

# Time Series with the BigML Dashboard

The BigML Team

Version 2.1



MACHINE LEARNING MADE BEAUTIFULLY SIMPLE

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

[info@bigml.com](mailto:info@bigml.com)

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

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

This work by BigML, Inc. is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/). Based on work at <http://bigml.com>.

*Last updated March 27, 2024*

# About this Document

This document provides a comprehensive description of how to perform time series analysis using the BigML [Dashboard](#). Learn how to use the BigML Dashboard to configure, visualize, and interpret this [supervised](#) model and use it to make forecasts.

This document assumes that you are familiar with:

- [Sources with the BigML Dashboard](#). The BigML Team. June 2016. [5]
- [Datasets with the BigML Dashboard](#). The BigML Team. June 2016. [4]

To learn how to use the BigML Dashboard to build other [supervised](#) predictive models read:

- [Classification and Regression with the BigML Dashboard](#). [2]

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

- [Cluster Analysis with the BigML Dashboard](#). The BigML Team. June 2016. [3]
- [Association Discovery with the BigML Dashboard](#). The BigML Team. June 2016. [1]
- [Topic Modeling with the BigML Dashboard](#). The BigML Team. November 2016. [6]

# Contents

- 1 Introduction** **1**
  
- 2 Understanding Time Series** **3**
  - 2.1 Time Series Data . . . . . 3
  - 2.2 Exponential Smoothing Methods . . . . . 3
  - 2.3 Model Performance Metrics . . . . . 5
  
- 3 Creating Time Series with 1-Click** **7**
  
- 4 Time Series Configuration Options** **9**
  - 4.1 Objective Fields . . . . . 9
  - 4.2 Dates . . . . . 10
  - 4.3 Default Numeric Value . . . . . 11
  - 4.4 Forecast Horizon . . . . . 12
  - 4.5 Model Components . . . . . 12
    - 4.5.1 Error . . . . . 13
    - 4.5.2 Trend and Damping Parameters . . . . . 13
    - 4.5.3 Seasonality and Period Length . . . . . 13
  - 4.6 Sampling . . . . . 14
  - 4.7 Creating Time Series with Configured Options . . . . . 14
  - 4.8 API Request Preview . . . . . 15
  
- 5 Visualizing Time Series** **17**
  - 5.1 Time Series Decomposition . . . . . 21
  
- 6 Time Series Evaluations** **24**
  - 6.1 Introduction . . . . . 24
  - 6.2 Creating Time Series Evaluations . . . . . 26
  - 6.3 Configuring Time Series Evaluations . . . . . 28
    - 6.3.1 Fields Mapping . . . . . 28
    - 6.3.2 Range . . . . . 29
  - 6.4 Visualizing Time Series Evaluations . . . . . 30
    - 6.4.1 Evaluation Chart . . . . . 30
    - 6.4.2 Evaluation Metrics . . . . . 34
  - 6.5 Consuming Time Series Evaluations . . . . . 36
    - 6.5.1 Using Time Series Evaluations Via the BigML API . . . . . 36
    - 6.5.2 Using Time Series Evaluations Via the BigML Bindings . . . . . 36
  - 6.6 Deleting Time Series Evaluations . . . . . 36
  
- 7 Time Series Forecasts** **39**
  - 7.1 Introduction . . . . . 39
  - 7.2 Creating and Configuring Forecasts . . . . . 39
  - 7.3 Visualizing Forecasts . . . . . 39

|  |           |
|--|-----------|
| 7.4 Consuming Forecasts . . . . .                      | 41        |
| 7.4.1 Using Forecasts Via the BigML API . . . . .      | 41        |
| 7.4.2 Using Forecasts Via the BigML Bindings . . . . . | 42        |
| 7.5 Deleting Forecasts . . . . .                       | 42        |
| <b>8 Consuming Time Series</b>                         | <b>43</b> |
| 8.1 Using Time Series Via the BigML API . . . . .      | 43        |
| 8.2 Using Time Series Via the BigML Bindings . . . . . | 43        |
| <b>9 Time Series Limits</b>                            | <b>44</b> |
| <b>10 Time Series Descriptive Information</b>          | <b>45</b> |
| 10.1 Time Series Name . . . . .                        | 45        |
| 10.2 Description . . . . .                             | 45        |
| 10.3 Category . . . . .                                | 46        |
| 10.4 Tags . . . . .                                    | 47        |
| 10.5 Counters . . . . .                                | 47        |
| <b>11 Time Series Privacy</b>                          | <b>49</b> |
| <b>12 Moving Time Series</b>                           | <b>50</b> |
| <b>13 Stopping Time Series Creation</b>                | <b>52</b> |
| <b>14 Deleting Time Series</b>                         | <b>54</b> |
| <b>15 Takeaways</b>                                    | <b>56</b> |
| <b>List of Figures</b>                                 | <b>59</b> |
| <b>List of Tables</b>                                  | <b>61</b> |
| <b>Glossary</b>  | <b>62</b> |
| <b>References</b>                                      | <b>64</b> |

# Introduction

Time series is a sequentially indexed representation of your historical data that can be used to **forecast** future values of numerical properties. It is used to analyze time-based data when historical patterns can explain the future behavior such as sales forecasting, website traffic, production and inventory analysis or weather forecasting among other use cases. A **time series** model needs to be trained with time series data, i.e., a **numeric field** containing a **time-ordered sequence of regularly spaced data points data points in time**.

BigML time series is an optimized implementation of the **exponential smoothing**<sup>1</sup> methods. The smoothing parameters assign exponentially increasing weights to more recent instances. In contrast with other methods, past instances are not equally weighted, recent instances are given more weight than older instances. Exponential smoothing methods also allow the modelization of data that contains **trend and seasonal patterns**. Learn more about exponential smoothing methods in **Chapter 2**.

This document provides a comprehensive description of BigML time series, including how they can be created with 1-click (**Chapter 3**), how to configure the different parameters (**Chapter 4**), and the visualization in a time series chart provided by BigML (**Chapter 5**). Once your time series has been created, you need to evaluate it to ensure that it makes accurate forecasts (**Chapter 6**). When you are satisfied with your time series performance, you can use it to forecast (see **Chapter 7**). You can also create, configure, retrieve, list, update, delete, evaluate and use your time series to make forecasts using the BigML API and bindings (**Section 8.1** and **Section 8.2**).

The TIME SERIES section within the “Supervised” tab of the main menu of the BigML **Dashboard** allows you to list all your available time series models. The time series list view (**Figure 1.1**), shows the dataset used to create each time series, the **Name** and the parameters configured to create the time series, the **objective fields** (explained in **Section 4.1**), **Age** (time elapsed since it was created), **Size**, and number of **Evaluations** and **Forecasts** that have been created using that time series. The SEARCH menu option in the top right menu of the time series list view allows you to **search** your time series by name and by configuration options.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Exponential\\_smoothing](https://en.wikipedia.org/wiki/Exponential_smoothing)

| Name  | Objective/s                                  |          |        |   |   |
|---|--|----------|--------|---|---|
| perrin-freres-monthly-champagne-<br>period=1, range=[1, 105]      | Perrin Freres monthly champagne sales...     | 0min     | 1.8 KB | 0 | 0 |
| _births_and_deaths   Training (80.0% - linear)<br>range=[1, 42]   | Male Live Births                             | 0min     | 1.2 KB | 0 | 0 |
| bt-1-births-by-sex-scotland-1855-miss<br>period=1, range=[1, 163] | Females Births                               | 1h 52min | 5.1 KB | 0 | 0 |
| monthly-gasoline-demand-ontario- v3<br>period=12, range=[1, 192]  | Monthly gasoline demand Ontario gallo...     | 9h 2min  | 3.5 KB | 0 | 0 |
| monthly-gasoline-demand-ontario- v2<br>period=12, range=[1, 156]  | Monthly gasoline demand Ontario gallo...     | 1d       | 3.5 KB | 0 | 1 |
| Monthly Min Tem Melbourne - 80% v1<br>range=[1, 100]              | Monthly-avg-temp-Melbourne                   | 1d       | 1.9 KB | 0 | 1 |
| Monthly Min Tem Melbourne - 80%<br>period=12, range=[1, 100]      | Monthly-avg-temp-Melbourne                   | 1d       | 1.9 KB | 0 | 0 |
| international-airline-passengers -miss<br>range=[1, 144]          | International airline passengers: monthly... | 1d       | 1.9 KB | 0 | 0 |
| monthly-gasoline-demand-ontario- v1<br>period=12, range=[1, 156]  | Monthly gasoline demand Ontario gallo...     | 1d       | 3.5 KB | 0 | 2 |
| monthly-gasoline-demand-ontario-<br>period=12, range=[1, 192]     | Monthly gasoline demand Ontario gallo...     | 1d 1h    | 3.5 KB | 0 | 0 |

Figure 1.1: Time series list view

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

| Name           | Objective/s |  |  |  |  |
|----------------|-------------|--|--|--|--|
| No time series |             |  |  |  |  |

Figure 1.2: Time series empty list view in the BigML Dashboard

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



Figure 1.3: Time series icon

# Understanding Time Series

A **time series** model is a supervised learning method to **forecast** the future values of a numeric field based on its previously observed values. A time series model needs to be trained with time series data, i.e., a field containing a time-ordered sequence of regularly spaced data points. BigML time series implement exponential smoothing methods to model data with different patterns, trends or seasonality. See below an explanation about how BigML interprets **time series data** (Section 2.1), the **exponential smoothing methods** that result in the different time series **models** (Section 2.2), and the **metrics** to measure the models performance (Section 2.3).

## 2.1 Time Series Data

A time series field is a numeric field which is time-ordered and regularly spaced. Temperatures, sales, production data, usage logs and website traffic, demographic and economic data (such as the number of births, GDP growth, unemployment rates, etc.) are typically serialized data.

When you create a time series model, BigML **assumes** your dataset contains time series data, i.e., the instances of the objective field are **sequentially arranged** in time and **regularly spaced**. This means that BigML will take the first instance in your dataset as the first data point to train the time series model, the second instance as the second data point and so on until the last instance in your dataset. It will also assume equal time intervals between your instances.

Missing values in your dataset will be replaced using **spline interpolation**<sup>1</sup> by default, but you can also select to replace them by zero, the mean, median, maximum or minimum of your field (see Section 4.3).

## 2.2 Exponential Smoothing Methods

BigML implements **exponential smoothing**<sup>2</sup> to train the time series models. Time series data is modeled as a **level** component and it can optionally include a **trend** (**damped** or not damped) and a **seasonality** component as explained below:

- The basic component of any **forecast equation** (Equation 2.1) is the **level equation** (Equation 2.2) which can be defined as the weighted average of the objective field past values.  $\alpha$  is the smoothing parameter of the level component and it can take values between 0 and 1. If  $\alpha$  is large, more weight is given to most recent observations.

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t \quad (2.1)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)l_{t-1} \quad (2.2)$$

<sup>1</sup>[https://en.wikipedia.org/wiki/Spline\\_interpolation](https://en.wikipedia.org/wiki/Spline_interpolation)

<sup>2</sup>[https://en.wikipedia.org/wiki/Exponential\\_smoothing](https://en.wikipedia.org/wiki/Exponential_smoothing)



The level equation can be re-arranged as a function of the forecast **error** ( $\epsilon$ ). The error represents the unpredictable variations in the time series data, and how they influence observed values. The error can be **additive** (see Equation 2.3) or **multiplicative** (see Equation 2.4) in which case the error will be relative to the level component. The error type mostly affects forecast **intervals**, and multiplicative error is only suitable for strictly positive data.

$$\text{Additive Error} \quad l_t = \alpha y_t + (1 - \alpha)l_{t-1} = l_{t-1} + \alpha(y_t - l_{t-1}) = l_{t-1} + \alpha\epsilon \quad (2.3)$$

$$\text{Multiplicative Error} \quad l_t = l_{t-1} + \alpha l_{t-1}\epsilon = l_{t-1}(1 + \alpha\epsilon) \quad (2.4)$$

- Data with a trend can be modeled by adding a **trend equation** (Equation 2.7) to the forecast formula (Equation 2.5), where  $b_t$  is the trend or slope of the time series at time  $t$ .  $\beta$  is the smoothing parameter for the trend and it can take values between 0 and 1. The trend component has two variations: **additive** or **multiplicative**. The additive variation is commonly known as **Holt's linear method**<sup>3</sup> (see Equation 2.5) because the trend growth is assumed to be linear, i.e. the trend slope is constant.

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t + hb_t \quad (2.5)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (2.6)$$

$$\text{Trend Equation} \quad b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (2.7)$$

The forecast equation with multiplicative trend is known as **exponential trend method** (see Equation 2.8) because the trend growth is exponential, i.e., it contains a constant growth rate that is multiplied by the level component.

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t b_t^h \quad (2.8)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (2.9)$$

$$\text{Trend Equation} \quad b_t = \beta \frac{l_t}{l_{t-1}} + (1 - \beta)b_{t-1} \quad (2.10)$$

- Trend models assume the trend will be repeated in the future indefinitely, but this does not usually happen in the real world. In cases in which you need to forecast a long time horizon ahead, a **damped parameter** can be included to “dampen” the trend to a flat line at some point in the future. The damping parameter is represented by a  $\phi$  coefficient in the forecast formula (see Equation 2.11).

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t + (\phi + \phi^2 + \dots + \phi^h)b_t \quad (2.11)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1}) \quad (2.12)$$

$$\text{Trend Equation} \quad b_t = \beta(l_t - l_{t-1}) + (1 - \beta)\phi b_{t-1} \quad (2.13)$$

The damped parameter can also be included when the trend is multiplicative resulting in a **multiplicative damped trend method** (see Equation 2.14). This method will generate more aggressive forecasts than the additive method but it will also produce a constant line in the long-term.

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t b_t^{(\phi + \phi^2 + \dots + \phi^h)} \quad (2.14)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}^\phi) \quad (2.15)$$

$$\text{Trend Equation} \quad b_t = \beta \frac{l_t}{l_{t-1}} + (1 - \beta)b_{t-1}^\phi \quad (2.16)$$

<sup>3</sup>[https://en.wikipedia.org/wiki/Exponential\\_smoothing#Basic\\_exponential\\_smoothing](https://en.wikipedia.org/wiki/Exponential_smoothing#Basic_exponential_smoothing)

- Finally, to model **seasonal data**<sup>4</sup>, i.e., data with fixed and regular periods or fluctuations, a **seasonality component** can be included in the forecast equation. The smoothing parameter of the seasonal component is  $\gamma$  and it takes values between 0 and 1. BigML will try to automatically learn the seasonality patterns in your data or you can manually configure the period length ( $m$ ), i.e., the number of data points per period (see **Subsection 4.5.3**). The seasonal component can be **additive** or **multiplicative**. The **Equation 2.21** shows the forecast equation with additive seasonality:

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = l_t + hb_t + s_{t-m+h_m^+} \quad (2.17)$$

$$\text{Level Equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (2.18)$$

$$\text{Trend Equation} \quad b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (2.19)$$

$$\text{Seasonal Equation} \quad s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (2.20)$$

The multiplicative method shown in **Equation 2.21** makes the seasonal variations proportional to the level of the series:

$$\text{Forecast Equation} \quad \hat{y}_{t+h} = (l_t + hb_t)s_{t-m+h_m^+} \quad (2.21)$$

$$\text{Level Equation} \quad l_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (2.22)$$

$$\text{Trend Equation} \quad b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (2.23)$$

$$\text{Seasonal Equation} \quad s_t = \gamma \frac{y_t}{(l_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m} \quad (2.24)$$

The error, the trend (damped and not damped) and the seasonality components along with their variations (multiplicative or additive) can be **combined** producing thirty different forecast equations. Each one of the different combinations is called a **model** in the BigML Dashboard (“ets\_model” in the API) and you can see them all in your time series view (**Chapter 5**). However, not all combinations are explored due to numeric stability reasons. The following **combinations** are **excluded**:

- **Additive errors** with **multiplicative trends or seasonalities**.
- **Multiplicative trends** with **additive seasonality**.
- **Multiplicative seasonality** if your objective field contains **zeros or negative values**.

Usually, if your data does not contain seasonality, you will get eight different models. If your data contains seasonality components you will get up to nineteen different models.

Each model has a unique **name** associated based on its components variations following the schema **[Error, Trend, Seasonality]**. Each component can be represented by an “A” indicating the component is additive, an “M” multiplicative and an “N” indicating the component has not been included. Additionally, the trend component can be represented by an “Ad” or “Md” indicating damped trends. Thus, a model with the name [M,Ad,N] indicates the model has multiplicative errors, damped additive trend, and no seasonality.

If you want to learn more about the technicalities and actual formulas for each combination please visit [this page](#)<sup>5</sup>.

Each **model** learned from the data to fit the objective field has a set of **performance metrics** associated so you can select the best one. See an explanation of these metrics in the following section (**Section 2.3**).

## 2.3 Model Performance Metrics

For each model result of the multiple possible combinations of the components explained in **Section 2.2**, BigML provides a set of metrics so you can select the **best performing model**.

<sup>4</sup><https://en.wikipedia.org/wiki/Seasonality>

<sup>5</sup><https://www.otexts.org/fpp/7/1>

**Note: the performance metrics explained in this section are different from the metrics provided during the time series evaluation (see [Chapter 6](#)). The only common metric with evaluations is the R square.**

BigML provides four different metrics to measure your models' performance:

- **AIC**<sup>6</sup>: the Akaike Information Criterion measures the **trade-off** between the **goodness-of-fit** and the model's **complexity**. More complex models tend to better fit the training data but they generalize worse with data that the model has not seen before, resulting in less accurate predictions. This problem is known as **overfitting** and it is a common problem shared by all supervised learning methods. The AIC tries to minimize the errors in the training data as well as the risk of overfitting. AIC measures the errors made by the model by aggregating the differences between the model values and the original data. The model complexity is a function of the number of the model parameters. The smaller the AIC, the better.
- **AICc**: the **Corrected Akaike Information Criterion**<sup>7</sup> introduces a correction element for smaller datasets. When the number of instances is relatively small the AIC overvalues models that overfit the training data, therefore, the AICc is a better metric in this case. For bigger datasets, the AIC and AICc tend to yield similar results.
- **BIC**<sup>8</sup>: the Schwarz Bayesian Information Criterion is similar to the AIC, but it penalizes more heavily model complexity. The lower the BIC, the better.
- **R squared**<sup>9</sup>: the R squared measures the model's errors compared to the objective field actual values. The R squared can be interpreted as the model's goodness-of-fit to the training data. It can take values up to 1. The higher the R squared, the better. A downside of the R squared compared to the previously defined metrics is that it does not measure the model's complexity. This means that the AIC is usually a more accurate measure to evaluate the model's performance. Please refer to [this article](#)<sup>10</sup> to get an illustration on the technical reasons for this.

To learn more about the performance metrics for models please visit [this page](#)<sup>11</sup>.

