Eureqa Python API Documentation

Release 1.0

Dennis Oleksyuk

March 30, 2016

Contents

The Eureqa Python API provides a way to automate the entire Eureqa modeling process. Use it to:

- Connect to a data platform
- Automate model creation
- Generate summary reports
- Integrate with other modeling tools
- Deploy production models
- and much more!

The Python API integrates seamlessly with your Eureqa SaaS or Eureqa On Prem environment – access data, models, and analyses built via the API in the Eureqa UI and vice versa.

Please click on one of the links below or in the sidebar for more information.

CHAPTER 1

Getting Started

It is very easy to become productive with the Nutonian Python API - just ensure that you have the prerequisite Python version and Eureqa libraries installed, obtain an access key, and then connect. Details are below.

1.1 Installation

Prerequisites

Install or Verify Python

The Eureqa API requires Python 2.7. Before getting started, verify your Python version or install it if you do not have it.

To verify your Python version in your terminal:

```
python --version
```

If Python is not installed, you can download it here: https://www.python.org.downloads

Install or Verify Pip

The Eureqa Python API is installed using *pip*. If you don't have pip, follow the installation instructions here: https://pip.pypa.io/en/stable/installing/

Eureqa API Installation

Installing the Eureqa API

To install the Eureqa API, use your terminal to run:

```
pip install eureqa[==version]
```

You can find the correct version by opening Eureqa and going to Settings > Eureqa API

Upgrading or Downgrading the Eureqa API

To upgrade or downgrade the API, install again with the version specified, e.g.:

```
pip install eureqa[==version]
```

Note: pip does have "-upgrade" flag which will upgrade to the latest version. We recommend *not* using this flag and instead using the command above to install the specific version that you need.

Uninstalling the Eureqa API

To uninstall the Eureqa API:

```
pip uninstall eureqa
```

1.2 Connecting to Eureqa

To get started with the API, connect to Eureqa as follows:

Eureqa SaaS:

```
from eureqa import Eureqa

e = Eureqa(url='Eureqa URL', user_name='user name', key='access key')
```

Where:

- 'Eureqa URL' is the base URL used to connect to Eureqa (e.g. "https://rds.nutonian.com")
- 'user name' is your Eureqa user name (e.g. "bob@nutonian.com")
- 'access key' is your Eureqa API access key (see below)

Eureqa Local:

```
from eureqa import EureqaLocal

e = EureqaLocal(user_name='user name', key='access key')
```

or, if you have multiple Eureqa Local instances running on your computer:

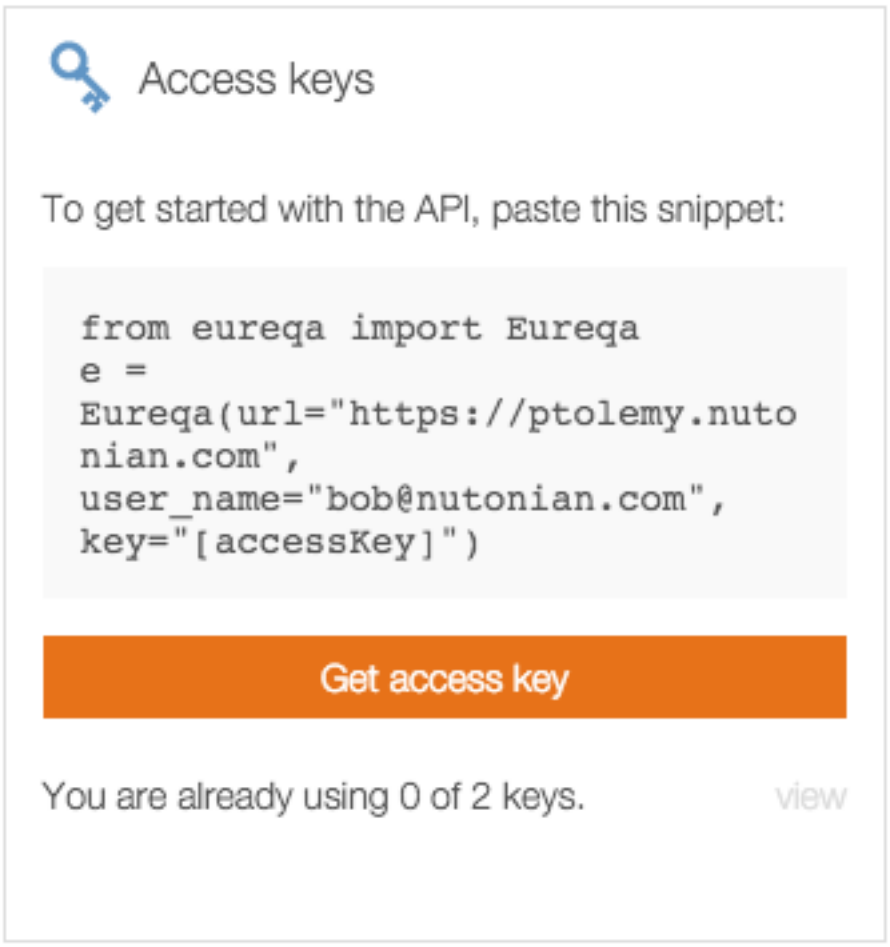
```
from eureqa import EureqaLocal

e = EureqaLocal(instance='instance number', user_name='user name', key='access key')
```

Where:

