as creating a dataset, or training a model. A given task can also create subtasks, as, in the case of a WhizzML script that contains calls to create other resources. 18, 87

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. 73

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. *Sources with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.
[6] The BigML Team. *Time Series with the BigML Dashboard*. Tech. rep. BigML, Inc., July 2017.
[7] The BigML Team. *Topic Models with the BigML Dashboard*. Tech. rep. BigML, Inc., Nov. 2016.


bigml