---

<sup>6</sup>[https://en.wikipedia.org/wiki/Akaike\\_information\\_criterion](https://en.wikipedia.org/wiki/Akaike_information_criterion)

<sup>7</sup>[https://en.wikipedia.org/wiki/Akaike\\_information\\_criterion#AICc](https://en.wikipedia.org/wiki/Akaike_information_criterion#AICc)

<sup>8</sup>[https://en.wikipedia.org/wiki/Bayesian\\_information\\_criterion](https://en.wikipedia.org/wiki/Bayesian_information_criterion)

<sup>9</sup>[https://en.wikipedia.org/wiki/Coefficient\\_of\\_determination](https://en.wikipedia.org/wiki/Coefficient_of_determination)

<sup>10</sup><http://www.statisticalengineering.com/r-squared.htm>

<sup>11</sup><https://www.otexts.org/fpp/5/3>

## Creating Time Series with 1-Click

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

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

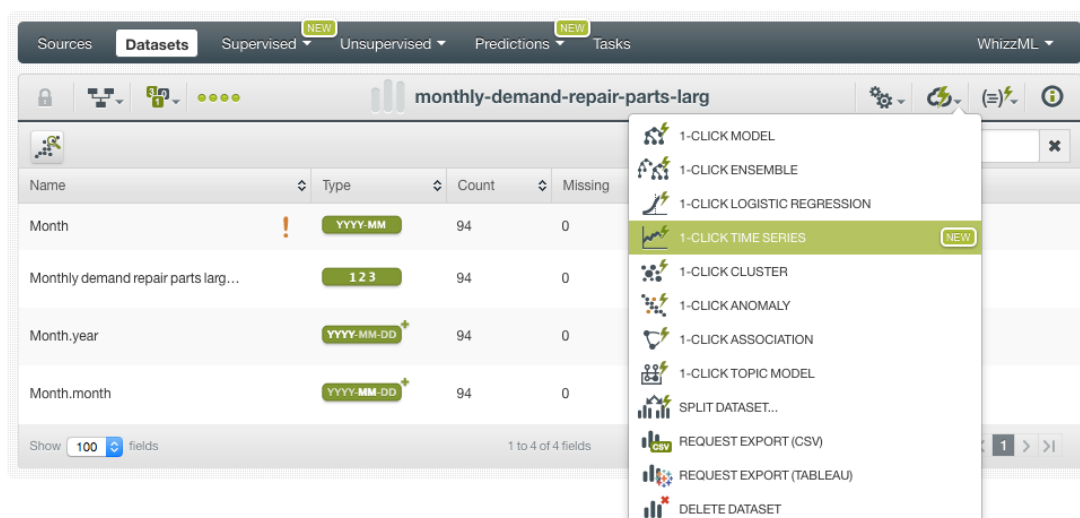


Figure 3.1: Create time series from 1-click action menu

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

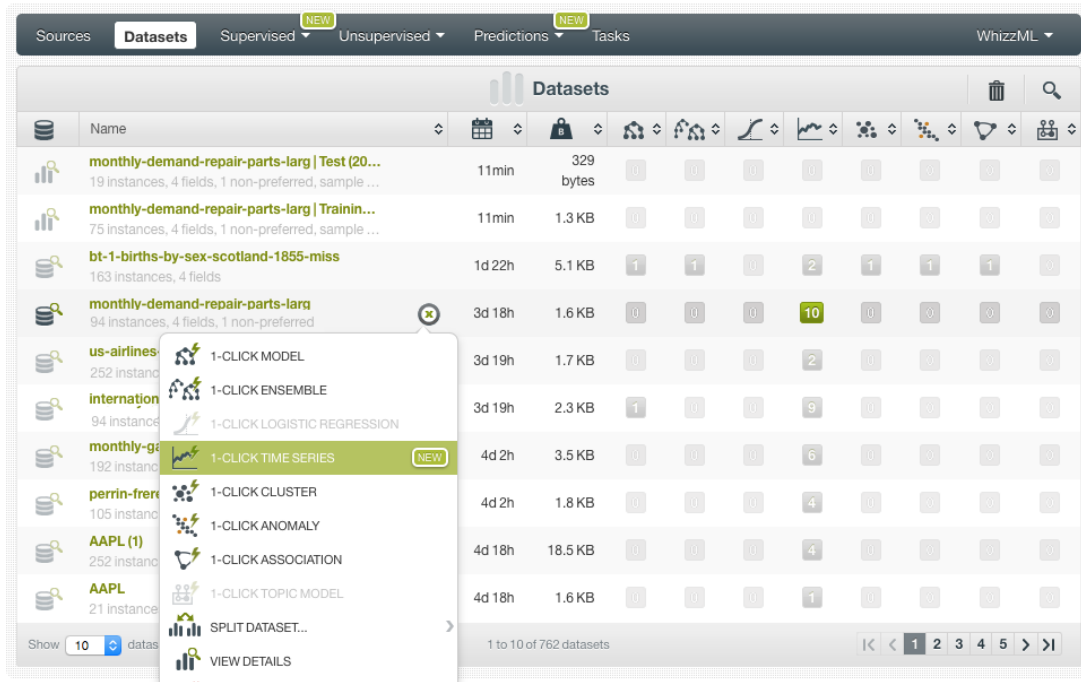


Figure 3.2: Create time series from pop up menu

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

## Time Series Configuration Options

You can configure a number of parameters that affect the way BigML creates time series. See the following sections for a detailed explanation.

To display the configuration panel, click the CONFIGURE TIME SERIES menu option in the **configuration menu** from the dataset view. (See [Figure 4.1.](#))

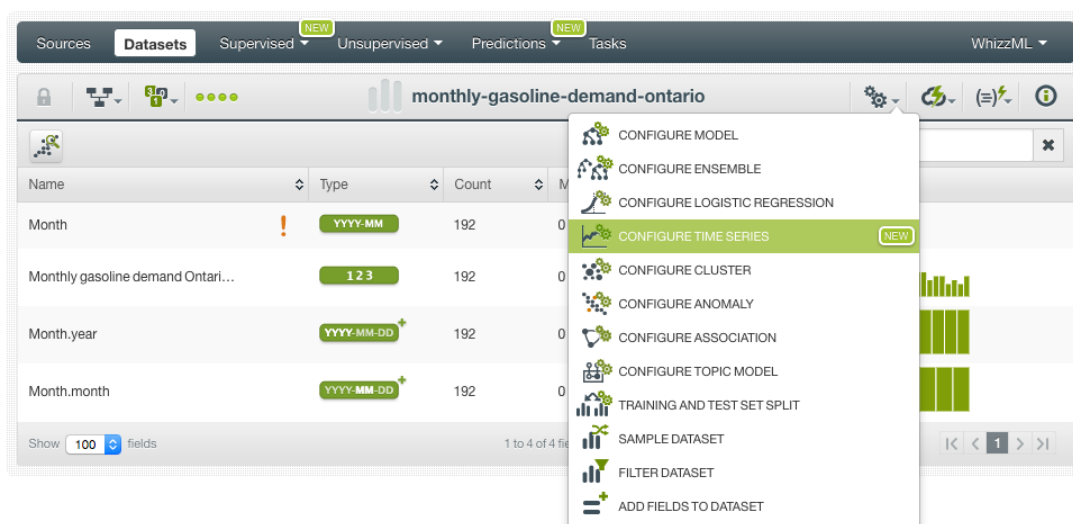


Figure 4.1: Configure time series

### 4.1 Objective Fields

The objective fields are the fields you want to predict. Time series only support **numeric** fields as **objective fields**.

You can **select one or more objective fields** to create your time series by including them in the search box shown in [Figure 4.2](#). BigML takes the **last numeric non-date field** in your dataset as the objective field by default. You can include all numeric fields in your dataset by clicking the option shown in [Figure 4.2](#).

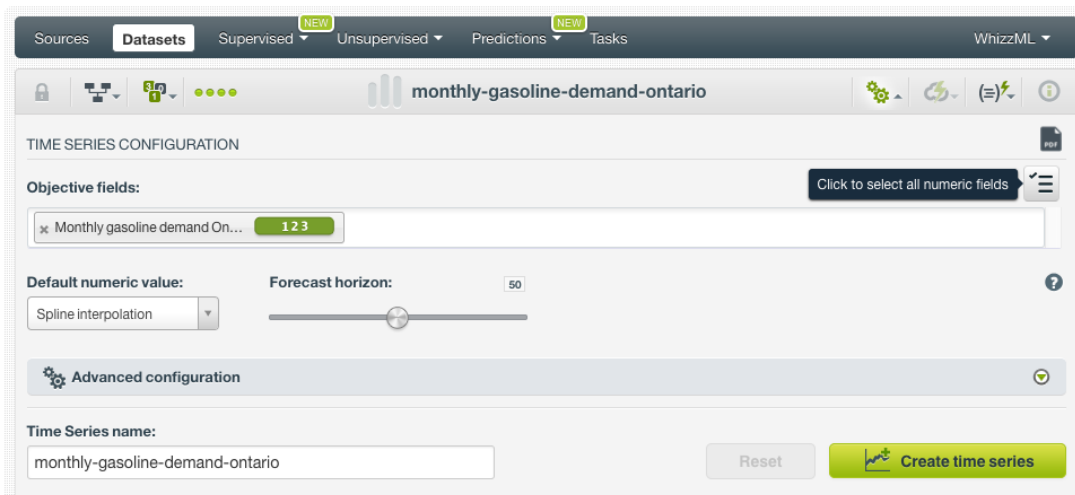


Figure 4.2: Configure time series objective fields

BigML treats each objective field independently, i.e., a set of time series models will be trained for each selected field separately.

## 4.2 Dates

BigML allows you to set the **dates** of your data so you can visualize them in the x-axis of the time series chart (see [Chapter 5](#)).

If your data contains a **date-time field**, it will be automatically selected to be included in the chart (see [Figure 4.3](#)).

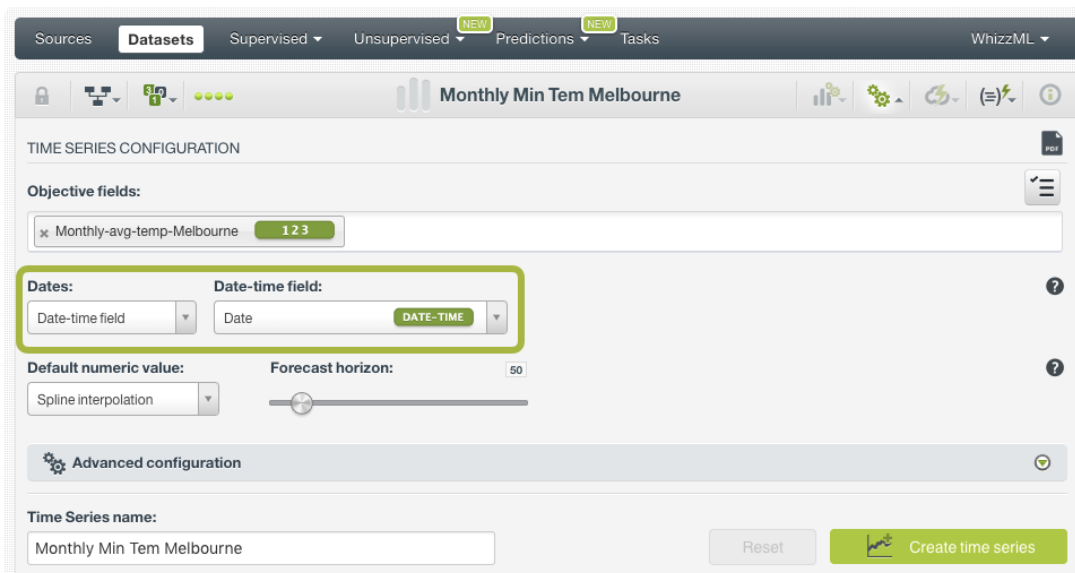


Figure 4.3: Select a date-time field for the chart

Alternatively, you can manually configure the **dates** of your data to be shown in the chart. You need to input the date associated with the first instance and the **time interval** between successive instances (or rows). Sequential values based on the interval will be assigned to each of the instances. The **final date** will be automatically calculated according to the start date, the date interval and the number of instances in your dataset. See how to configure the dates for your instances in [Figure 4.4](#).

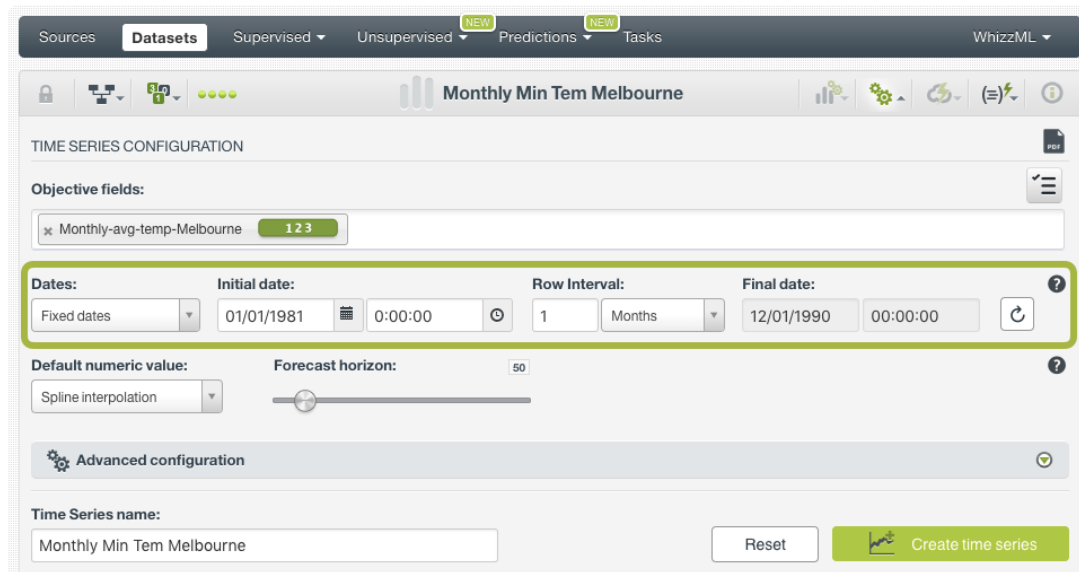


Figure 4.4: Set the dates of your time series

When you create the time series, you will be able to see the dates associated with each instance in the x-axis of the time series chart. If your dataset does not contain any date-time field or you do not configure the dates, the **index** value associated to each instance will be just the **instance position** in the dataset starting at zero for the first instance.

### 4.3 Default Numeric Value

In the case your objective fields include **missing values**, BigML automatically replaces them by [spline interpolation](https://en.wikipedia.org/wiki/Spline_interpolation)<sup>1</sup>. You can also select to replace them by the field's **Mean**, **Median**, **Maximum**, **Minimum** or by **Zero**. (See [Figure 4.5](#).)

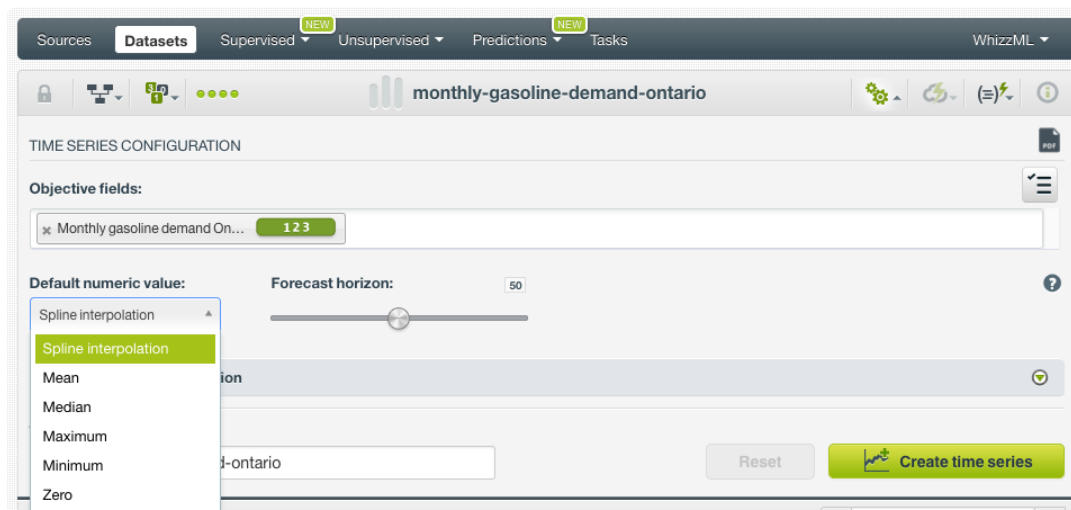


Figure 4.5: Select a default numeric value to replace missing numeric values

<sup>1</sup>[https://en.wikipedia.org/wiki/Spline\\_interpolation](https://en.wikipedia.org/wiki/Spline_interpolation)



## 4.4 Forecast Horizon

BigML automatically computes a **forecast** along with your time series creation. By doing this, you will be able to see your model forecast along with your model view (see [Chapter 5](#)). By default, BigML computes the forecast for a horizon of 50 future data points but you can use the slider shown in [Figure 4.6](#) to forecast up to 400 data points.

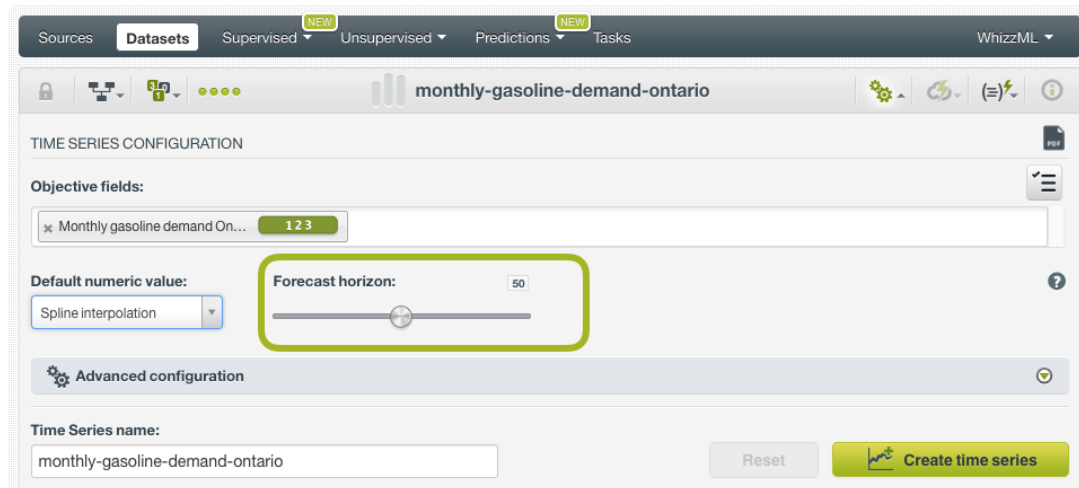


Figure 4.6: Set the horizon for your forecast

## 4.5 Model Components

By combining the different **components** (the error, the trend and the seasonality) and their **variations** (multiplicative, additive, damped and not damped) BigML explores multiple versions of the **forecast formula** so you can choose the model that better fits your training data and make most accurate predictions. These combinations are explained in [Section 2.2](#).

Although BigML explores all combinations by default, you can also exclude certain components or variations from your time series models by using the configuration options shown in [Figure 4.7](#) and explained in the subsections below.

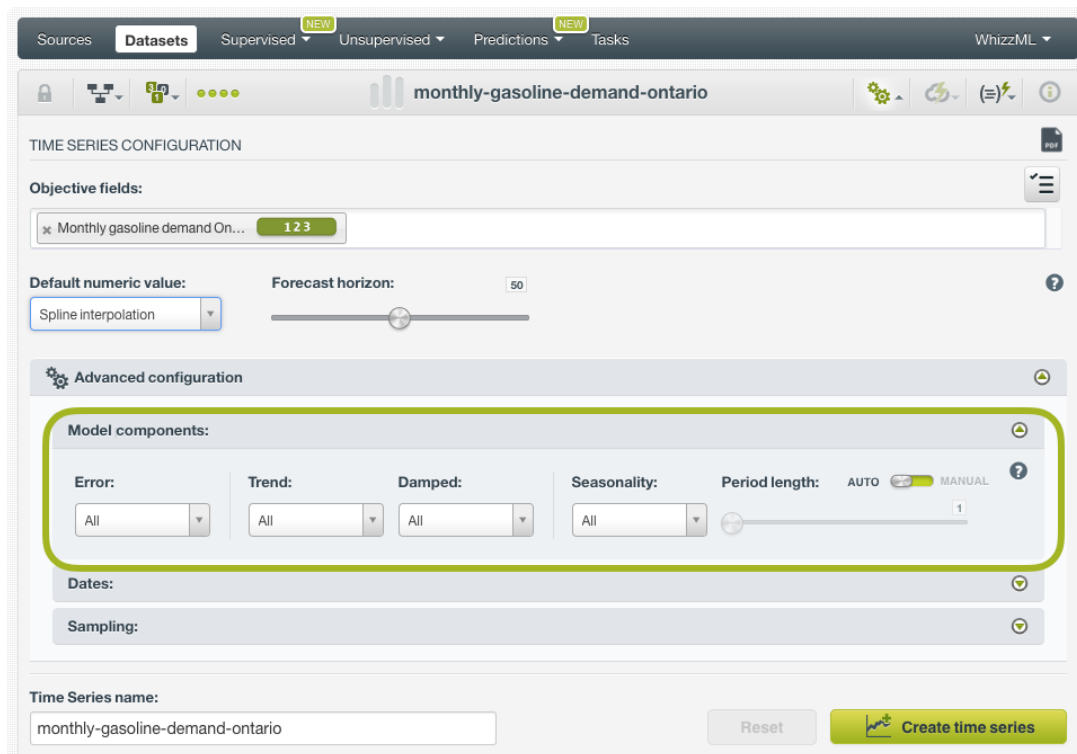


Figure 4.7: Configure the model components to create your time series

### 4.5.1 Error

The **error** represents the unpredictable variations in the time series data, and how they influence observed values. It can be **additive** or **multiplicative** (see the formulas in [Section 2.2](#)). The error type mostly affects forecast **intervals** (see [Section 7.3](#)), and multiplicative error is only suitable for strictly positive data. By default, BigML explores all error variations.