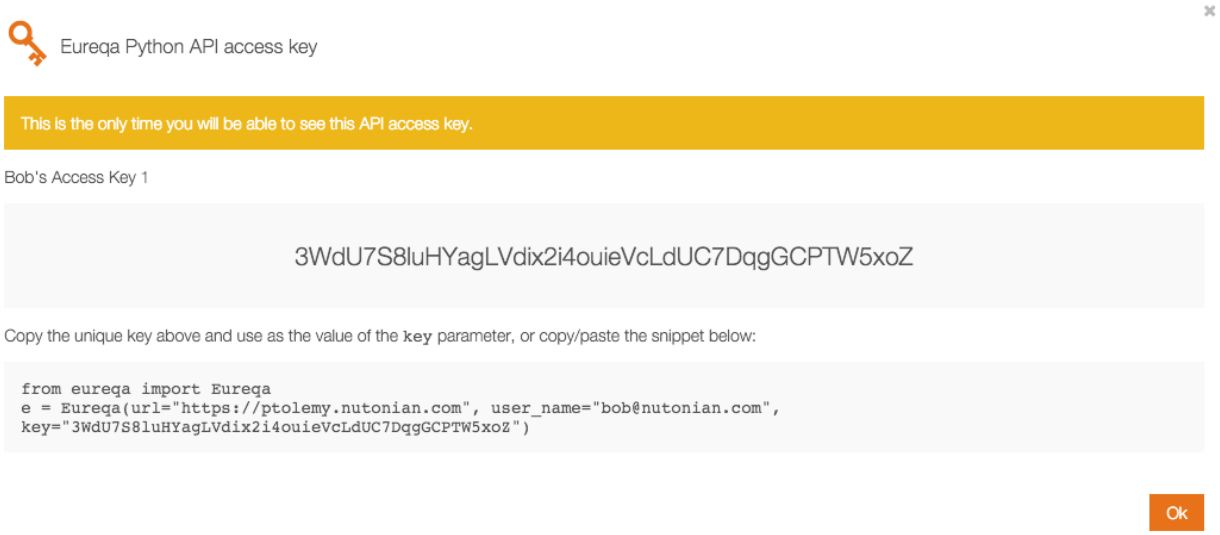
- 'instance number' is a number (first one is '1') indicating the launch order of the instances of Eureqa Local
- 'user name' is your Eureqa user name (e.g. "bob@nutonian.com")
- 'access key' is your Eureqa API access key (see below)

Access Keys

The Eureqa Python API uses key-based authentication. To generate a license key for your user account, open Eureqa and navigate to Settings > Python API and select the option to "Get access key"



Edit the Key name / description if desired and then select Generate Key. Copy and paste the access key into your code – the key will only be shown once!



You may have up to 2 active keys a time. You can delete an existing key and generate a new key at any time.

1.3 Next Steps

Now that you have the Eureqa API installed and have successfully connected to Eureqa, get started uploading data, building models, generating reports and more.

Check out the *Examples* for sample code and the *Eureqa API Reference* for details on specific classes and functions.

CHAPTER 2

Examples

2.1 CreateSimpleSearch

Build a model to forecast sales and build an analysis in Eureqa that displays the results.

The example dataset includes weekly metrics for Sales along with marketing spend in various channels, website activity, and weather data.

Create a connection to Eureqa:

```
from eureqa import Eureqa

e = Eureqa(url="https://rds.nutonian.com", user_name="user@nutonian.com",
           key="JA24SZI94AKV0EJAEVPK")
```

Create a data source:

```
data_source = e.create_data_source("Sales & Marketing", "sales_marketing.csv")
```

```
<eureqa.data_source.DataSource instance at 0x00000000042F24C8>
```

Download sales_marketing.csv.

Create a search. The goal is to forecast Sales using all other variables in the dataset. The time series template default settings will work well for this search:

```
variables = set(data_source.get_variables())
target_variable = "Sales"
settings = e.search_templates.time_series("Sales Forecast", target_variable,
                                          variables - {target_variable})
search = data_source.create_search(settings)
```

Submit the search and wait until the search is done:

```
search.submit(30)
search.wait_until_done()
```

Get the best solution. Eureqa identifies the solution that gives the best trade-off between error and complexity:

```
solution = search.get_best_solution()
print("The best model found is:\n %s = %s" % (solution.target, solution.model))
```

```
The best model found is:
 Sales = 109569817.15572 + 1055.11588490586*delay(Emails_Sent, 1) + 1.0544595112534*wma(delay(Avg_Ter
```

Get the model performance:

```
print("The %s value for this search is %.2f" % (solution.optimized_error_metric,
                                                solution.optimized_error_metric_value))
```

```
The R^2 Goodness of Fit value for this search is 0.79
```

Create an analysis that summarizes the results:

```
analysis = e.create_analysis("Sales Forecasting Model",
                             "This model forecasts sales for next week.")
analysis.create_model_summary_card(solution)
analysis.create_model_fit_by_row_plot_card(solution, title="Model Fit with Sales Forecast",
                   description="View the predicted value for next week by positioning
                                              your mouse over the yellow predicted value.")
analysis.create_model_card(solution, title="Model Details",
                  description="Use the interactive model mode to see how predicted Sales
                            changes as the model inputs change.")
```

```
<eureqa.analysis_cards.model_card.ModelCard at 0x4629668>
```

```
Log into Eureqa to view and interact with the resulting analysis!
```

2.2 CreateCustomSearch

Create a search with customized search settings. Validate the model against a test dataset and get actual and predicted values to be used for additional analysis.

This dataset uses experimental data from a double pendulum. Variables represent the x positions (x1 and x2), velocities (v1 and v2), and accelerations (a1 and a2) of the two arms at different points in time.