### 4.5.2 Trend and Damping Parameters

Your time series data may have an increasing **trend** (positive slope), a decreasing trend (negative slope) or no trend (horizontal line). When the data has a trend it is important to include a trend component. The trend component can be **additive**, when the growth is linear, or **multiplicative**, when the growth is exponential. The trend can also be **damped**, in which case the growth will be constant at some point in the future, or **not damped**, if the trend keeps growing indefinitely. For long-term forecasts, the damped trend seems a more realistic approach. By default BigML explores all the options mentioned.

You can find a more detailed explanation of the differences for all the trend variations in [Section 2.2](#).

### 4.5.3 Seasonality and Period Length

The data contains **seasonality** when it presents regular fluctuations over time. Seasonal data needs to be modeled with the seasonality component. The seasonality component can be included in your models by setting a **period length higher than 1**. The period length is the number of data points per seasonal period in your data, and it needs to be set taking into account your instances interval and the seasonal frequency. For example, if you have monthly data and annual seasonality, the period should be 12; if you have daily data and a weekly seasonality, the period should be 7. BigML tries to learn the period length in your data by default, but you can also manually configure it by using the slider shown in [Figure 4.7](#).

## 4.6 Sampling

You can specify a subset of instances to create the time series. The **range** option allows you specify the instances you want to use to train your model, e.g., from instance 1 until 200. This option is very useful when you want to train and test your model with different subsets of data and you did not use the linear train and test split explained in [Section 6.1](#).

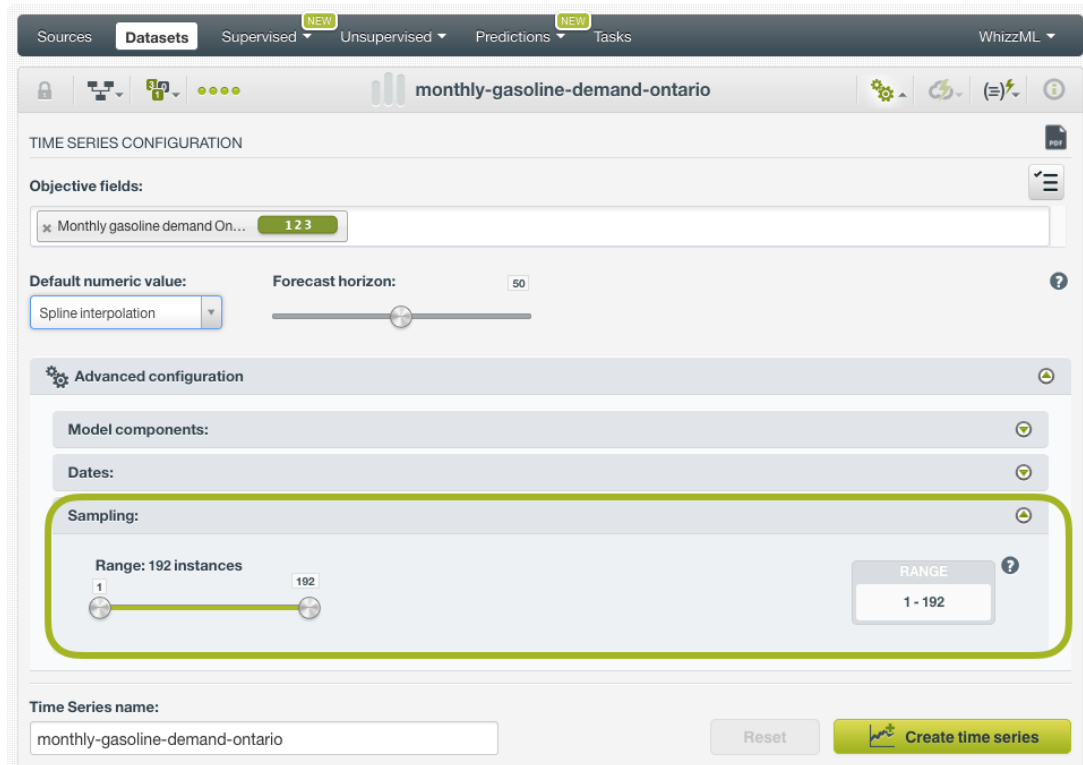


Figure 4.8: Set a range for your time series

## 4.7 Creating Time Series with Configured Options

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

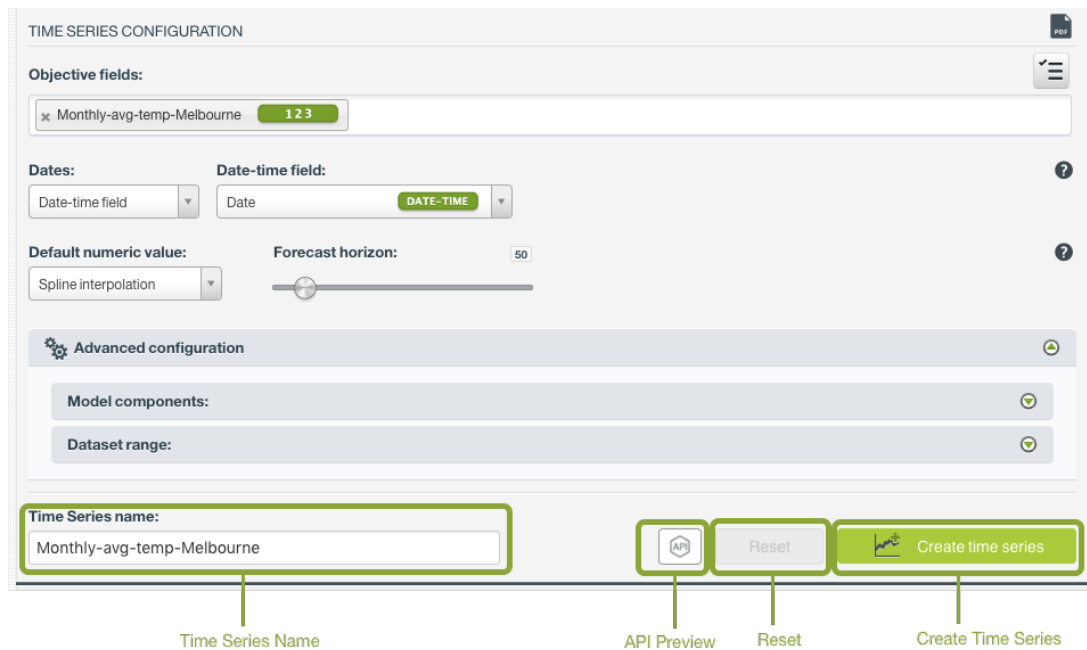


Figure 4.9: Create time series after configuration

## 4.8 API Request Preview

The **API Request Preview** button is in the middle on the bottom of the configuration panel, next to the **Reset** button (See (Figure 4.9)). This is to show how to create the time series programmatically: the endpoint of the REST API call and the JSON that specifies the arguments configured in the panel. Please see (Figure 4.10) below:

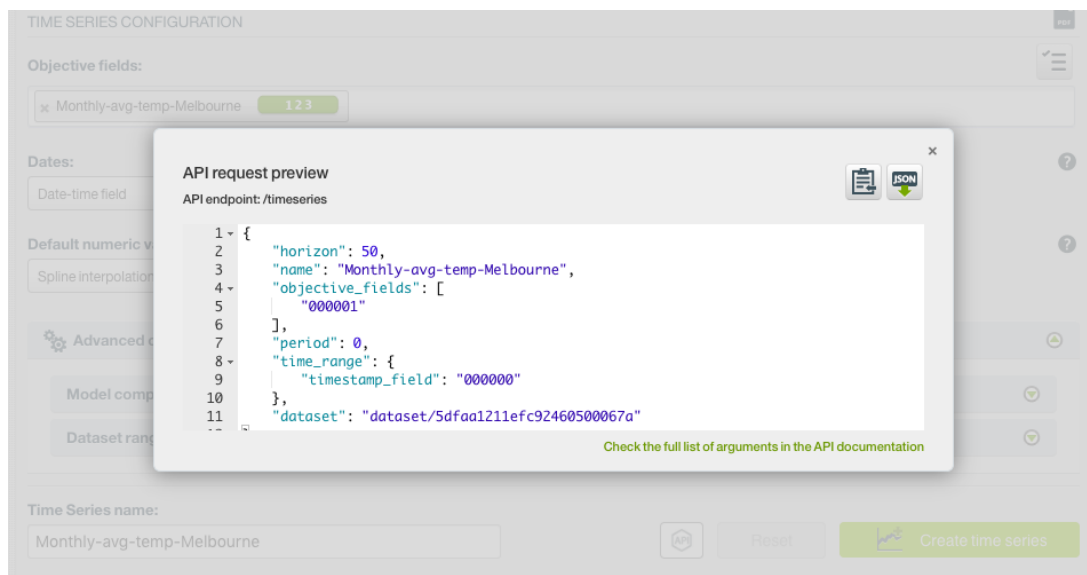


Figure 4.10: Time series API request preview

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

Please note: when a default value for an argument is used in the chosen configuration, the argument won't appear in the generated JSON. Because during API calls, default values are used when arguments

are missing, there is no need to send them in the creation request.

## Visualizing Time Series

BigML time series view shows a chart where you can see the objective field **data** and the **best model** according to the **AIC metric** by default. Below the chart, you can find the selected metric values (the AIC in the image below) and the model components: the **trend**, the **error**, and the **seasonality**. In the [Figure 5.1](#) below the best model has multiplicative errors, trend and seasonality (see [Section 4.5](#) for a detailed explanation of the model components).

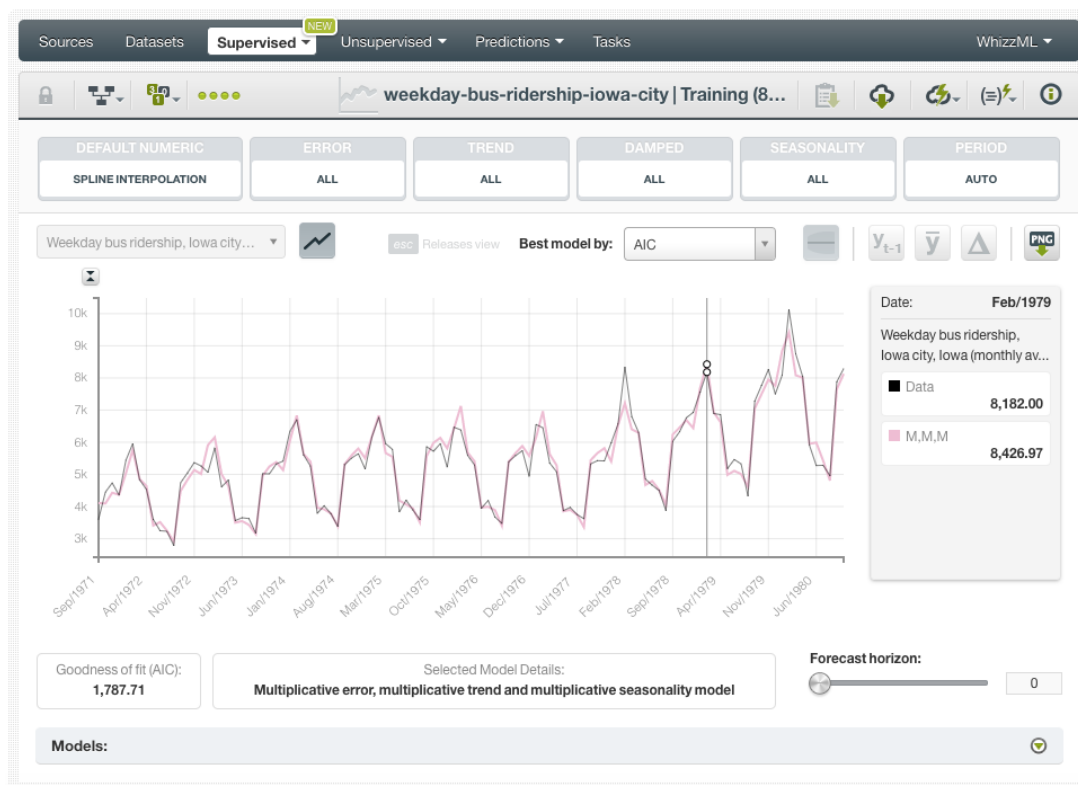


Figure 5.1: Time series view

The AIC is the default metric, but BigML offers three additional performance metrics for you to select the best model:

- **AIC** (Akaike Information Criterion): it measures the trade-off between the model goodness-of-fit and the model complexity to avoid **overfitting**. The lower the AIC, the better.
- **AICc** (Corrected Akaike Information Criterion): it measures the same as AIC metric does, but it is a more accurate metric to avoid overfitting for **small datasets**. For larger datasets it yields same

results as the AIC.

- **BIC** (Schwarz Bayesian Information Criterion): it measures the same trade-off as the AIC between the model goodness-of-fit and its complexity, but it penalizes more heavily the risk of overfitting. The lower the BIC, the better.
- **R squared**: it measures the model goodness-of-fit compared to a mean-based model. The higher the R squared, the better.

Learn about the performance metrics in [Section 2.3](#).

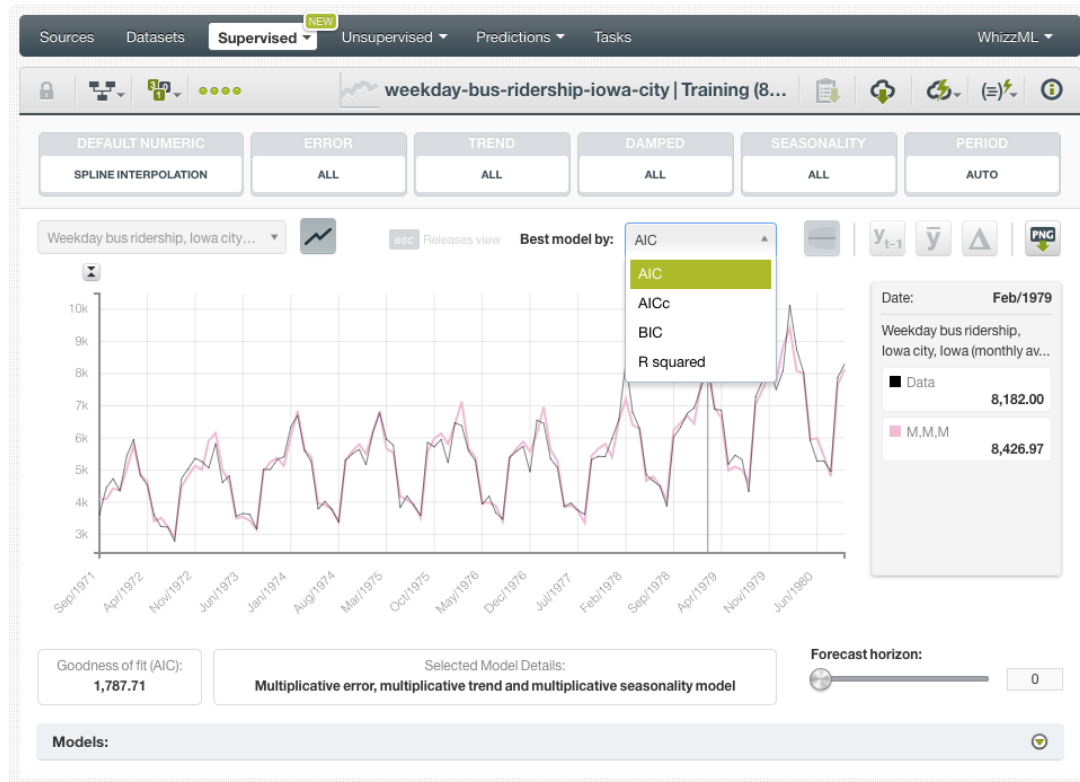


Figure 5.2: Select the best model by any of the available metrics

In the chart, the y-axis represents the **objective field values** and the x-axis the **time horizon**. If you configured more than one objective field (see [Section 4.1](#)), you will be able to see them using the selector shown in [Figure 5.3](#). If you included the **dates** before creating your model (see [Section 4.2](#)) you will be able to see them in the x-axis, otherwise, you will see the instances index starting at zero.

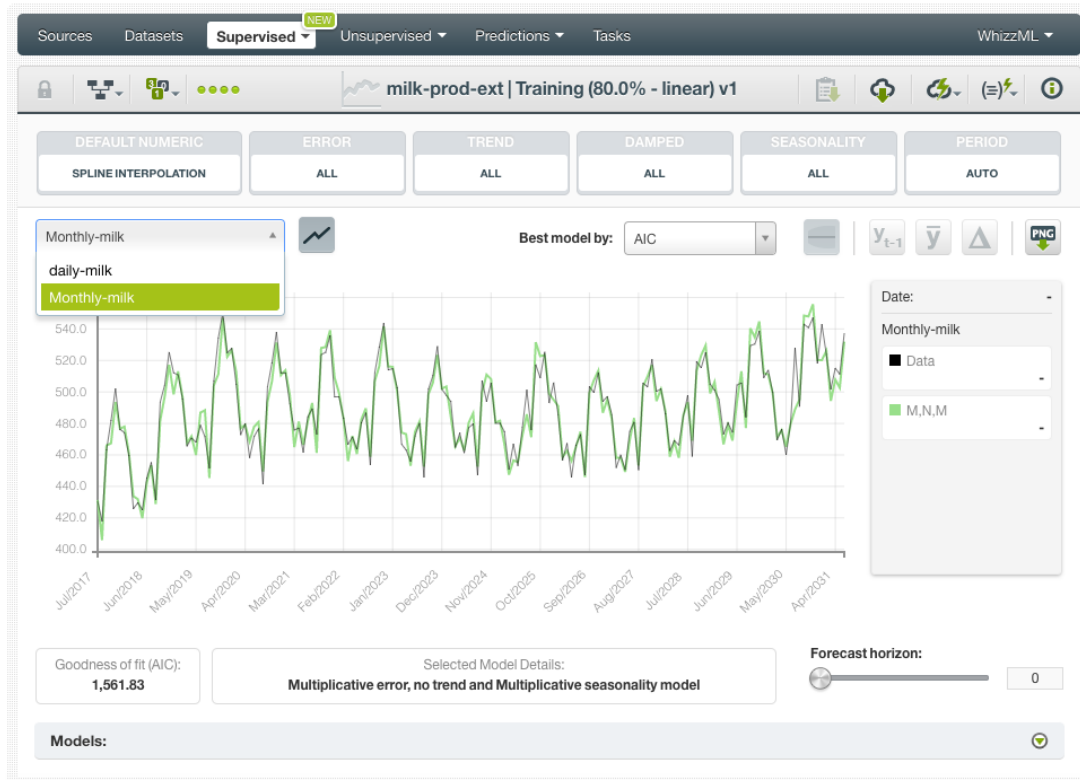


Figure 5.3: Select the objective field to visualize in the time series chart

By hovering over the chart area you will be able to see the axis values in the **legend** to the right. (See [Figure 5.4](#)). You can freeze this view by pressing “Shift”, and release it by pressing “Esc”. You can also **export the chart** in **PNG** format with or without legends.