Create a connection to Eureqa:

```
from eureqa import Eureqa
from eureqa import error_metric

e = Eureqa(url="https://rds.nutonian.com", user_name="user@nutonian.com",
           key="JA24SZI94AKV0EJAEVPK")
```

Create a data source:

```
data_source = e.create_data_source("Double Pendulum - Training Data",
                                   "double_pendulum_train.csv")
```

```
Download double_pendulum_train.csv.
```

Initialize search settings with the "numeric" template. Target variable is "a2", the acceleration of the second arm of the double pendulum:

```
variables = set(data_source.get_variables())
target_variable = "a2"
settings = e.search_templates.numeric("Model a2", target_variable, variables - {target_variable, 't'}
```

The double pendulum is a physical system. Customize the search settings to disable irrelevant building blocks like if-then-else, and to enable those that are relevant like the trigonometric operators:

```
settings.math_blocks.const.complexity = 1
settings.math_blocks.if_op.disable()  # disable if-then-else
```

```
settings.math_blocks.less.disable()    # disable less
settings.math_blocks.sin.enable(3)     # enable sine and set complexity to 3
settings.math_blocks.cos.enable(3)     # enable cosine and set complexity to 3
settings.error_metric = error_metric.mean_square_error()
```

Create a search and run for 30 seconds:

```
search = data_source.create_search(settings)
search.submit(30)
search.wait_until_done()
```

Get the best model, view the model and the error metrics:

```
solution = search.get_best_solution()
print("The best model found is:\n %s = %s" % (solution.target, solution.model))
```

```
The best model found is:
 a2 = -0.0239994769615275 - a1*cos(x2 - x1) - v1^2*sin(x2 - x1) - 9.81639183106058*sin(x2)
```

Get the model performance:

```
print("The %s value for this search is %.2f" % (solution.optimized_error_metric, solution.optimized_
```

```
The Mean Squared Error value for this search is 0.49
```

Evaluate the model against a test dataset withheld from Eureqa:

```
test_data_source = e.create_data_source("Double Pendulum - Test Data", "double_pendulum_test.csv")
test_metrics = e.compute_error_metrics(test_data_source, target_variable, solution.model)
test_mse = test_metrics.mean_square_error
print("The %s value for the test data is %.2f" % (solution.optimized_error_metric, test_mse))
```

```
The Mean Squared Error value for the test data is 1.34
```

Download double_pendulum_test.csv.

Get the actual and predicted values for a test set and load into a DataFrame for future analysis:

```
predicted_and_actual = e.evaluate_expression(test_data_source, ["t", 'a2', solution.model])
import pandas as pd
df = pd.DataFrame(predicted_and_actual)
df
```

2.3 SearchAutomation

Automate a set of models and summarize model results. In this example we'll test how well we can model each variable in a dataset based on the other variables.

Create a connection to Eureqa:

```
from eureqa import Eureqa

e = Eureqa(url="https://rds.nutonian.com", user_name="user@nutonian.com",
           key="JA24SZI94AKV0EJAEVPK")
```

Get the list of variables in the dataset:

```
data_source = e.create_data_source("Sales & Marketing", "sales_marketing.csv")
variables = set(data_source.get_variables())
variables
```

```
{u'Avg_Temp',
 u'Days_Precipitation',
 u'Days_Rain',
 u'Days_Snow',
 u'Emails_Clicked',
 u'Emails_Opened',
 u'Emails_Sent',
 u'Flyer_Spend',
 u'Inches_Rain',
 u'Inches_Snow',
 u'Max_Temp',
 u'Min_Temp',
 u'Online_Ad_Spend',
 u'Sales',
 u'TV_Ad_Spend',
 u'Web_New_Visitors',
 u'Web_Returning_Visitors',
 u'Web_Total_Visits',
 u'Web_Unique_Visitors',
 u'Week'}
```

Download sales_marketing.csv.

Create a search for each variable, modeling the variable as the target with all other variables as inputs:

```
searches = []
for variable in variables:
    settings = e.search_templates.numeric("Model for variable: %s" % variable,
                                          variable, variables - {variable})
    search = data_source.create_search(settings)
    search.submit(2)
    searches.append(search)
searches
```

```
[<eureqa.search.Search instance at 0x00000000046A3E48>,
 <eureqa.search.Search instance at 0x00000000046A3848>,
 <eureqa.search.Search instance at 0x000000000472BDC8>,
 <eureqa.search.Search instance at 0x00000000046A39C8>,
 <eureqa.search.Search instance at 0x00000000046A3608>,
 <eureqa.search.Search instance at 0x0000000004748E48>,
 <eureqa.search.Search instance at 0x00000000046E6A48>,
 <eureqa.search.Search instance at 0x0000000004701D48>,
 <eureqa.search.Search instance at 0x0000000004748A08>,
 <eureqa.search.Search instance at 0x0000000004748A48>,
 <eureqa.search.Search instance at 0x0000000004701F48>,
 <eureqa.search.Search instance at 0x000000000480A748>,
 <eureqa.search.Search instance at 0x00000000047EA408>,
 <eureqa.search.Search instance at 0x000000000486AB08>,
 <eureqa.search.Search instance at 0x000000000480A588>,
 <eureqa.search.Search instance at 0x0000000004889A48>,
 <eureqa.search.Search instance at 0x000000000486A548>,
 <eureqa.search.Search instance at 0x00000000048FEF88>,
 <eureqa.search.Search instance at 0x0000000004911C88>,
 <eureqa.search.Search instance at 0x00000000048FE888>]
```

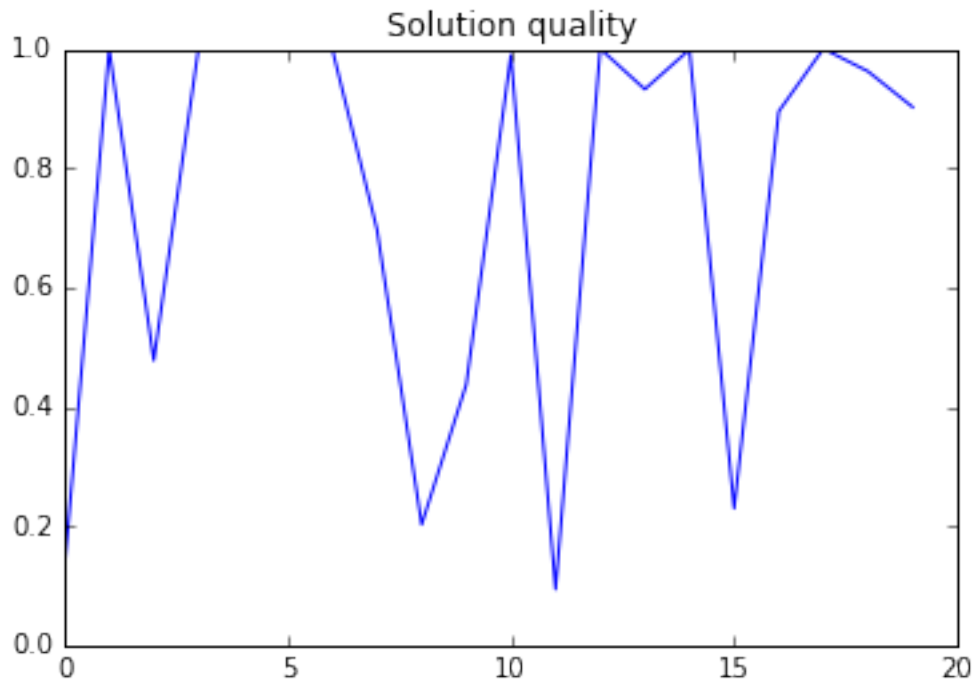
Wait for searches to complete. Print the best solution for each search as they complete:

```
for search in searches:
    search.wait_until_done()
    best_solution = search.get_best_solution()
    print 'Best model for the %s variable is %s' % (best_solution.target, best_solution.model)
```

```
Best model for the Week variable is 0.318191337724807 + 0.648759972160649*step(14.9834893404746 + 224
Best model for the Web_Unique_Visitors variable is 1.04484030337699 + 1.00003420535283*Web_New_Visito
Best model for the Inches_Rain variable is 0.14180331189039 + 0.389285765629462*Days_Rain + 0.012459
Best model for the Web_Returning_Visitors variable is 1.00001733340471*Web_Unique_Visitors - 0.94661
Best model for the Days_Snow variable is Days_Precipitation - Days_Rain
Best model for the Days_Precipitation variable is Days_Rain + Days_Snow
Best model for the Emails_Opened variable is Week + 3.58602233572101*Emails_Clicked + 1.795696851791
Best model for the Sales variable is 88547999.0622628 + 553925.992067077*Avg_Temp + 57999.4727128087
Best model for the Inches_Snow variable is Days_Snow
Best model for the Online_Ad_Spend variable is 324756.430393052 + Emails_Opened*Emails_Clicked*less(1
Best model for the Emails_Sent variable is 4.88394640914473*Emails_Opened + 1.36338463476562*min(Emai
Best model for the TV_Ad_Spend variable is 230229.215764672 + 11226.1289950602*Inches_Snow + 5367.927
Best model for the Web_New_Visitors variable is 0.999992737715856*Web_Unique_Visitors - 0.979362480
Best model for the Avg_Temp variable is 36.4457896586784 + 1.52605710485033*Max_Temp + 6.714859448643
Best model for the Emails_Clicked variable is 4.70332166064393 + 0.249648462804642*Emails_Opened - 0.
Best model for the Flyer_Spend variable is 121967.639533629 + 1.27247056501962*Web_New_Visitors*Week
Best model for the Min_Temp variable is if(15.2531317411839, Avg_Temp, 15.2531317411839) - 7.4821757
Best model for the Days_Rain variable is Days_Precipitation - Days_Snow
Best model for the Web_Total_Visits variable is 15.6700031160911*Emails_Opened + 7.8982810405798*Web_
Best model for the Max_Temp variable is 8.63740515122003 + max(1 + 0.961215843860963*Avg_Temp, Min_Te
```

Plot the error metric values to show which variables are easiest and hardest to predict:

```
%matplotlib inline
import matplotlib.pyplot as plt

error_metric = searches[0].error_metric
solutions = [search.get_best_solution() for search in searches]

plt.plot(range(len(solutions)), [s.get_error_metric_value(error_metric) for s in solutions])
plt.title('Solution quality')
plt.show()
```



2.4 AnalysisCustomPlot

Build two custom plots and add them to an analysis. Custom plot functionality is currently in beta.