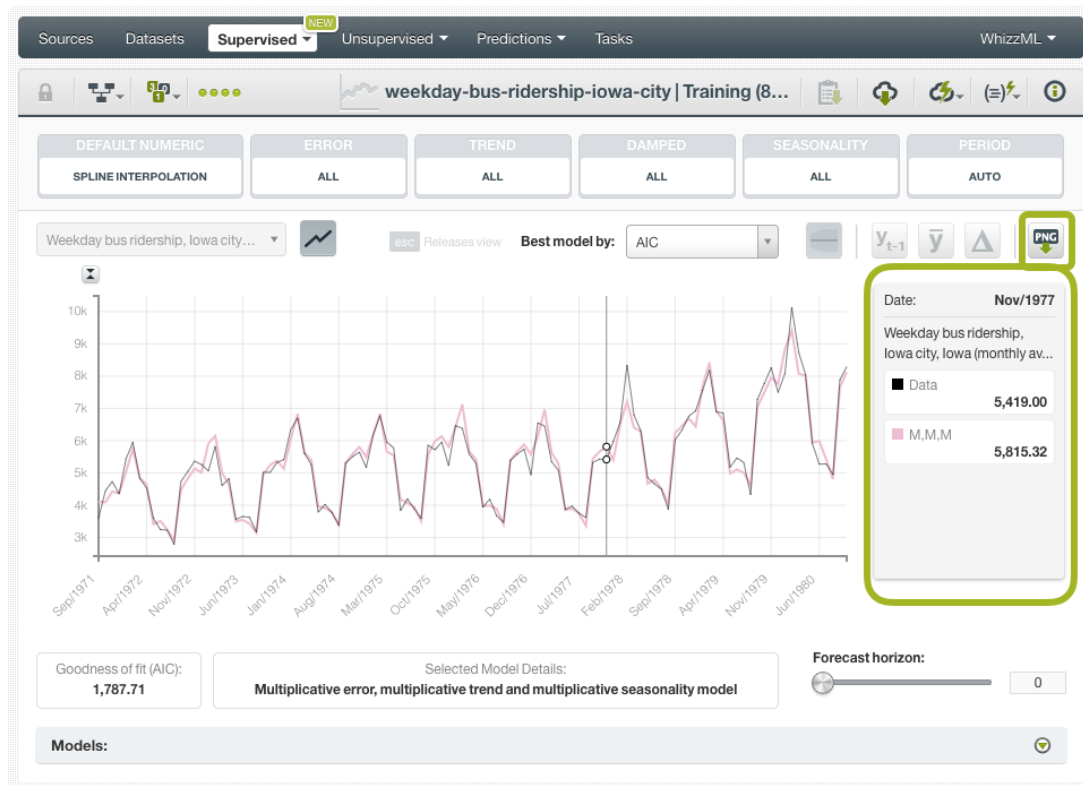


Figure 5.4: See the chart values in the legend

If you display the panel below the chart, you will be able to see all the models learned from your training data. These models represent the different combinations of the forecast components (**error**, **trend**, **seasonality**) and their variations (multiplicative, additive, damped, not damped). The models you select in the table will be plotted in the chart. Each model contains the following information:

- **Name:** each model has a name that represents the variations of its main components "**Error, Trend, Seasonality**". A name like "**A,Md,A**" represents an **additive** error, **multiplicative damped** trend, and an **additive** seasonality model.
- **Model Details:** the model details include the description of the components explained before and its standard name in the literature. For example, a model named "**A,A,N**" (additive errors, additive trend, and no seasonality) is commonly known as **Holt's linear method**.

Learn about the different model components in [Section 2.2](#) and [Section 4.5](#).

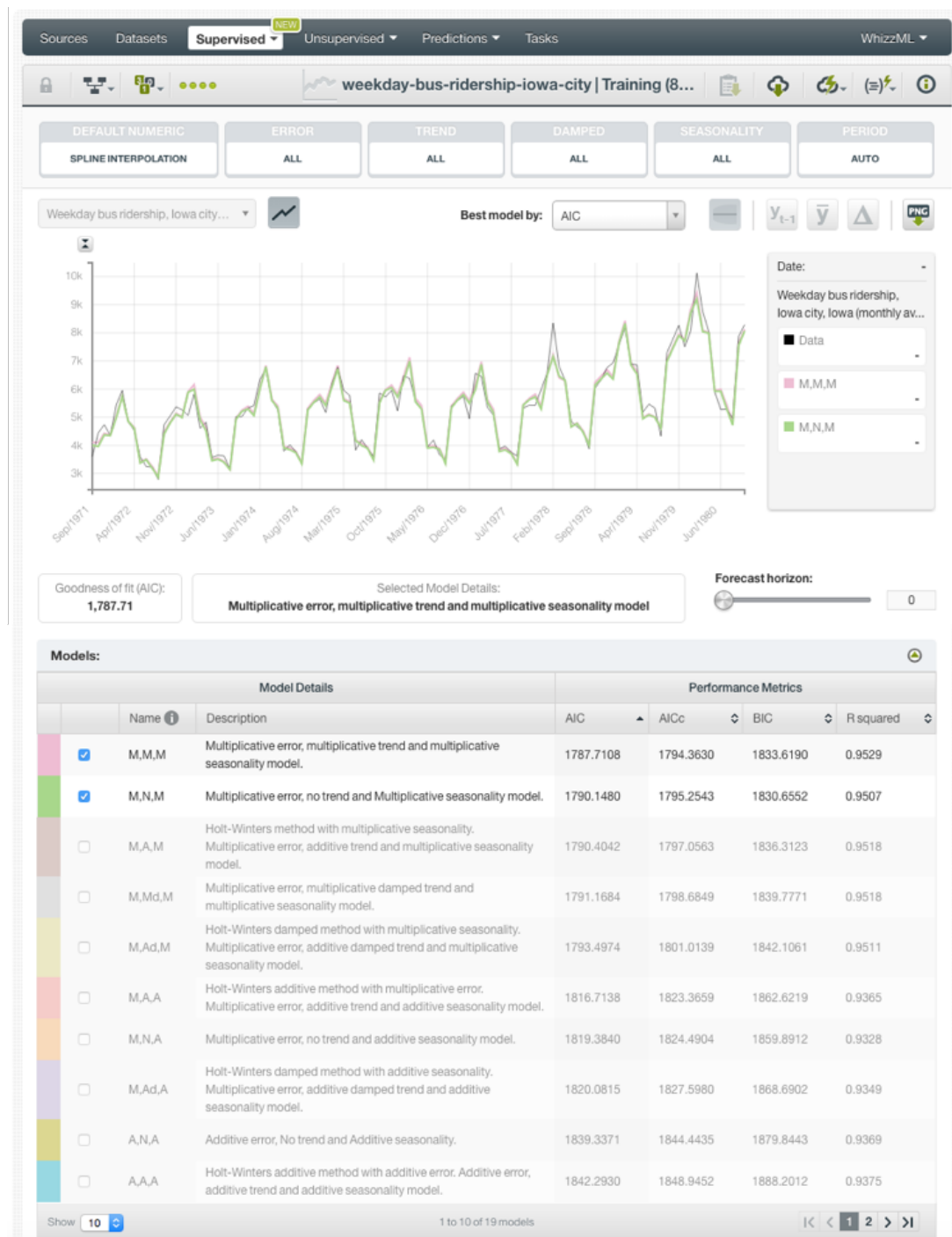


Figure 5.5: See all the models learned from the training data

## 5.1 Time Series Decomposition

You can decompose your models to display **each of the time series components** plotted in a separate chart: the **level** (or error), the **trend** and the **seasonality** (see [Section 4.5](#)).

You need to select a model and click on the option on top of the chart shown in [Figure 5.6](#). If you select more than one model at the same time this option will be disabled.

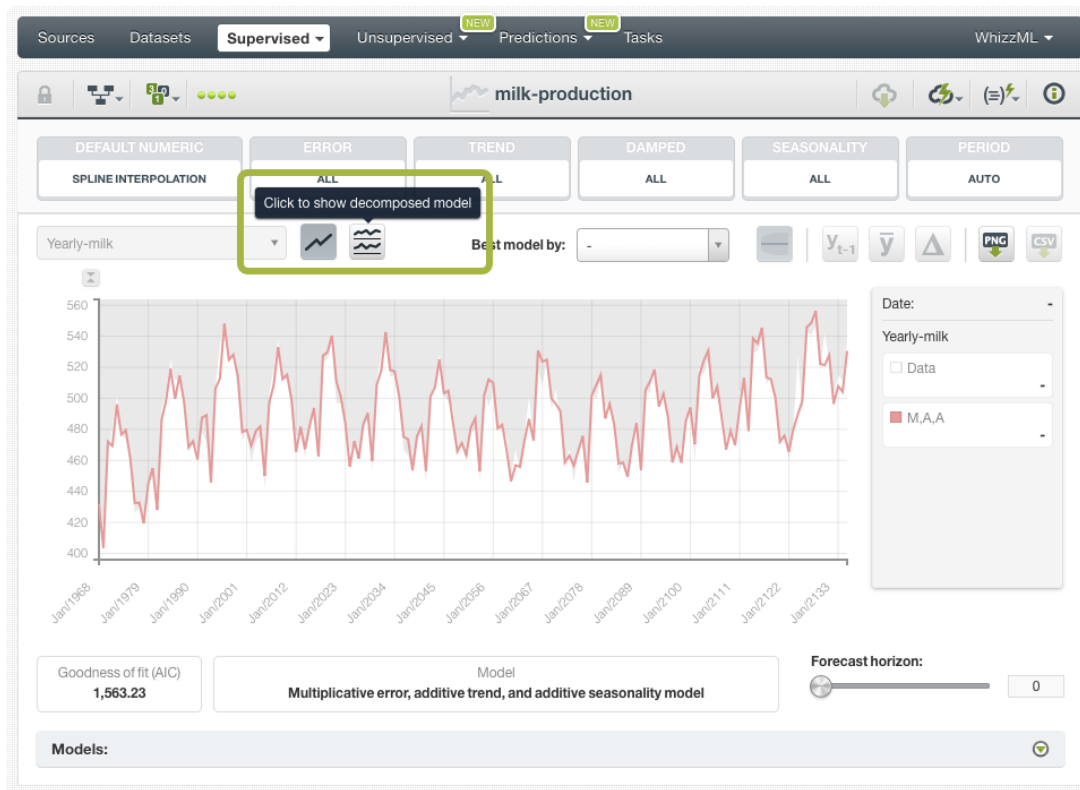


Figure 5.6: Show decomposed model

See the three components (level, trend, and seasonality) plotted in different charts (see [Figure 5.7](#)). Close this view by clicking again on the same icon that you used to display it.

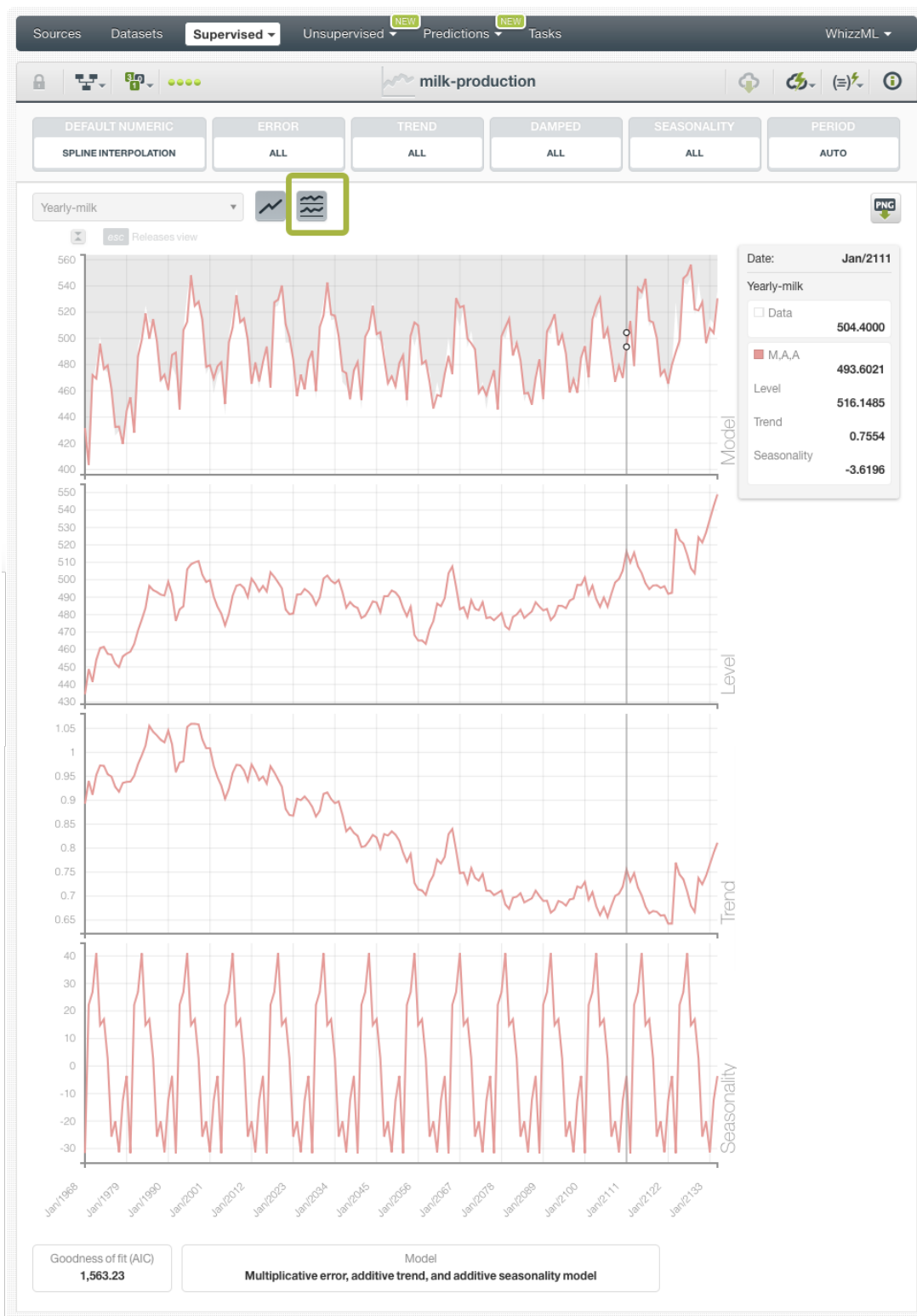


Figure 5.7: Visualize the model components separately

# Time Series Evaluations

## 6.1 Introduction

Time series evaluations provide an easy way to measure the performance of time series models. The main purpose of evaluating your time series is twofold:

- First, obtaining an estimation of the model's performance in production (i.e., making predictions for new instances the model has never seen before).
- Second, providing a framework to compare models to help identify the ones with best predictive performance.

The basic idea behind evaluations is to take some test data different from the one used to train your model and create a forecast for every instance and every model. Then compare the actual **objective field** values of the instances in the test data against the forecasts and compute several performance measures based on the correct results as well as the errors made by the model.

The usual way to obtain some test data is to **split** a **dataset** into two subsets: a training set and a test set. For time series this split needs to be **linear** instead of random because your instances need to be sequentially distributed over time to train the models. You can easily do this from your BigML **Dashboard** by using the **1-click menu option** that automatically takes the first **80% of the instances for training** and the most recent **20% for testing** (see [Figure 6.1](#)).

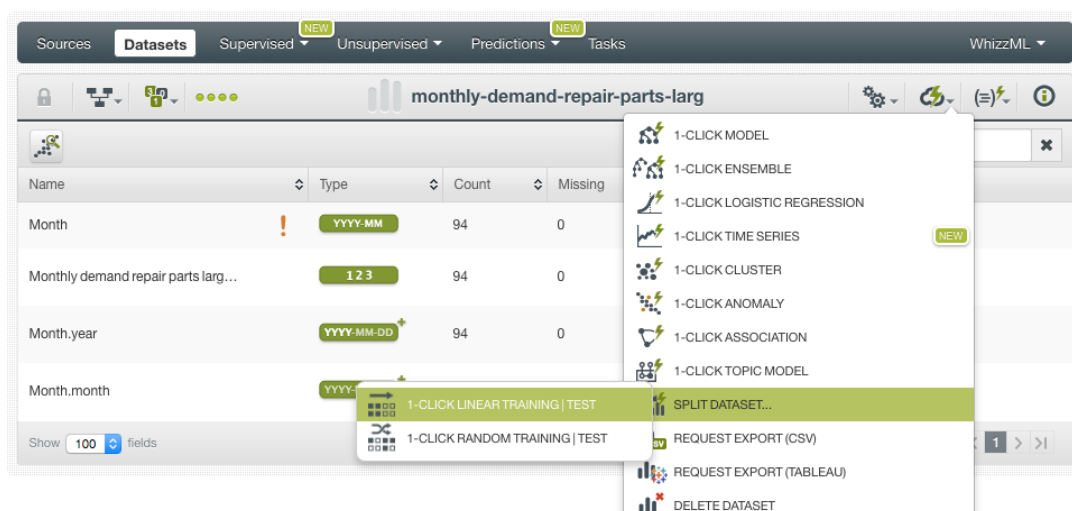


Figure 6.1: 1-click linear train and test split

Alternatively, you can also **configure** those percentages as shown in [Figure 6.2](#). Please keep in mind

that you need to click the **linear split** option if you are creating a time series so your dataset is not randomly split.

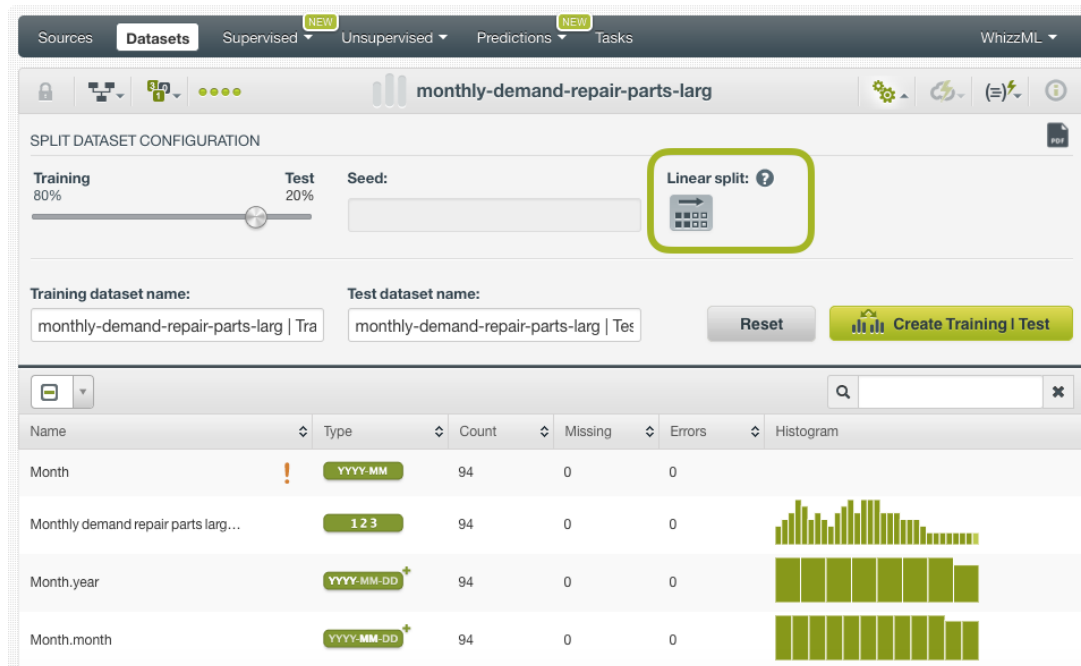


Figure 6.2: Configure the linear train and test split

After creating the time series evaluation (see [Section 6.2](#)), you will be able to graphically see the goodness-of-fit for your training data and the different performance measures to help you select the best performing model (see [Section 6.4](#)). BigML evaluations are first-class citizens. This means that they can be created via the BigML API and can also be queried automatically (see [Subsection 6.5.1](#) for an example).