Create a connection to Eureqa:

```
from eureqa import Eureqa
from eureqa.analysis_cards import Plot

e = Eureqa(url="https://rds.nutonian.com", user_name="user@nutonian.com",
           key="JA24SZI94AKV0EJAEVPK")

analysis = e.create_analysis('Custom Plot Analysis')
```

Create a plot with three components. The X and Y positions for these components are specified directly using Python lists.

```
plot = Plot()
plot.plot([1,2,3], [4,5,6], color='blue', line_width=3, legend_label='Label 1')
plot.plot([11,22,33], [44,55,66], style='o', legend_label='Label 2',
          color='red', circle_radius=7)
plot.plot([10,11,12], [13,14,15], line_width=5, color='orange')
plot_card = analysis.create_custom_plot_card(plot, 'This card contains my custom plot with data from
```

Create a second plot with two components. The X and Y positions for these components come from variables within a Eureqa data source. The "<row>" variable is the row number from the data source.

```
datasource = e.create_data_source('Nutrition Data', 'nutrition.csv')
plot2 = Plot(width='500px', height='500px', x_axis_label='x axis label', y_axis_label='y axis label')
plot2.plot('<row>', 'Calories', datasource=datasource, style='o', line_width=3,
           circle_radius=5, color='blue', legend_label='Calories')
plot2.plot('<row>', '30*(Fat (g))', datasource=datasource, style='-', line_width=3,
```

```
            circle_radius=5, color='red', legend_label='30*(Fat (g))')
plot_card_2 = analysis.create_custom_plot_card(plot2, 'This plot contains data from my Nutrition data
```

`Download nutrition.csv`.

Log into Eureqa and navigate to the Analysis area to view the custom plots.

CHAPTER 3

Eureqa API Reference

3.1 analysis

class eureqa.analysis.**Analysis**(*body*, *eureqa*)

Represents an analysis on the server.

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **body** (*dict*) – Class metadata as dictionary.

Variables

- ***name*** (*str*) – The analysis name.
- ***description*** (*str*) – The analysis description.

add_card(*card*)

Add an AnalysisCard to this analysis

Parameters **card** (eureqa.analysis_cards.AnalysisCard) – Card to add to the analysis. Must not already be associated with an AnalysisCard.

analysis_id

The id of the analysis

create_binned_mean_plot_card(*datasource*, *x_var*, *y_var*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Creates a new binned-mean-plot card. Adds the card to this analysis.

Parameters

- **datasource** (*DataSource*) – Data source for the card's data
- **x_var** (*str*) – The X-axis variable for the card's plot
- **y_var** (*str*) – The Y-axis variable for the card's plot
- **title** (*str*) – The title of the card
- **description** (*str*) – The card's description.
- **needs_guides** (*bool*) – Whether the card needs guides
- **axis_labels** (*list*) – Axis labels for this card's plot
- **label_format** (*list*) – Label format for this card

- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *BinnedMeanPlotCard*

create_box_plot_card(*datasource*, *x_var*, *y_var*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Creates a new box-plot card. Adds the card to this analysis.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – The X-axis variable for the card's plot
- **y_var** (*str*) – The Y-axis variable for the card's plot
- **title** (*str*) – The title of the card
- **description** (*str*) – The description of the card
- **needs_guides** (*bool*) – Whether the card needs guides
- **axis_labels** (*list*) – Axis labels for this card's plot
- **label_format** (*str*) – Label format for this card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *BoxPlotCard*

create_by_row_plot_card(*datasource*, *x_var*, *plotted_vars*, *title=None*, *description=None*, *focus_variable=None*, *should_center=True*, *should_scale=False*, *collapse=False*)

Create a new by-row plot card. Adds the card to this analysis.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – Name of the variable to plot as the X axis
- **plotted_vars** (*str*) – List of string-names of variables to plot. (To modify a variable's display name, first create the card; then modify the display name directly on it.)
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **focus_variable** (*str*) – Name of the variable in 'plotted_vars' to bring to the foreground
- **should_center** (*bool*) – Should the plot be centered?
- **should_scale** (*bool*) – Should the plot scale?
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *ByRowPlotCard*

create_custom_plot_card(*plot*, *title=None*, *description=None*, *collapse=False*)

Beta

Creates a new custom plot card. Adds the card to this analysis.

Parameters

- **plot** (eureqa.analysis_cards.Plot) – The Plot to be displayed in the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *CustomPlotCard*

create_distribution_plot_card(*data_source*, *variable*, *title=None*, *description=None*, *collapse=False*)

Creates a new distribution plot card. Adds the card to this analysis.

Parameters

- **data_source** (*eureqa.DataSource*) – The data source to which the variable belongs.
- **variable** (*str*) – The name of the variable that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *DistributionPlotCard*

create_double_histogram_plot_card(*datasource*, *x_var*, *y_var*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Creates a new box-plot card. Adds the card to this analysis.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – The X-axis variable for the card's plot
- **y_var** (*str*) – The Y-axis variable for the card's plot
- **title** (*str*) – The title of the card
- **description** (*str*) – The description of the card
- **needs_guides** (*bool*) – Whether the card needs guides
- **axis_labels** (*list*) – Axis labels for this card's plot
- **label_format** (*str*) – Label format for this card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *DoubleHistogramPlotCard*

create_html_card(*html*, *title='HTML'*, *description=None*, *collapse=False*)
Creates a new HTML card. Adds the card to this analysis.

Parameters

- **html** (*str*) – The card's HTML body.
- **title** (*str*) – The card title.
- **description** (*str*) – Deprecated; unused. (This card doesn't have a Description.)
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *HtmlCard*

create_image_card(*image_path*, *title=None*, *description=None*, *collapse=False*)
Creates a new text card containing only header text and one image.

Parameters

- **image_path** (*str*) – the filepath to the image in your filesystem.
- **title** (*str*) – The card title.
- **description** (*str*) – a description of the card, to appear above the image
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *TextCard*

create_model_card(*solution*, *title=None*, *description=None*, *collapse=False*)
Creates a new model card. Adds the card to this analysis.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *ModelCard*

create_model_evaluator_card(*datasource*, *solution*, *title=None*, *description=None*, *collapse=False*)
Creates a new model evaluator card. Adds the card to this analysis.

Parameters

- **datasource** (*eureqa.DataSource*) – DataSource to fetch data from for the primary model evaluator
- **solution** (*eureqa.Solution*) – Solution to fetch results from for the primary model evaluator
- **title** (*str*) – Title of the card. Defaults to 'Evaluate Model'.
- **description** (*str*) – Description of the card.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card.

Return type *ModelEvaluatorCard*

Additional models can be added to this model by calling the `add_solution_info` method on the returned object.

create_model_fit_by_row_plot_card(*solution*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Create a new model fit by-row plot card. Adds the card to this analysis. Note that by-row plots are meant for use with Numeric searches. They may not work properly if used with other types of searches.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *ModelFitByRowPlotCard*

create_model_fit_plot_card(*solution*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Create a new model fit card. Adds the card to this analysis. Automatically choose the correct type of card (by-row plot or separation plot) based on the specified search. Numeric searches use by-row plots; time-series searches use separation plots.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *ModelFitPlotCard*

create_model_fit_separation_plot_card(*solution*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Create a new model fit separation-plot card. Adds the card to this analysis. Note that separation plots are meant for use with time-series searches. They may not work properly if used with other types of searches.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *ModelFitSeparationPlotCard*

create_model_summary_card(*solution*, *collapse=False*)
Creates a new model summary card. Adds the card to this analysis.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *ModelSummaryCard*

create_scatter_plot_card(*datasource*, *x_var*, *y_var*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)
Creates a new scatter-plot card. Adds the card to this analysis.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – The X-axis variable for the card's plot
- **y_var** (*str*) – The Y-axis variable for the card's plot
- **title** (*str*) – The title of the card
- **description** (*str*) – The card's description
- **needs_guides** (*bool*) – Whether the card needs guides
- **axis_labels** (*list*) – Axis labels for this card's plot
- **label_format** (*list*) – Label format for this card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns Object representing the created card

Return type *ScatterPlotCard*

create_text_card(*text*, *title='Text'*, *description=None*, *collapse=False*)
Creates a new text card. Adds the card to this analysis.

Parameters

- **text** (*str*) – The card text.
- **title** (*str*) – The card title.
- **description** (*str*) – Deprecated; unused. (This card doesn't have a Description.)
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Returns An object that represents the newly created card.

Return type *TextCard*

delete()
Deletes the analysis from the server.

description
The description of the analysis

get_cards()
Returns the cards belonging to the analysis.

Returns A list of the card objects from *analysis*.

Return type list of `AnalysisCard`

name
The name of the analysis

upload_image(*image_path*)
Upload an image to the server, to be embedded in analysis cards.

Parameters **image_path** (*str*) – the filepath to the image on your filesystem.

Returns An object that represents the newly uploaded image.

Return type Image

3.2 analysis_cards

class eureqa.analysis_cards.**AnalysisCard**
The base class for all card classes. API returns an instance of this class if it cannot recognize the type of the card that it receives from the server.

Don't construct this class directly. Instead, construct any class that inherits from it.

Parameters **object** – A configuration object used internally

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

copy()
Duplicates this AnalysisCard. If the current card is associated with an Analysis, the new card will not be associated with that Analysis and can be added to it or to any other Analysis. :return: A copy of this AnalysisCard.

delete()
Deletes the card from the server.

classmethod from_json(*cls*, *body*, *eureqa=None*)
Construct a new AnalysisCard from the output of 'to_json()'

Parameters

- **cls** (*obj*) – Classmethod object
- **body** (*dict*) – Content of the card
- **eureqa** (Eureqa) – Eureqa-API instance

Returns AnalysisCard

move_above(*other_card*)
Moves this card above another card.

Parameters **other_card** (eureqa.analysis_cards.AnalysisCard) – The other card above which to move this card.

move_below(*other_card*)
Moves this card below another card.

Parameters **other_card** (eureqa.analysis_cards.AnalysisCard) – The other card object below which to move this card.

replace(*other*)
Replace this card's contents with the contents of *other*

Parameters **other** (*eureqa.analysis_card.AnalysisCard*) – Card to replace our contents with

to_json()
Convert this AnalysisCard to a JSON-serializable structure

Returns A representation of this card's contents as primitive Python objects

class eureqa.analysis_cards.**DistributionPlotCard**(*datasource=None*, *variable=None*, *title=None*, *description=None*, *collapse=False*)
Represents a distribution plot card on the server.

Parameters

- **datasource** (*eureqa.DataSource*) – The data source to which the variable belongs.
- **variable** (*str*) – The name of the variable that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card title.
- **datasource** (*str*) – The datasource used by the card.
- **variable** (*str*) – The variable plotted by the coard.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**ModelCard**(*solution=None*, *title=None*, *description=None*, *collapse=False*)
Represents a model card on the server.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card description.

- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card title.
- ***description*** (*str*) – The card description.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

solution

Returns Solution that the card is rendering

Return type *Solution*

title
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**ModelSummaryCard**(*solution=None*, *title=''*, *description=None*, *collapse=False*)
Represents a model summary card on the server.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **title** (*str*) – Title of the card
- **description** (*str*) – Title of the card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***title*** (*str*) – The card title.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

solution

Returns Solution that the card is rendering

Return type *Solution*

title
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**TextCard**(*text=''*, *title='Text'*, *description=None*, *collapse=False*)
Represents a text card on the server.

Parameters

- **title** (*str*) – Title of the card
- **description** (*str*) – Description of the card
- **text** (*str*) – Body of the card. (Markdown-formatted text.)
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***text*** (*str*) – The card text.

delete()
Deletes the card from the server.