The evaluations section in the BigML [Dashboard](#) can be found in the third tab under the model menu (see [Figure 6.3](#).) This section contains all your model's evaluations ordered by creation date so most recent evaluations are found at the top of the list. Order your evaluations by **Name**, by **Type** (classification or regression and cross-validation icons below), by **Performance**, by **Age** (time since the evaluation was created), **Dataset Size** (the testing dataset weight) and **Instances** (number of instances in the testing dataset). The icons on the left of the evaluations names in the **evaluation list view** allow you to go to the original [resources](#) used to create the evaluation (in this case the time series). In the **evaluation list view**, you can search evaluations by name by clicking the SEARCH menu option in the top right corner.

| Name   | Type | Performance | Instances |
|--|------|-------------|-----------|
| monthly-gasoline-demand-ontario- v2 vs. monthly-g...<br>period=12, range=[1, 156]  |      |             | 36        |
| monthly-gasoline-demand-ontario- v1 vs. monthly-g...<br>period=12, range=[1, 156]  |      |             | 36        |
| Monthly Min Tem Melbourne - 80% v1 vs. Monthly Mi...<br>range=[1, 100]   |      |             | 20        |
| monthly-gasoline-demand-ontario- v1 vs. monthly-g...<br>period=12, range=[1, 156]  |      |             | 36        |
| international-airline-passengers (1) vs. international-...<br>period=12, range=[1, 144]                                  |      |             | 144       |
| international-airline-passengers -miss vs. internatio...<br>period=12, range=[1, 119]                                    |      |             | 25        |
| international-airline-passengers -miss vs. internatio...<br>period=12, range=[1, 119]                                    |      |             | 25        |
| _births_and_deaths vs. _births_and_deat...(80.0% - li...<br>use all numeric objectives, none seasonality, period=1, r... |      |             | 42        |
| Monthly Min Tem Melbourne - 80% vs. Monthly Min T...<br>period=12, range=[1, 100]  |      |             | 20        |
| monthly-milk-production-pounds vs. monthly-milk-p...<br>period=12, range=[1, 156]  |      |             | 25        |

Figure 6.3: Evaluations section in the BigML Dashboard

Finally, you can see the corresponding icons used to represent a single time series **evaluation** in [Figure 6.4](#).



Figure 6.4: Time series evaluations icon

## 6.2 Creating Time Series Evaluations

To create a time series evaluation, you need two resources: a **time series** and a **testing** dataset. You can use the EVALUATE option in the **1-click action menu** from the time series view (see [Figure 6.5](#)).

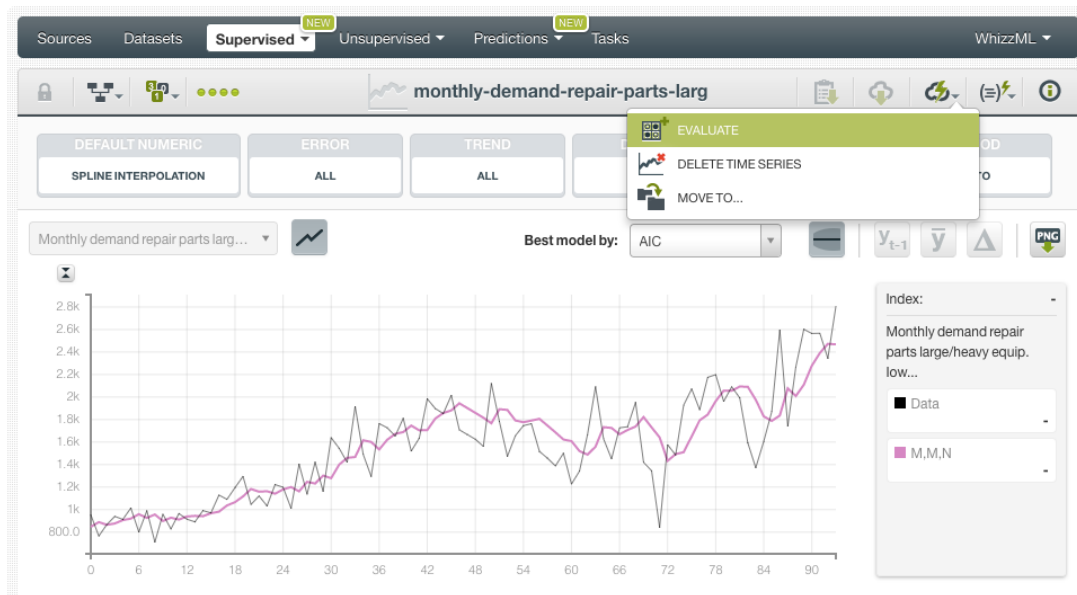


Figure 6.5: Evaluate time series from 1-click action menu

Alternatively, you can use the **EVALUATE** option in the **pop up menu** from the time series list view (see Figure 6.6).

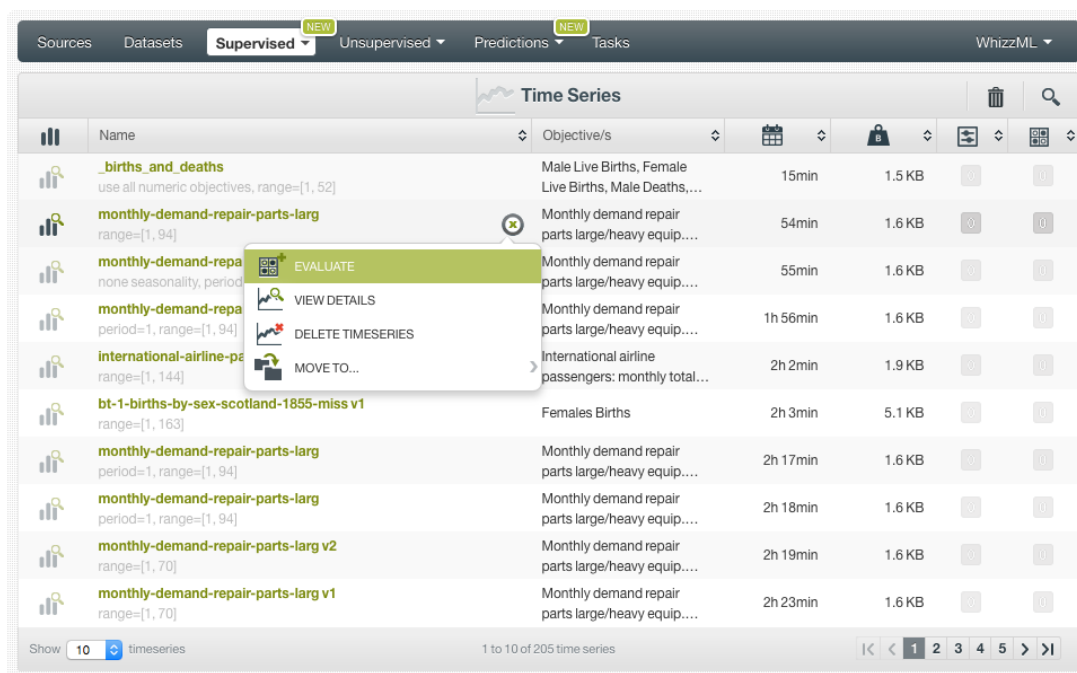


Figure 6.6: Evaluate time series from pop up menu

By using any of these options, you will be redirected to the **New Evaluation** view where the time series will be pre-filled in the selector and you only have to choose the testing dataset. If you previously split your original dataset into two subsets, one for training and another for testing (see Section 6.1), BigML will automatically select the corresponding testing dataset. Finally, click the **Evaluate** green button to perform the evaluation. (See Figure 6.7.)



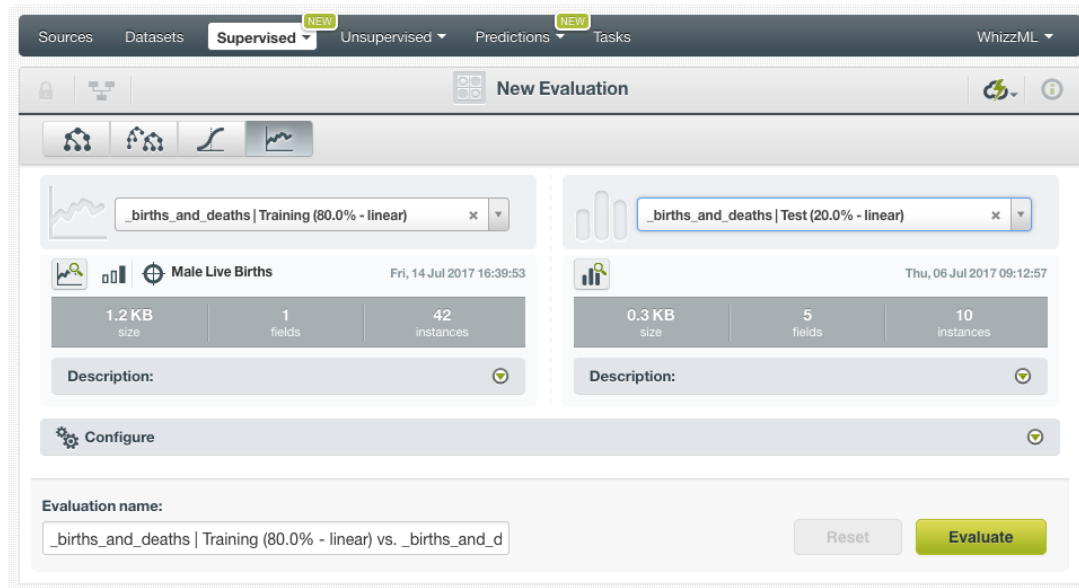


Figure 6.7: Evaluate time series using a testing dataset

## 6.3 Configuring Time Series Evaluations

BigML allows you to configure the **field mapping** and the dataset **range** for time series evaluations.

### 6.3.1 Fields Mapping

You can specify which fields in the time series model match with the fields in the testing dataset. BigML automatically matches fields by **name**, but you can also set an automatic match by **field ID** when clicking on the green switcher shown in [Figure 6.8](#). You can also **manually** search for fields or remove them from the **Dataset fields** column if you do not want them to be considered during the evaluation.

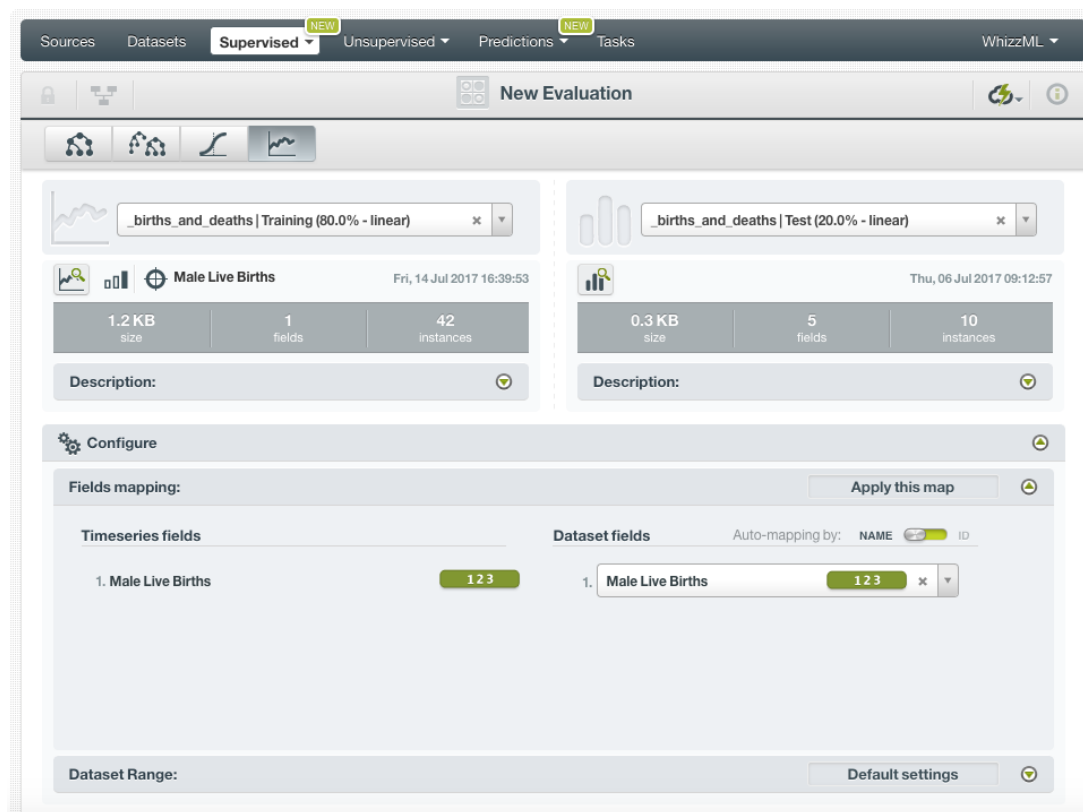


Figure 6.8: Fields Mapping for evaluations

**Note:** the fields mapping option has a limit of 200 fields from the BigML Dashboard. If you need to map your fields for evaluations with more than 200 fields, please use the [BigML API](https://bigml.com/api/evaluations#ev_evaluation_arguments)<sup>1</sup>.

### 6.3.2 Range

Sometimes you do not need all the data contained in your testing dataset to generate your evaluations. If you have a very large dataset, sampling may be a good way of getting faster results. BigML allows you to take a linear sample before creating an evaluation so you do not need to create a different dataset. By using the **range** you can specify a linear subset of instances from which to sample, e.g., choosing from instance 1 to 30.

<sup>1</sup>[https://bigml.com/api/evaluations#ev\\_evaluation\\_arguments](https://bigml.com/api/evaluations#ev_evaluation_arguments)

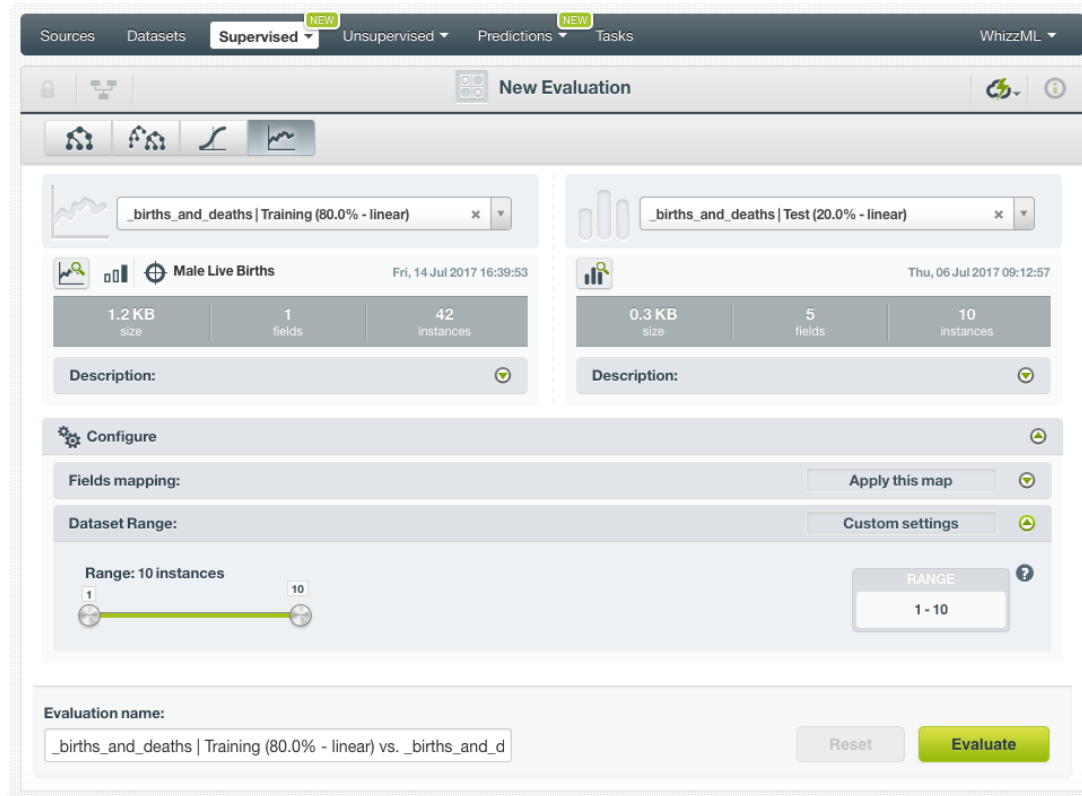


Figure 6.9: Configure the range for evaluations

## 6.4 Visualizing Time Series Evaluations

Time series visualizations are composed of two main parts: a **chart** to graphically inspect the goodness of fit of the selected models, and a **table** which contains **all the models** learned along with their **evaluation metrics**.

### 6.4.1 Evaluation Chart

The evaluation chart shows the **model view** and the selected model **forecasts for the testing data** (see Figure 6.10). This chart allows you to **visually analyze** the **goodness-of-fit** of your models over the testing data. By default, BigML selects the model with the **best R squared** metric, but you can select the best model by other metrics such as the MAE, MSE, sMAPE, or MASE explained in the subsections below. You can see the MAE, MSE and R squared metrics of the selected model below the chart (see Figure 6.10).

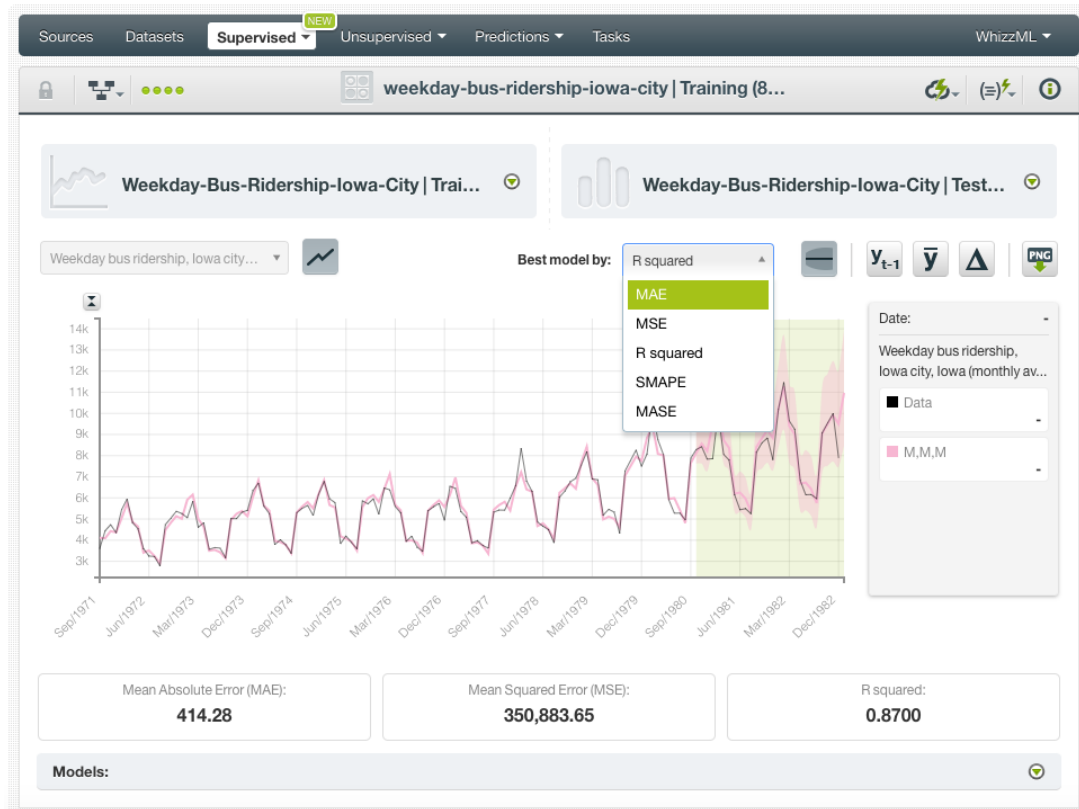


Figure 6.10: Time series evaluations

The forecast data points for the test data are plotted along with a **confidence interval** which is represented by a color area with less opacity in the chart. This is the error interval in which the forecast can lay with a confidence of 95%. You can hide or show these intervals by clicking in the option shown in Figure 6.11. Longer forecast horizons will usually generate wider error intervals, since time series diminish their prediction quality as the horizon is larger.