If the card contains any images, they will be deleted.

description
The description of this card.

Returns description of this card

Return type str

text
The text contents of this card.

Returns text contained in this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**HtmlCard**(*html=''*, *title='HTML'*, *description=None*, *collapse=False*)
** BETA ** Represents a text card on the server.

Parameters

- **title** (*str*) – Title of the card
- **description** (*str*) – Description of the card
- **html** (*str*) – Body of the card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***text*** (*str*) – The card text.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

html
The body of this card.

Returns body of this card

Return type str

text
The text contents of this card.

Returns text contained in this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**ModelFitByRowPlotCard**(*solution=None*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Represents a projection card on the server.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***x_axis*** (*str*) – The card's X axis

class eureqa.analysis_cards.**ModelFitSeparationPlotCard**(*solution=None*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Represents a centipede plot card on the server.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***x_axis*** (*str*) – The card's X axis

class eureqa.analysis_cards.**ModelEvaluatorCard**(*title=None*, *description=None*, *collapse=False*, *solution_infos=None*)

Represents an evaluate model card on the server.

Parameters

- **title** (*str*) – Title of the card. Defaults to 'Evaluate Model'.
- **description** (*str*) – Description of the card.
- **collapse** (*bool*) – Whether the card should default to be collapsed.
- **solution_infos** (*list(eureqa.analysis_cards.ModelEvaluatorCard.SolutionInfo)*) – List of solutions that this card is rendering

Variables

- ***title*** (*str*) – The title of the card
- ***solution_infos*** (*tuple*) – Immutable ordered set of SolutionInfo objects representing the solutions shown on this card.

class **SolutionInfo**(*datasource=None*, *solution=None*)

The solution information for a single model evaluation ("tab") on ModelEvaluatorCard

Parameters

- **datasource** (eureqa.data_source.DataSource) – Data source for this solution
- **solution** (eureqa.solution.Solution) – Solution

Variables

- ***dataset_id*** (*str*) – ID of the DataSet referenced by this solution-tab
- ***datasource_id*** (*str*) – ID of the DataSource referenced by this solution-tab
- ***search_id*** (*str*) – ID of the Search referenced by this solution-tab
- ***solution_id*** (*str*) – ID of the Solution referenced by this solution-tab
- ***has_target_variable*** (*bool*) – Whether the dataset contains the target variable

dataset_id

ID of the DataSet used for this solution

Returns unique DataSet identifier

datasource_id

ID of the DataSource used for this solution

Returns unique DataSource identifier

has_target_variable

Whether the dataset contains the target variable

Returns Whether the dataset contains the target variable

search_id

ID of the Search used for this solution

Returns unique Search identifier

solution_id
ID of the Solution used for this solution
Returns unique Solution identifier

ModelEvaluatorCard.**add_solution_info**(*datasource*, *solution*)
Add a new (non-default) solution and tab to this card. Once added, this object will show up in the list returned by *solution_infos*.

Parameters

- **datasource** (DataSource) – DataSource used with this solution.
- **solution** (Solution) – Solution associated with the model evaluation.

ModelEvaluatorCard.**collapse**
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

ModelEvaluatorCard.**description**
The description of this card.

Returns description of this card

Return type str

ModelEvaluatorCard.**solution_infos**
The set of all SolutionInfo objects associated with this card. One per solution tab displayed in the UI. Note that *solution_infos[0]* is the default card; it may be treated specially by the UI.

Returns List or tuple of SolutionInfo objects

ModelEvaluatorCard.**title**
The title of this card.

Returns title of this card

Return type str

class eureqa.analysis_cards.**BoxPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)
Represents a box plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

class eureqa.analysis_cards.**DoubleHistogramPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a double-histogram plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

class eureqa.analysis_cards.**ScatterPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a scatter plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

class eureqa.analysis_cards.**BinnedMeanPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a binned mean plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.

- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

class eureqa.analysis_cards.**ByRowPlotCard**(*datasource=None*, *x_var=None*, *plotted_vars=None*, *title=None*, *description=None*, *focus_variable=None*, *should_center=True*, *should_scale=False*, *collapse=False*)

Represents a variable line graph on the server.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – Name of the variable to plot as the X axis
- **plotted_vars** (*str*) – List of string-names of variables to plot. (To modify a variable's display name, first create the card; then modify the display name directly on it.)
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **focus_variable** (*str*) – Name of the variable in 'plotted_vars' to bring to the foreground
- **should_center** (*bool*) – Should the plot be centered?
- **should_scale** (*bool*) – Should the plot scale?
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***focus_variable*** (*str*) – Focused (foreground) variable for the card. Must be a member of ~eureqa.analysis_cards.ByRowPlotCard.plotted_variables
- ***x_var*** (*str*) – Name of the variable to plot as the X axis
- ***plotted_variables*** (*list*) – Variables to plot. (List of string variable names.)
- ***should_center*** (*bool*) – Should the plot be centered?
- ***should_scale*** (*bool*) – Should the plot scale?

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

datasource
The data source providing data for this card

Returns data source providing data for this card

description
The description of this card.

Returns description of this card

Return type str

focus_variable
The variable that is currently in focus (in the foreground) for this card. Must be a member of 'plotted_variables'.

Returns focus_variable for this card

Return type str

plotted_variables
The plotted variables for this card.

Returns List of the names of the variables being plotted against the X axis

Return type tuple

should_center
The should_center option for this card.

Returns whether this plot should be centered

Return type bool

should_scale
The should_scale option for this card.

Returns whether this plot should be scaled

Return type bool

title
The title of this card.