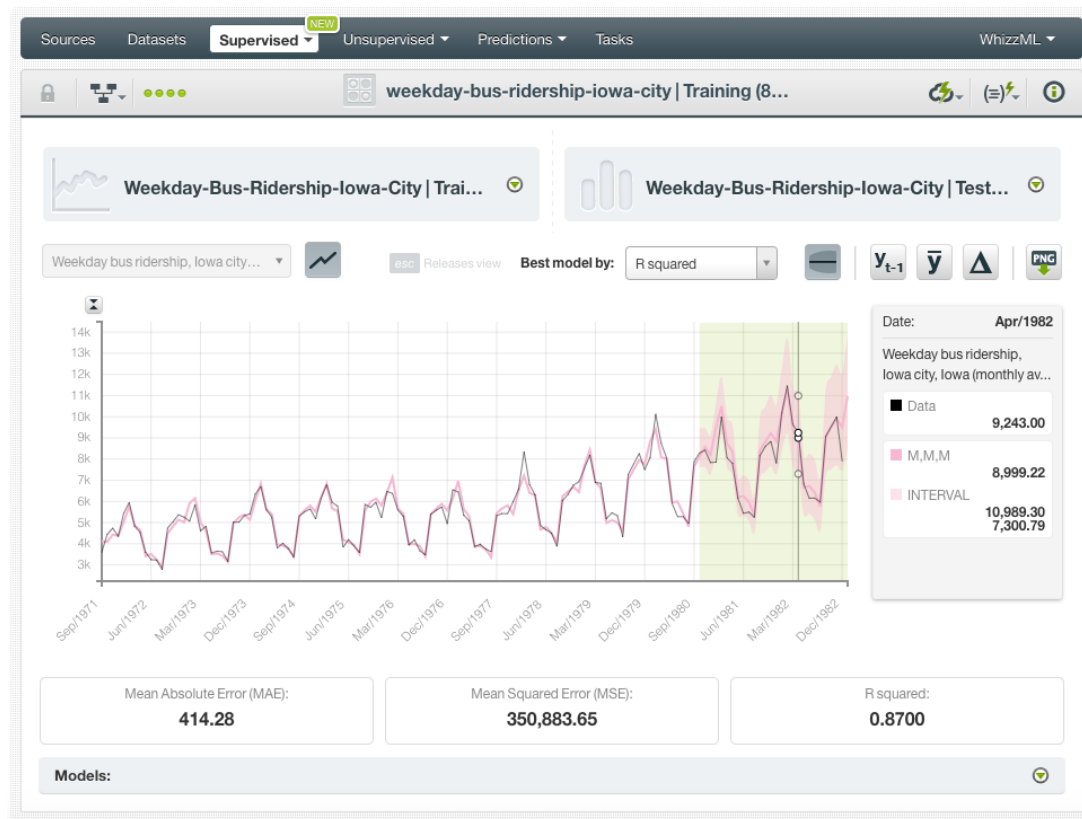


Figure 6.11: Time series evaluation intervals

BigML also provides **three benchmark models** that work as a baseline, so you can compare your model forecasts against them:

- A **mean-based model** that always predicts the mean for all instances in your training dataset.
- A **naive model** that always predicts the last observed value in the dataset.
- A **drift model** that takes the first and last observation and extrapolates the future values.

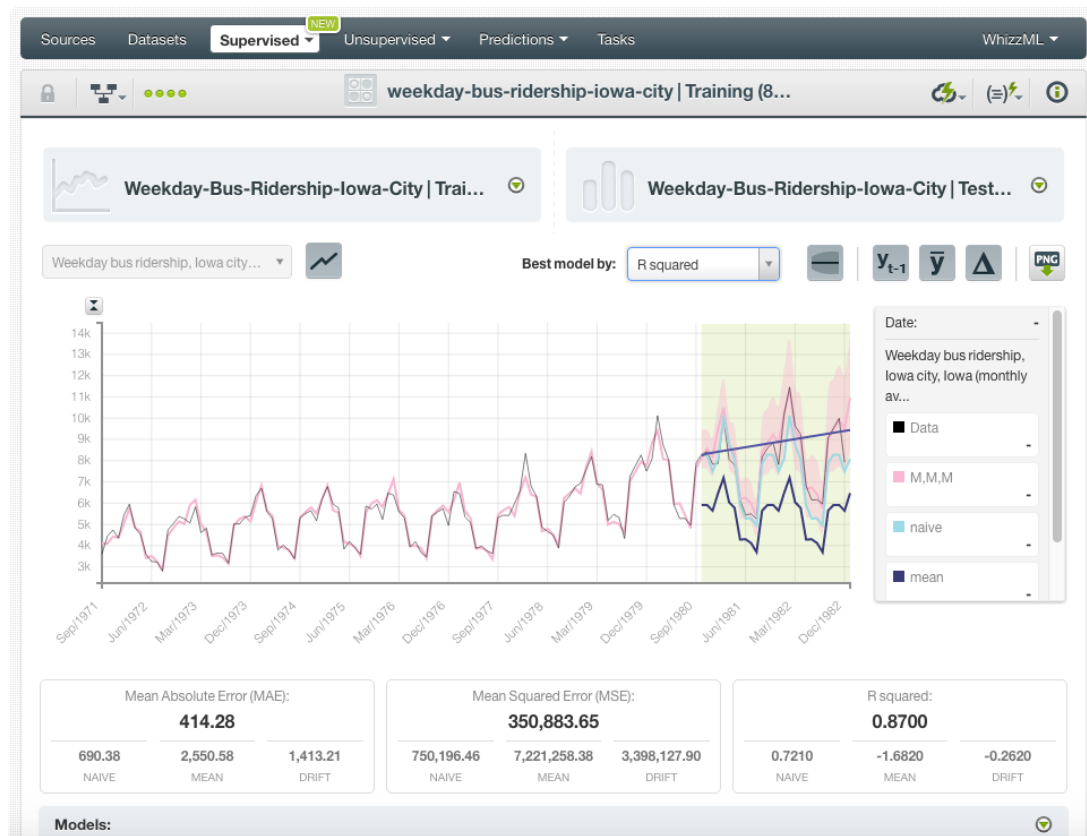


Figure 6.12: Benchmark models for time series

Learn more about these benchmarking methods [here](https://www.otexts.org/fpp/2/3)<sup>2</sup>.

Below the chart, you will find a panel which contains a table with the rest of your models learned from the training data. These models differ from one another because they have different components and variations as explained in [Chapter 2](#). You can visualize them in the chart by selecting them in the table.

<sup>2</sup><https://www.otexts.org/fpp/2/3>

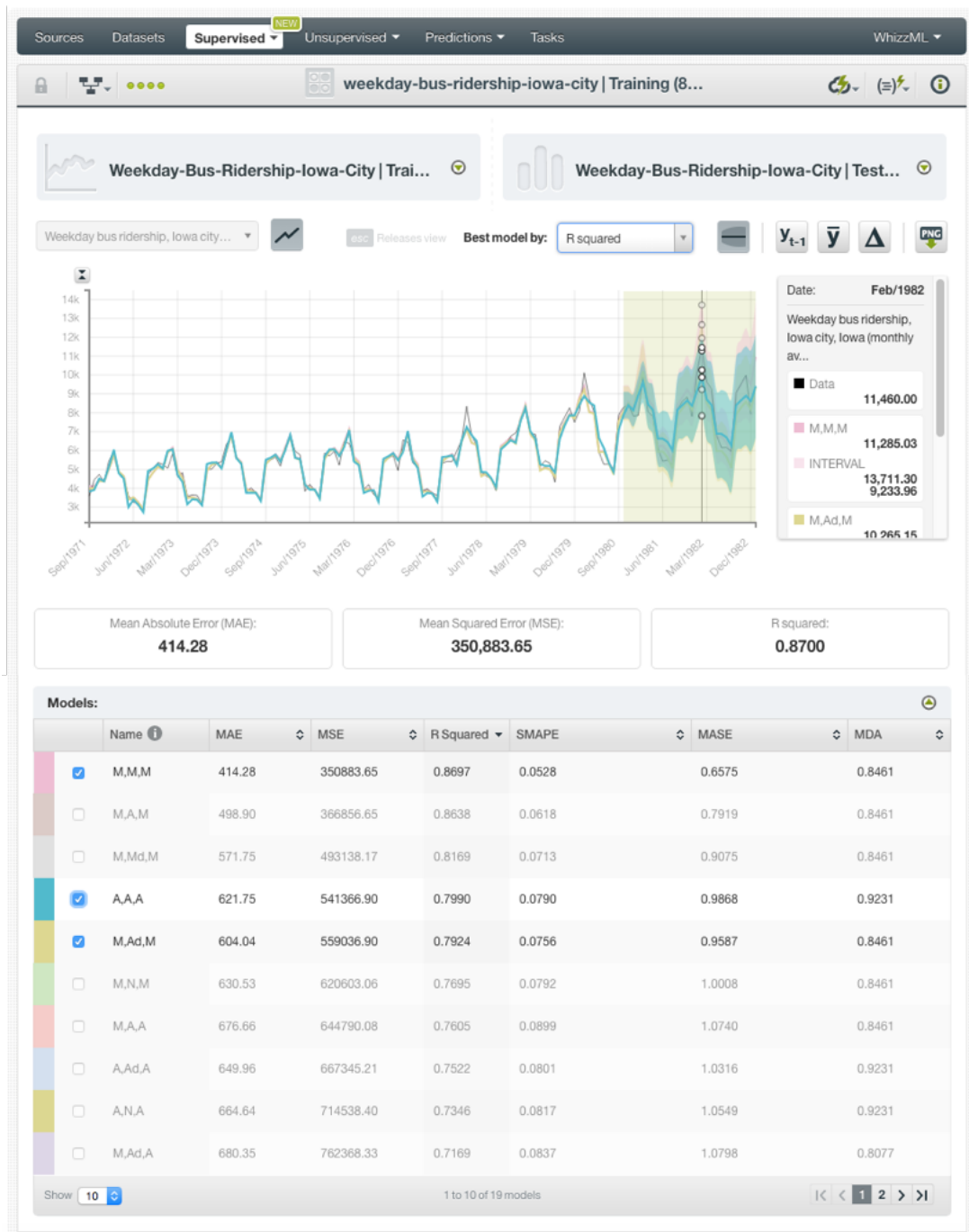


Figure 6.13: Select models to analyze their performance

## 6.4.2 Evaluation Metrics

Time series evaluations contain a set of performance metrics per model to help you identify the best performing model. See an explanation for each of them in the subsections below.

### Mean Absolute Error

The **Mean Absolute Error (MAE)**<sup>3</sup> is the mean of the model forecast errors for each instance. It is computed as the average of the absolute values of the differences between the target variable predicted

<sup>3</sup>[https://en.wikipedia.org/wiki/Mean\\_absolute\\_error](https://en.wikipedia.org/wiki/Mean_absolute_error)

by the model ( $\hat{y}$ ) vs. the actual values ( $y$ ). The lower the MAE, the better. Letting  $T$  be the period length of the time series in the test dataset, i.e., the total number of instances evaluated, then:

$$\text{MAE} = \frac{\sum_t |\hat{y}_t - y_t|}{T}$$

### Mean Square Error

The **Mean Square Error (MSE)**<sup>4</sup> is similar to the Mean Absolute Error, but the differences between predictions and actual values are squared. It is computed as the average of the squares of the differences between the target variable predicted by the model ( $\hat{y}$ ) vs. the actual values ( $y$ ). Letting  $T$  be the period length of the time series in the test dataset, i.e., the total number of instances evaluated, then:

$$\text{MSE} = \frac{\sum_t (\hat{y}_t - y_t)^2}{T}$$

### R Squared

The  $R^2$ , also called the **coefficient of determination**<sup>5</sup>, measures how much better the model is than always predicting the mean value of the target variable ( $\bar{y}$ ) in the test set. It can produce values up to 1. Values below 0 indicate the model is worse than predicting the mean; a value of 0 means the model is not any better than predicting the mean; and 1 means the model perfectly fits the data. Higher values for  $R^2$  usually mean better performance.

$$R^2 = 1 - \frac{\sum_t (\hat{y}_t - y_t)^2}{\sum_t (y_t - \bar{y})^2}$$

### Symmetric Mean Absolute Percentage Error

Percentage errors calculate the error as a percentage of the objective field actual values:

$$p_t = 100 \frac{\hat{y}_t - y_t}{y_t} = 100 \frac{e_t}{y_t}$$

The main advantage of percentage errors is the scale independence, which makes the errors comparable for different objective fields. However, if  $y_t$  is zero for any  $t$  in the time series, the percentage error will be undefined. This problem is corrected by the **Symmetric Mean Absolute Percentage Error (sMAPE)**<sup>6</sup>:

$$\text{sMAPE} = \frac{\sum_t 200 \frac{|\hat{y}_t - y_t|}{(\hat{y}_t + y_t)}}{T}$$

However, the sMAPE can also be undefined in the case  $(\hat{y}_t + y_t)$  is closer to zero. Moreover,  $(\hat{y}_t + y_t)$  can be negative depending on the objective field values. The scaled errors explained in the following subsection correct these problems.

### Mean Absolute Scaled Error

The **Mean Absolute Scaled Error (MASE)**<sup>7</sup> corrects the problems of the sMAPE explained in the previous section when  $(\hat{y}_t + y_t)$  is closer to zero or negative. The scaled error divides the error by the naive forecast error (remember the naive forecast always forecasts the previous value in the time series):

$$\text{MASE} = \frac{\sum_t \frac{|\hat{y}_t - y_t|}{T-1}}{\sum_{t=2}^T |y_t - y_{t-1}|}$$

<sup>4</sup>[https://en.wikipedia.org/wiki/Mean\\_squared\\_error](https://en.wikipedia.org/wiki/Mean_squared_error)

<sup>5</sup>[https://en.wikipedia.org/wiki/Coefficient\\_of\\_determination](https://en.wikipedia.org/wiki/Coefficient_of_determination)

<sup>6</sup>[https://en.wikipedia.org/wiki/Symmetric\\_mean\\_absolute\\_percentage\\_error](https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error)

<sup>7</sup>[https://en.wikipedia.org/wiki/Mean\\_absolute\\_scaled\\_error](https://en.wikipedia.org/wiki/Mean_absolute_scaled_error)



## Mean Directional Accuracy

The [Mean Directional Accuracy \(MDA\)](#)<sup>8</sup>, compares the forecast direction (upward or downward) to the actual data direction. MDA provides the probability that the model can detect the correct direction of the time series. A value of 1 indicates a perfect fit between the time series direction and the model forecast direction.

To learn more about the time series performance metrics please visit [this page](#)<sup>9</sup>.

## 6.5 Consuming Time Series Evaluations

BigML allows you to create and consume your evaluations programmatically via the **BigML API and bindings**. The following subsections explain both tools.

### 6.5.1 Using Time Series Evaluations Via the BigML API

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

See below how to create an evaluation just using a model and a testing dataset once you have properly set the `BIGML_AUTH` environment variable to contain your authentication credentials:

```
curl "https://bigml.io/evaluation?${BIGML_AUTH}" \  
-X POST \  
-H 'content-type: application/json' \  
-d '{"dataset": "dataset/50650bdf3c19201b64000020", \  
    "timeseries": "timeseries/50650bea3c19201b64000024"}'
```

### 6.5.2 Using Time Series Evaluations Via the BigML Bindings

You can create and use evaluations via BigML bindings which are libraries aimed to make it easier to use the BigML API from your language of choice. BigML offers bindings for several languages including Python, Node.js, Java, Swift or Objective-C. See below an example to create an evaluation with the BigML Python bindings.

```
from bigml.api import BigML  
api = BigML()  
evaluation = api.create_evaluation('timeseries/57506c472275c1666b004b10',  
                                  'dataset/576d213d983efc63e8000038')
```

For more information on BigML bindings, please refer to the [bindings page](#)<sup>10</sup>.

## 6.6 Deleting Time Series Evaluations

You can delete your evaluations by clicking on the `DELETE EVALUATION` option from the **1-click action menu** from the evaluation view ([Figure 6.14](#)).

<sup>8</sup>[https://en.wikipedia.org/wiki/Mean\\_Directional\\_Accuracy\\_\(MDA\)](https://en.wikipedia.org/wiki/Mean_Directional_Accuracy_(MDA))

<sup>9</sup><https://www.otexts.org/fpp/2/5>

<sup>10</sup><https://bigml.com/tools/bindings>

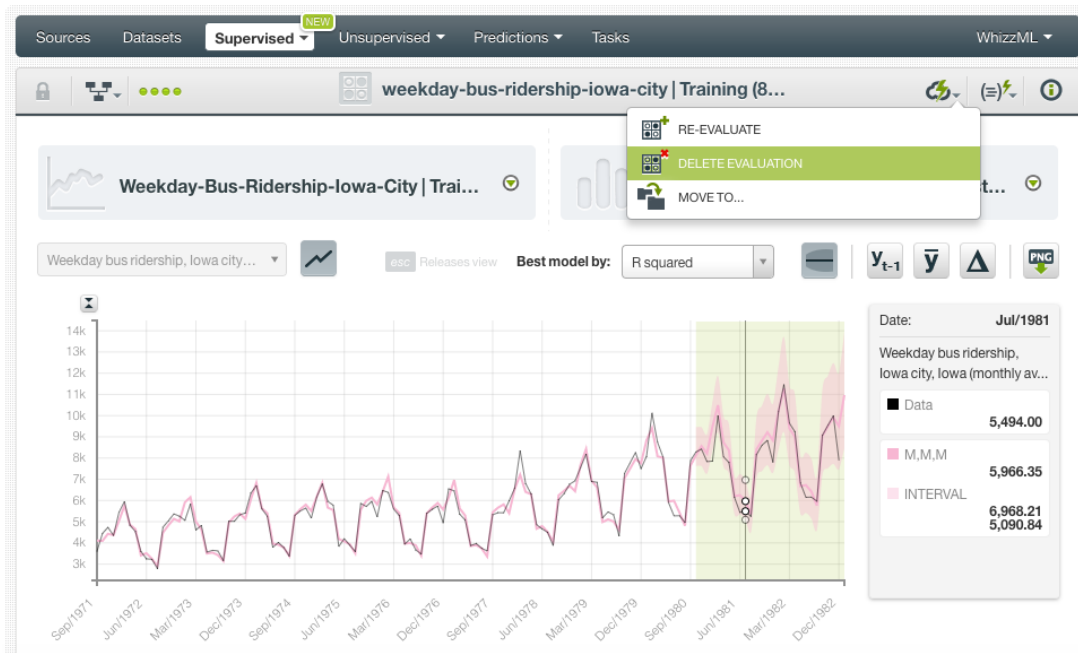


Figure 6.14: Delete evaluation from the 1-click action menu

Alternatively, you can click the DELETE EVALUATION option in the **pop up menu** on the evaluation list view (Figure 6.15).

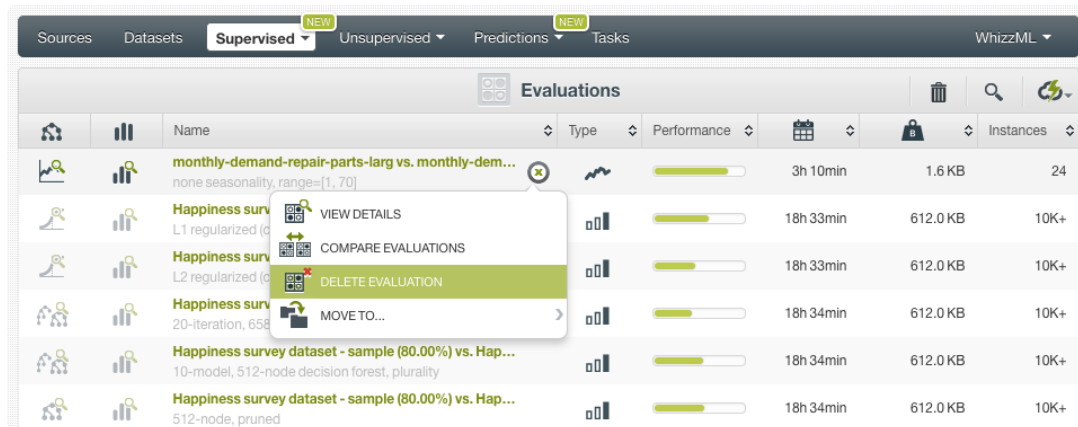


Figure 6.15: Delete evaluation from pop up menu

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

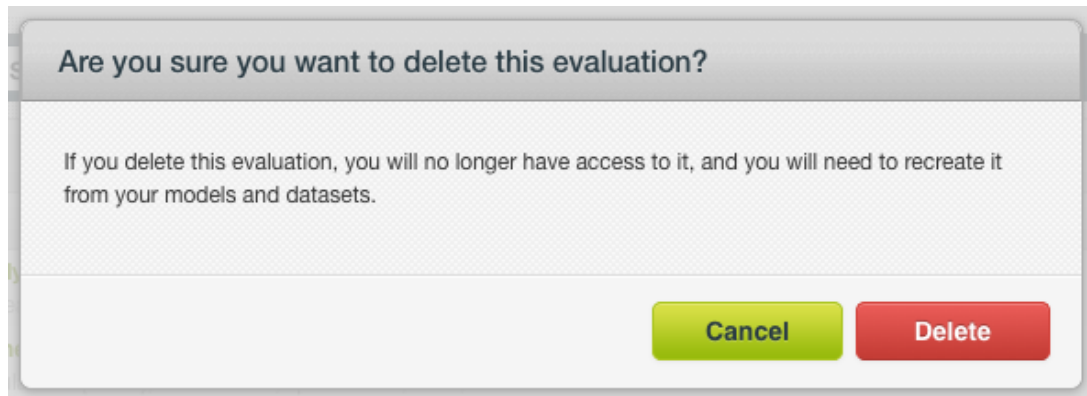


Figure 6.16: Delete evaluation confirmation

# Time Series Forecasts

## 7.1 Introduction

The ultimate goal of building a BigML time series is being able to make **predictions** for the future. In BigML, time series predictions are called **forecasts**. The forecasts for time series are created during the time series creation.

## 7.2 Creating and Configuring Forecasts

BigML automatically creates a **forecast** of 50 data points during your **time series** creation. You can increase the forecasted horizon up to 400 future data points by configuring the option explained in [Section 4.4](#). When your time series is created, you are able to see the forecast for any of the selected models in the chart by using the slider shown in [Section 7.3](#).

## 7.3 Visualizing Forecasts

When your time series is created, you can use the **Forecast horizon** slider to see the model predictions. As you increase the horizon, the forecasted data points will appear in the chart. Each forecast is generated with a **confidence interval** that sets the maximum and minimum values that the forecast can take with a 95% of confidence. They are obtained by simulating 5,000 predictions paths with noise and computing the sample quantiles. Usually intervals grow as the forecast period increases since there is more uncertainty when the predicted time horizon is longer. You can hide or show these intervals by using the option shown in [Figure 7.1](#).

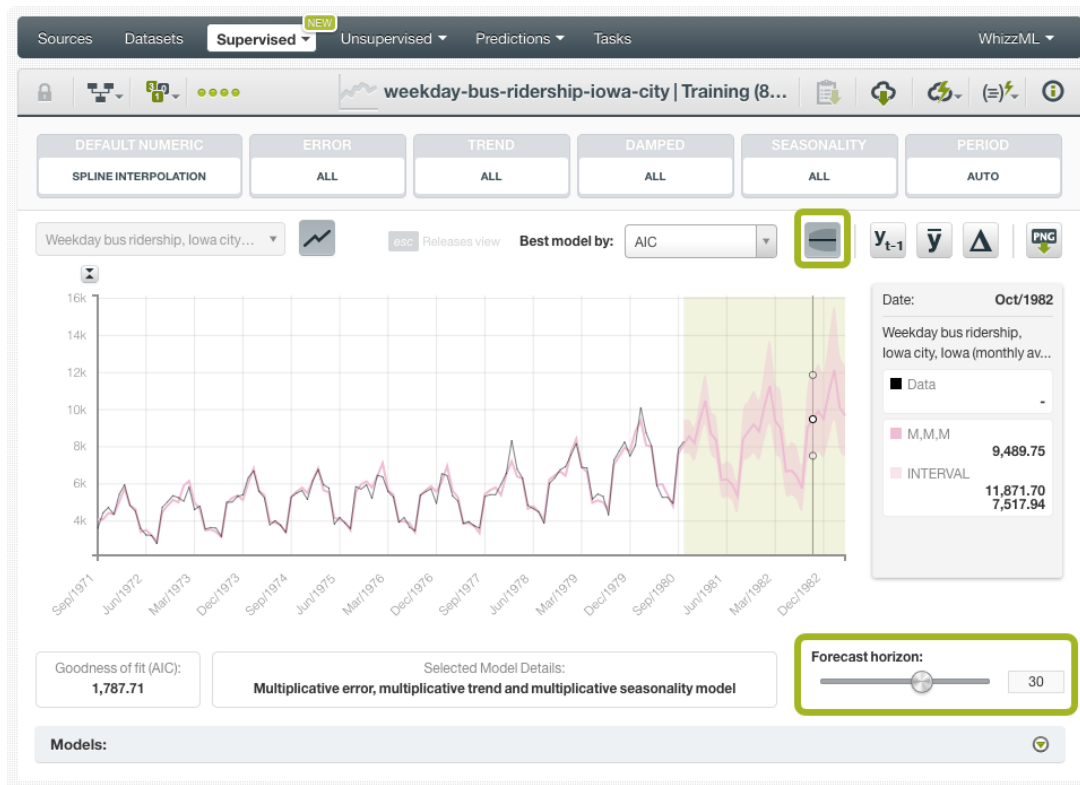


Figure 7.1: Forecast chart

If you configured **dates** for your time series (see [Section 4.2](#)), BigML will automatically compute the corresponding future dates according to the time horizon length and they will be plotted in the x-axis.

When you make forecasts, BigML offers the possibility of showing three benchmark models that work as a baseline for the other models:

- A **mean-based model** that always predicts the mean for all instances in your training dataset.
- A **naive model** that always predicts the last observed value in the dataset.
- A **drift model** that takes the first and last observation and extrapolates the future values.

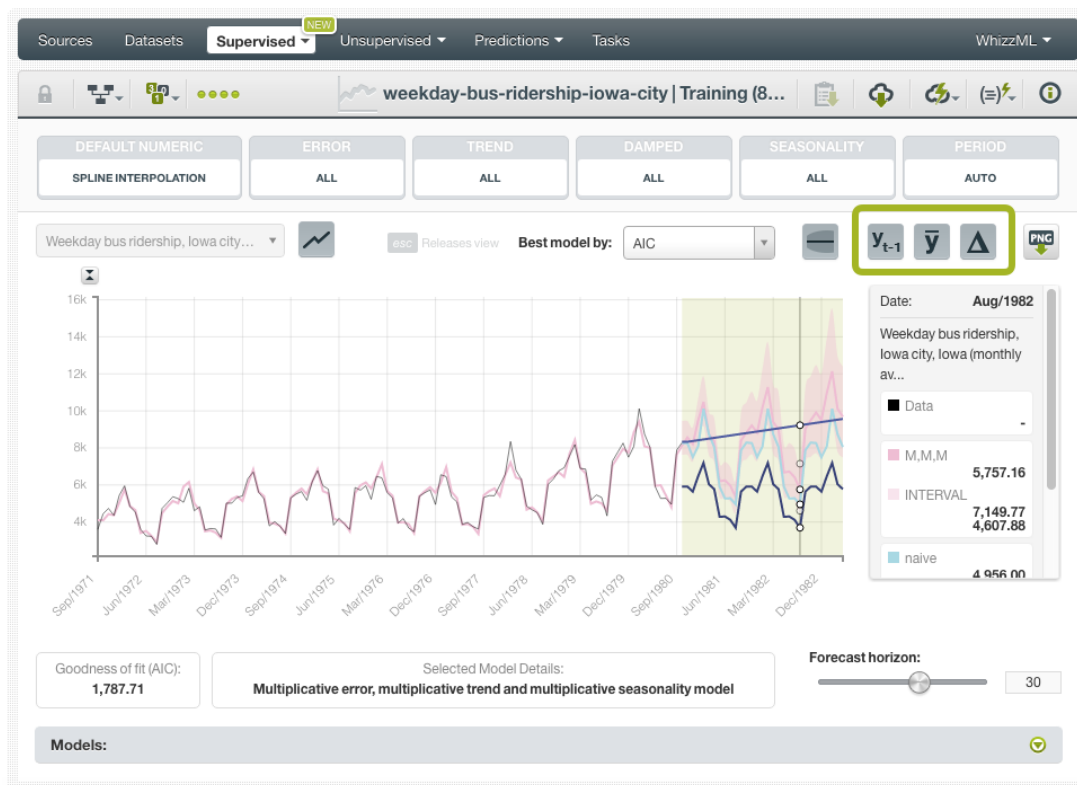


Figure 7.2: Benchmark models for time series

Learn more about these benchmarking methods [here](#)<sup>1</sup>.

## 7.4 Consuming Forecasts

BigML provides several ways for developers to integrate BigML time series forecasts within their predictive applications. In the following sections, we will describe how you can use the BigML REST API and the BigML Python bindings to work with time series forecasts.

### 7.4.1 Using Forecasts Via the BigML API

Time series forecasts have full citizenship in the BigML API. This means you can programmatically create, update, list, delete, and use them for predictions. For example, this is how you can create a forecast using the command line from a given time series and a given time horizon. This will require properly setting the `BIGML_AUTH` environment variable to contain your authentication credentials:

```
curl "https://bigml.io/forecast?${BIGML_AUTH}" \
-X POST \
-H 'content-type: application/json' \
-d '{"timeseries": "timeseries/50650bdf3c19201b64000020",
    "horizon": 24}'
```

For more information on using model predictions through the BigML API, please refer to the [forecast REST API documentation](#)<sup>2</sup>.

<sup>1</sup><https://www.otexts.org/fpp/2/3>

<sup>2</sup><https://bigml.com/api/forecast>

## 7.4.2 Using Forecasts Via the BigML Bindings

BigML bindings provide a convenient way to access the BigML REST API from your language of choice. They offer a higher-level view of BigML Machine Learning resources and algorithms in a number of languages, including Python, Node.js, Java, Swift, and Objective-C. See below an example **to create a forecast in Python** using BigML bindings where “00005” is the ID of the objective field and the “horizon” is the number of data points to forecast:

```
from bigml.api import BigML
api = BigML()
source = api.create_source("my_ts_data.csv")
api.ok(source)
dataset = api.create_dataset(source)
api.ok(dataset)
time_series = api.create_time_series(dataset)
api.ok(time_series)
input_data = {"000005": {"horizon": 10}}
forecast = api.create_forecast(time_series, {"000005": {"horizon": 10}})
```

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

```
from bigml.timeseries import TimeSeries
local_time_series = TimeSeries("timeseries/5968ec46983efc21b000001b")
local_time_series.forecast({"000005": {"horizon": 10}})
```

For more information on using forecasts through the BigML bindings, please refer to the [BigML bindings documentation](#).

## 7.5 Deleting Forecasts

You can DELETE your **time series forecast** by deleting your model. See [Chapter 14](#) for an explanation.

## Consuming Time Series

Similarly, you can also create and consume your time series programmatically via the **BigML API and bindings**. The following subsections explain those options.

### 8.1 Using Time Series Via the BigML API

Time series have full citizenship in the BigML API which allows you to programmatically create, configure, retrieve, list, update, delete, and use them to make forecasts.

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

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

For more information on using time series through the BigML API, please refer to the [time series REST API documentation](#)<sup>1</sup>.

### 8.2 Using Time Series Via the BigML Bindings

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

```
from bigml.api import BigML  
api = BigML()  
timeseries = api.create_timeseries('dataset/57506c472275c1666b004b10')
```

For more information on BigML bindings, please refer to the dedicated [bindings page](#)<sup>2</sup>.

<sup>1</sup><https://bigml.com/api/timeseries>

<sup>2</sup><https://bigml.com/tools/bindings>



---

## Time Series Limits

BigML time series have a few limitations regarding some of the configurable parameters to create models and forecasts. See the corresponding limits listed below:

- **Period length:** a maximum of 60 data points per period for seasonal data is allowed due to computational costs reasons.
- **Forecast horizon:** a maximum of 400 data points per forecast is allowed. It does not make sense to forecast longer horizons for time series since the forecasts rapidly decrease their quality as the horizon increases.

## Time Series Descriptive Information

Time series models have an associated **name**, **description**, **category**, and **tags**. The following subsections briefly describe each concept. See in [Figure 10.1](#) the options under MORE INFO menu to edit time series.

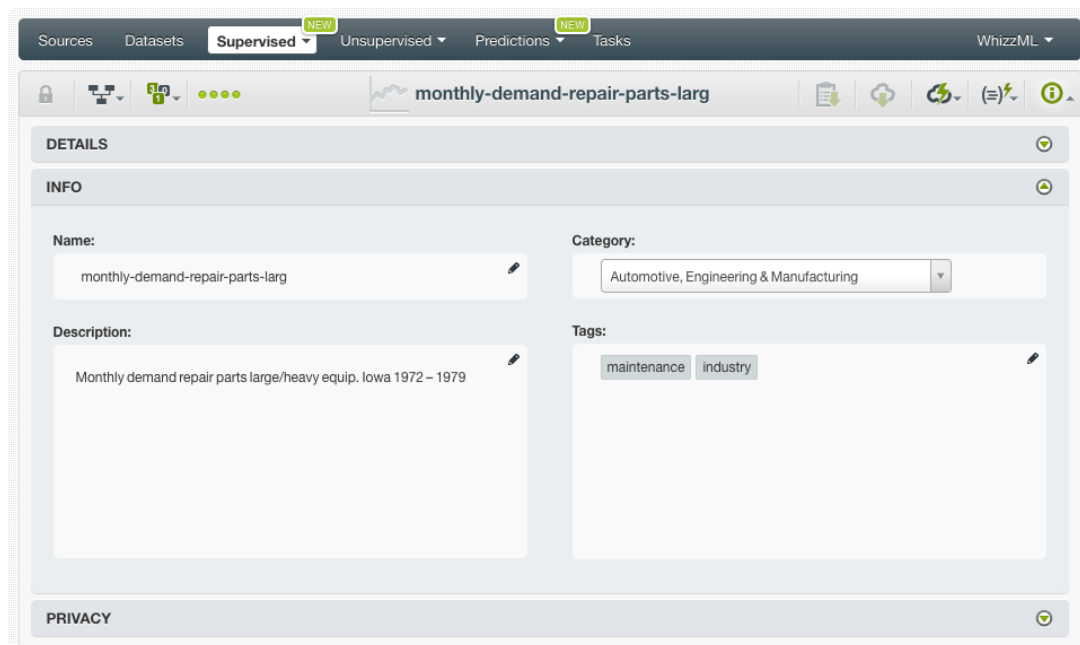


Figure 10.1: Editing time series

### 10.1 Time Series Name

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

### 10.2 Description

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

Descriptions can be written using plain text and also [markdown](#)<sup>1</sup>. BigML provides a simple markdown editor that accepts a subset of markdown syntax. (See [Figure 10.2](#).)

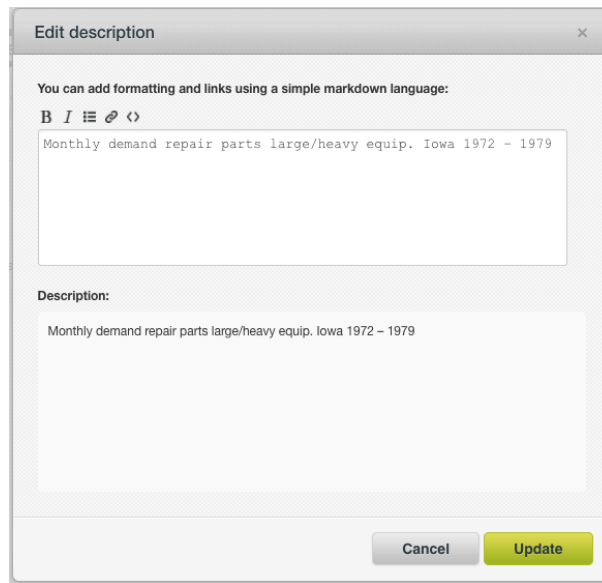


Figure 10.2: Markdown editor for time series descriptions

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

### 10.3 Category

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

A time series category must be one of the **24** categories listed in [Table 10.1](#).

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

Table 10.1: Categories used to classify time series by BigML

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

## 10.4 Tags

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

## 10.5 Counters

For each time series, BigML also stores a number of counters to track the number of other resources that have been created using the time series as a starting point. Display the counters by mousing over the menu option at the top of the time series view. Click on **VIEW # EVALUATIONS FROM THIS TIME SERIES** menu option to quickly access the evaluations. (See [Figure 10.3.](#))

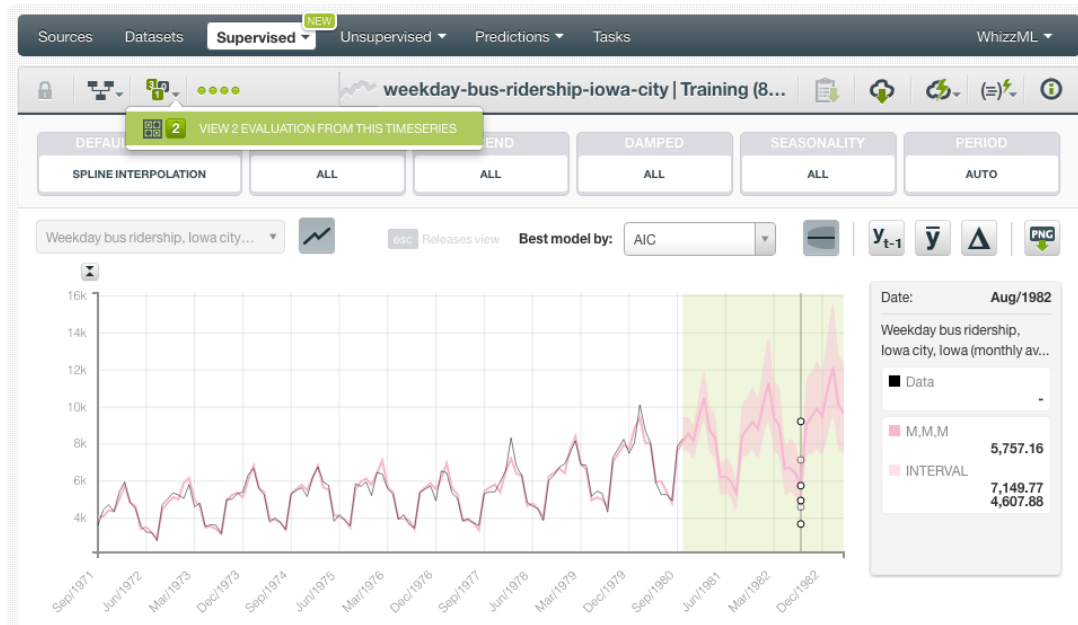


Figure 10.3: Counters for time series

## Time Series Privacy

Privacy options for time series can be defined in the MORE INFO menu option. (See [Figure 11.1](#).) By now, time series are only **private**, i.e., accessible by authorized users (the owner and those who have been granted access by him or her). The **secret link** option to share time series with other users will be available soon.

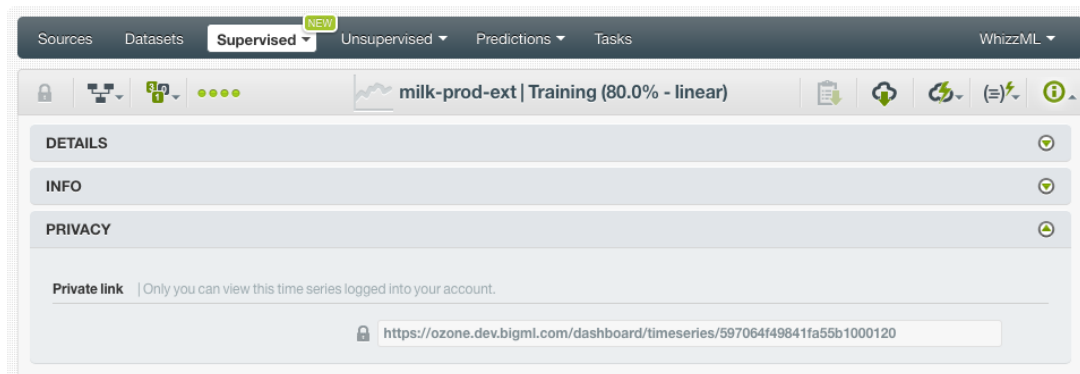


Figure 11.1: Time series privacy

## Moving Time Series

When you create a time series model, it will be assigned to the same **project** where the original dataset is located. Time series can only be assigned to a single project. However, you can move time series models between projects. The menu option to do this can be found in two places:

1. Click **MOVE TO...** within the **1-click action menu** from the time series view. (See [Figure 12.1.](#))

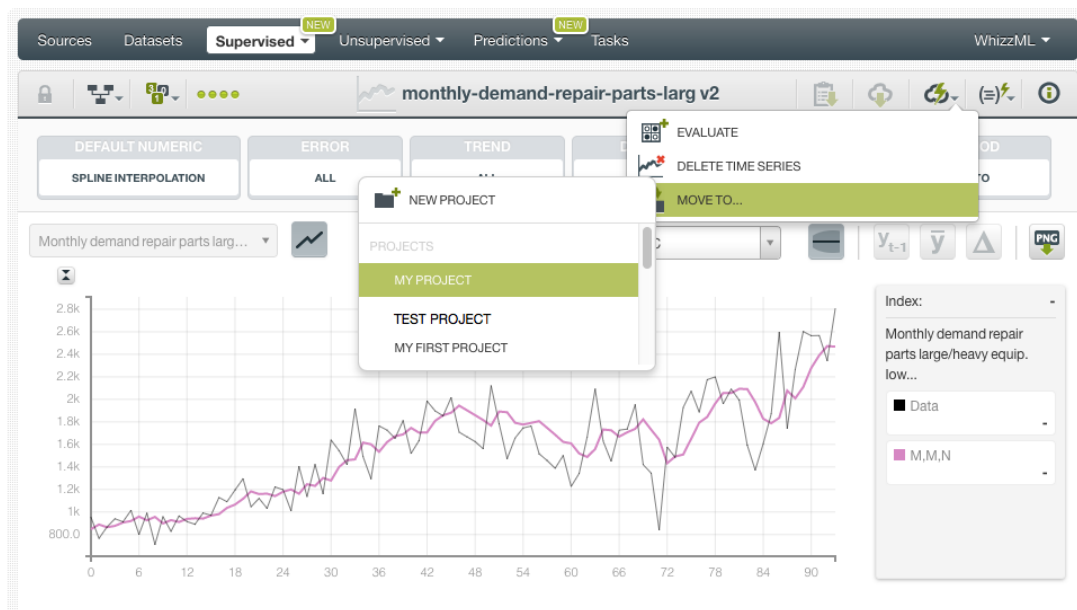


Figure 12.1: Change project from 1-click menu

2. Click **MOVE TO...** within the **pop up menu** from the time series list view. (See [Figure 12.2](#))

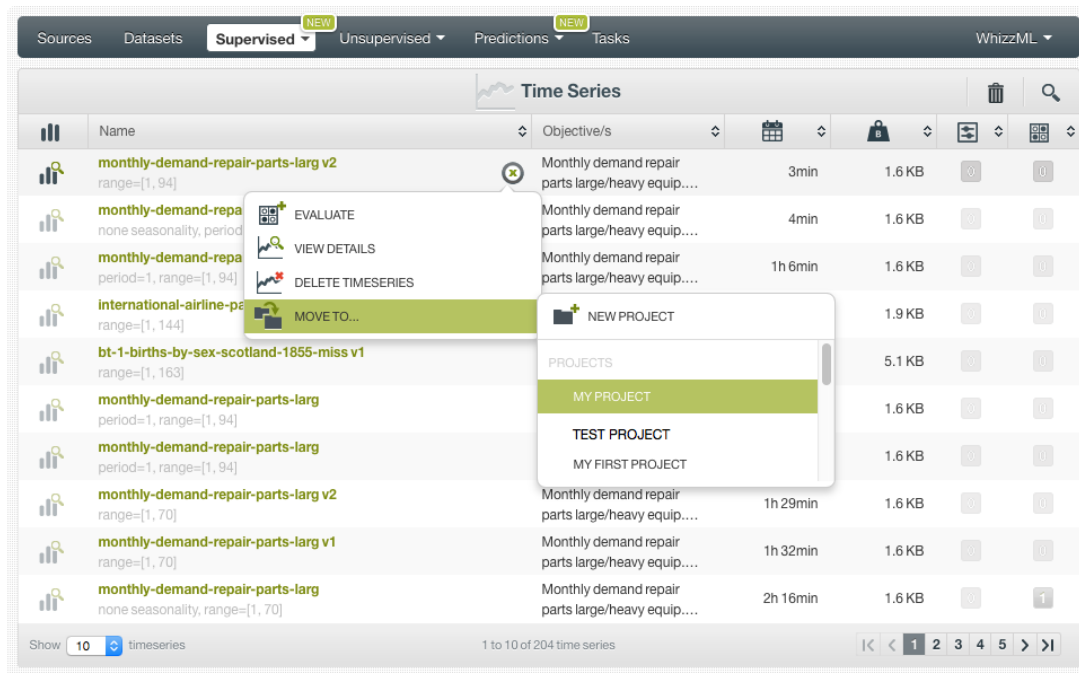


Figure 12.2: Change project from pop up menu



## Stopping Time Series Creation

You can stop the creation of a time series before the task is finished by clicking the **DELETE TIME SERIES** option in the **1-click action menu** from the time series view. (See [Figure 13.1.](#))

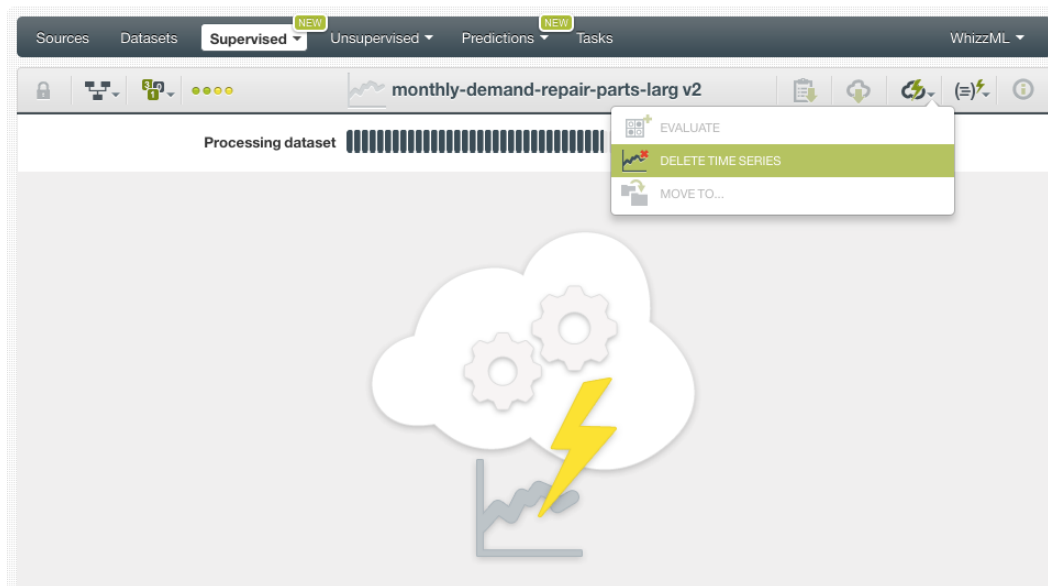


Figure 13.1: Stop time series creation from 1-click menu

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

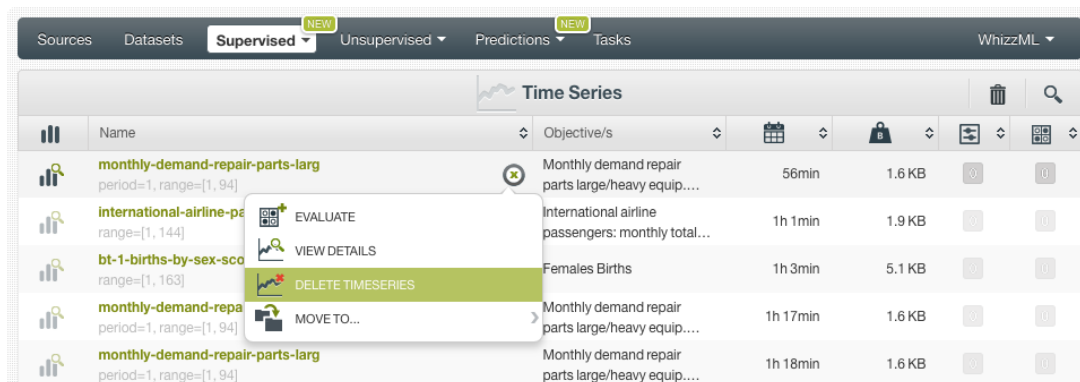


Figure 13.2: Stop time series creation from pop up menu

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

## Deleting Time Series

You can delete your time series by clicking the **DELETE TIME SERIES** option in the **1-click action menu** from the time series view. (See [Figure 14.1.](#))

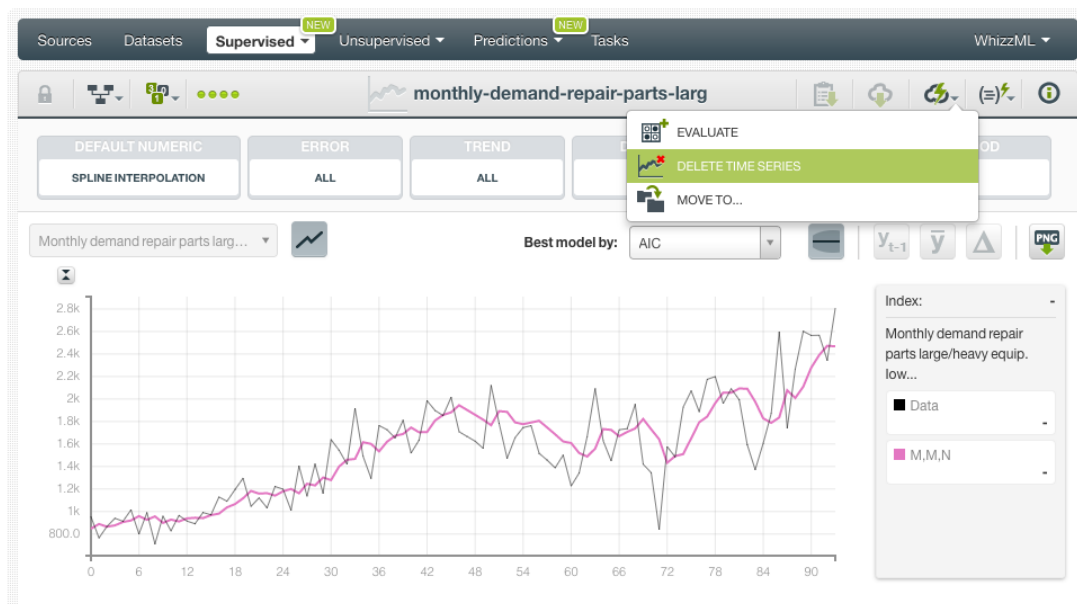


Figure 14.1: Delete time series from 1-click menu

Alternatively, click the **DELETE TIME SERIES** in the **pop up menu** from the time series list view. (See [Figure 14.2.](#))

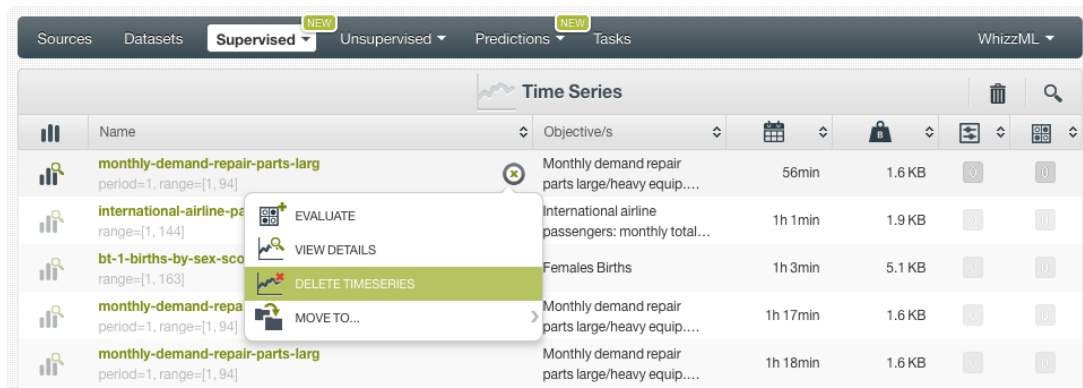


Figure 14.2: Delete time series from pop up menu

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

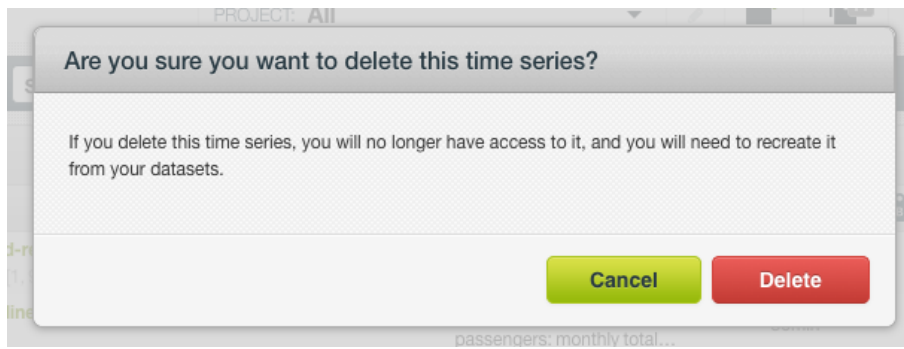


Figure 14.3: Confirmation message to delete a time series

## Takeaways

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

- Time series is a **supervised** learning method used to forecast a numeric field containing time series data.
- BigML time series are an optimized implementation of the **Exponential smoothing** method which can forecast data with trend and seasonal patterns.
- You can set a **forecast horizon** to make predictions with your time series after the models creation.
- You can define one or multiple objective fields to create your time series model.
- Time series models automatically replace missing values using spline interpolation. You can also replace them by other values.
- To create time series you just need an existing **dataset** containing a numeric field with a time-ordered sequence of regularly spaced data points in time. You can also evaluate your model with data that the model has not seen before. Then time series can be used to make forecasts. (See [Figure 15.1](#).)
- You can use the **1-click option** to create your time series or you can **configure** the several parameters provided by BigML.
- When the time series has been created, you get a chart showing the original data along with all the models learned from your training data ranked by the performance metric **AIC**.
- BigML provides different performance metrics by model so you can select the models that will yield most accurate forecasts.
- You can evaluate your time series model and visually analyze the goodness-of-fit in the time series chart.
- You can use your time series model to **forecast** data points in the future.
- You can create, configure, update, and use your time series programmatically via the **BigML API and bindings**.
- You can add **descriptive information** to your time series.
- You can **move** your time series between projects.
- You can **stop** your time series creation by deleting them.
- You can permanently **delete** your existing time series.



Figure 15.1: Time series workflow

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

# List of Figures

|      |  |    |
|------|--|----|
| 1.1  | Time series list view  | 2  |
| 1.2  | Time series empty list view in the BigML Dashboard               | 2  |
| 1.3  | Time series icon   | 2  |
| 3.1  | Create time series from 1-click action menu                      | 7  |
| 3.2  | Create time series from pop up menu                              | 8  |
| 4.1  | Configure time series  | 9  |
| 4.2  | Configure time series objective fields                           | 10 |
| 4.3  | Select a date-time field for the chart                           | 10 |
| 4.4  | Set the dates of your time series                                | 11 |
| 4.5  | Select a default numeric value to replace missing numeric values | 11 |
| 4.6  | Set the horizon for your forecast                                | 12 |
| 4.7  | Configure the model components to create your time series        | 13 |
| 4.8  | Set a range for your time series                                 | 14 |
| 4.9  | Create time series after configuration                           | 15 |
| 4.10 | Time series API request preview                                  | 15 |
| 5.1  | Time series view   | 17 |
| 5.2  | Select the best model by any of the available metrics            | 18 |
| 5.3  | Select the objective field to visualize in the time series chart | 19 |
| 5.4  | See the chart values in the legend                               | 20 |
| 5.5  | See all the models learned from the training data                | 21 |
| 5.6  | Show decomposed model  | 22 |
| 5.7  | Visualize the model components separately                        | 23 |
| 6.1  | 1-click linear train and test split                              | 24 |
| 6.2  | Configure the linear train and test split                        | 25 |
| 6.3  | Evaluations section in the BigML Dashboard                       | 26 |
| 6.4  | Time series evaluations icon                                     | 26 |
| 6.5  | Evaluate time series from 1-click action menu                    | 27 |
| 6.6  | Evaluate time series from pop up menu                            | 27 |
| 6.7  | Evaluate time series using a testing dataset                     | 28 |
| 6.8  | Fields Mapping for evaluations                                   | 29 |
| 6.9  | Configure the range for evaluations                              | 30 |
| 6.10 | Time series evaluations  | 31 |
| 6.11 | Time series evaluation intervals                                 | 32 |
| 6.12 | Benchmark models for time series                                 | 33 |
| 6.13 | Select models to analyze their performance                       | 34 |
| 6.14 | Delete evaluation from the 1-click action menu                   | 37 |
| 6.15 | Delete evaluation from pop up menu                               | 37 |
| 6.16 | Delete evaluation confirmation                                   | 38 |



|   |    |
|---|----|
| 7.1 Forecast chart . . . . .                                | 40 |
| 7.2 Benchmark models for time series . . . . .              | 41 |
| 10.1 Editing time series . . . . .                          | 45 |
| 10.2 Markdown editor for time series descriptions . . . . . | 46 |
| 10.3 Counters for time series . . . . .                     | 48 |
| 11.1 Time series privacy . . . . .                          | 49 |
| 12.1 Change project from 1-click menu . . . . .             | 50 |
| 12.2 Change project from pop up menu . . . . .              | 51 |
| 13.1 Stop time series creation from 1-click menu . . . . .  | 52 |
| 13.2 Stop time series creation from pop up menu . . . . .   | 52 |
| 14.1 Delete time series from 1-click menu . . . . .         | 54 |
| 14.2 Delete time series from pop up menu . . . . .          | 55 |
| 14.3 Confirmation message to delete a time series . . . . . | 55 |
| 15.1 Time series workflow . . . . .                         | 57 |

# List of Tables

10.1 Categories used to classify time series by BigML . . . . . 47

# Glossary

**AIC** (Akaike Information Criterion) measures the trade-off between the model goodness-of-fit and the model complexity to avoid overfitting. The lower the AIC, the better. [17](#), [56](#)

**Damping parameter** a configurable parameter of time series models to “dampen” the trend to a flat line at some point in the future so the trend does not grow indefinitely. [3](#), [13](#)

**Dashboard** The BigML web-based interface that helps you privately navigate, visualize, and interact with your modeling resources. [ii](#), [1](#), [24](#), [25](#)

**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. [24](#)

**Error** represents the unpredictable variations in the time series data, and how they influence observed values. The error is one of the three components of time series models along with the trend and the seasonality. [4](#), [13](#), [17](#), [20](#)

**Forecast** the prediction of a time series model for future data points. [1](#), [3](#), [12](#), [30](#), [39](#)

**Forecast interval** a measure of the quality of the time series forecasts. The interval of the forecast for a future data point sets the upper and lower bounds in which the forecast may lay with a 95% confidence. [4](#), [13](#), [31](#), [39](#)

**Instances** the data points that represent the entity you want to model, also known as observations or examples. They are usually the rows in your data with a value (potentially missing) for each field that describes the entity. [10](#)

**Level** a weighted sum of all the preceding values in a time series model, with the weights being highest for the most recent values, and decreasing exponentially for past instances. [3](#)

**Objective Field** the field that a regression or classification model will predict (also known as target). [9](#), [24](#)

**Overfitting** the process of tailoring the model to fit the training data at the expense of generalization. [6](#), [17](#)

**Predicting** the result of obtaining the objective field value for your new data using an existing model. The model returns the predicted value along with a performance measure (confidence for classification or expected error for regression). [39](#)

**Project** an abstract resource that helps you group related BigML resources together. [2](#), [50](#)

**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. [25](#)

**Seasonality** a pattern of variation that takes over consecutive periods or fluctuations of fixed length. The seasonality is one of the three components of time series models along with the error and the trend. [3](#), [13](#), [17](#), [20](#)

**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](#)

**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. [1](#), [3](#), [39](#)

**Trend** represents the long term trajectory of the time-based data. The trend is one of the three components of time series models along with the error and the seasonality. [3](#), [13](#), [17](#), [20](#)

## References

- [1] The BigML Team. *Association Discovery with the BigML Dashboard*. Tech. rep. BigML, Inc., Dec. 2015.
- [2] The BigML Team. *Classification and Regression with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.
- [3] The BigML Team. *Cluster Analysis with the BigML Dashboard*. Tech. rep. BigML, Inc., May 2016.
- [4] The BigML Team. *Datasets with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.
- [5] The BigML Team. *Sources with the BigML Dashboard*. Tech. rep. BigML, Inc., Jan. 2016.
- [6] The BigML Team. *Topic Models with the BigML Dashboard*. Tech. rep. BigML, Inc., Nov. 2016.

bigml<sup>®</sup>