Returns title of this card

Return type str

x_var
The X-axis variable for this card

Returns the name of the X-axis variable for this card

Return type str

class eureqa.analysis_cards.**Plot**(*width=None*, *height='400px'*, *x_axis_label=None*, *y_axis_label=None*, *show_legend=True*, *zero_x_line=False*, *zero_y_line=False*, *x_tick_format=None*, *y_tick_format=None*, *num_x_ticks=None*, *num_y_ticks=None*, *max_displayed_points=200*, *guides_type='XY'*)

Beta

Represents a plot to be displayed in an analysis plot card.

By calling the plot() method multiple times on one instance of this class, multiple lines and scatter plots can be superimposed on this object.

Parameters

- **width** (*str*) – Set manual dimensions for the plot in "px" units (e.g. 350px). Defaults to full panel width.
- **height** (*str*) – Set manual dimensions for the plot in "px" units. Defaults to a constant height of 400px.
- **x_axis_label** (*str*) – The label to use on the x axis of the plot.
- **y_axis_label** (*str*) – The label to use on the y axis of the plot.
- **show_legend** (*bool*) – Controls whether or not a legend will be used in the plot.
- **zero_x_line** (*bool*) – Draws a horizontal line through the origin
- **zero_y_line** (*bool*) – Draws a vertical line through the origin
- **x_tick_format** (*str*) – Format for x axis tick labels. Valid values are "date" or anything supported by D3: https://github.com/mbostock/d3/wiki/Formatting#d3_format. Defaults to our internal number format
- **y_tick_format** (*str*) – Format for y axis tick labels. Valid values are "date" or anything supported by D3: https://github.com/mbostock/d3/wiki/Formatting#d3_format. Defaults to our internal number format
- **num_x_ticks** (*int*) – Number of x axis tick labels. Defaults to as many as fit comfortably without overlap
- **num_y_ticks** (*int*) – Number of y axis tick labels. Defaults to as many as fit comfortably without overlap
- **max_displayed_points** (*int*) – Maximum number of data points to sample for each component
- **guides_type** (*str*) – The type of value-guides to show when hovering over a point in the graph. Valid values are "XY", "YY" or False. XY guides will show the x and y values for the point under the cursor. YY guides will show the x and y values of each component for the data point closest to the cursor. False will turn off value guides.

delete()

Delete the Plot and any associated data which has been uploaded.

plot(*x*, *y*, *datasource=None*, *style='-'*, *color='black'*, *line_width=1*, *circle_radius=1*, *use_in_plot_range=True*, *error_bars_upper_values=None*, *error_bars_lower_values=None*, *legend_label=None*, *tooltip_template=None*)

Add new data to the plot with the specified options.

If a datasource is specified, x and y can be provided as expressions in terms of the variables in that datasource. Otherwise, x and y must be lists of data of identical length.

Parameters

- **x** (*list*) – X axis data. Either a list of input values, or if a datasource is specified, an expression in terms of that datasource's variables.
- **y** (*list*) – Y axis data. Either a list of input values, or if a datasource is specified, an expression in terms of that datasource's variables.
- **datasource** (DataSource) – If provided, a specific DataSource to use as a base for the variables to be plotted.

- **style** (*str*) – A matplotlib-like style string of 'o' or '-' or '-o' to indicate circle, line, or line-circle, respectively.
- **color** (*str*) – CSS color.
- **line_width** (*int*) – The width of the line, if applicable.
- **circle_radius** (*int*) – The radius of each circle, if applicable.
- **use_in_plot_range** (*bool*) – The chart auto-computes the x and y axis ranges based on the data for each component. If you want to add a component, but have its data not be used to compute the x and y axis ranges, set this value to False. For example, if you want to make a scatter plot, with a trend line going through it, then setting this field to False for the trend-line component will make the chart snugly fit the points, with the trend line extending beyond.
- **error_bars_upper_values** (*list*) – The column to use for the tops of error bars
- **error_bars_lower_values** (*list*) – The column to use for the bottoms of error bars
- **legend_label** (*str*) – A string label to be used in the legend. If unspecified, this plot will not appear in the legend.
- **tooltip_template** (*str*) – Any valid Handlebars template. The template will be provided the x and y values as "x_value" and "y_value", so you can use these to format a tooltip for each point. For example, " When age is {{x_value}}, salary is expected to be {{y_value}} "

upload_data(*eureqa*)

Upload the plot data to eureqa. This is required before the plot can be viewed.

Parameters **eureqa** (Eureqa) – a eureqa connection

class eureqa.analysis_cards.**CustomPlotCard**(*plot=None*, *title=None*, *description=None*, *collapse=False*)

Represents a card which can display a custom plot.

Parameters

- **plot** (eureqa.analysis_cards.Plot) – The Plot to be displayed in the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***plot*** (eureqa.analysis_cards.Plot) – the underlying Plot displayed by this card.

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

delete()

Deletes the card from the server.

This will delete any data which has been uploaded internally to support plotting.

description

The description of this card.

Returns description of this card

Return type str

plot
The underlying Plot displayed by this card.

title
The title of this card.

Returns title of this card

Return type str

3.2.1 analysis_card

class eureqa.analysis_cards.analysis_card.**AnalysisCard**
The base class for all card classes. API returns an instance of this class if it cannot recognize the type of the card that it receives from the server.

Don't construct this class directly. Instead, construct any class that inherits from it.

Parameters **object** – A configuration object used internally

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

copy()
Duplicates this AnalysisCard. If the current card is associated with an Analysis, the new card will not be associated with that Analysis and can be added to it or to any other Analysis. :return: A copy of this AnalysisCard.

delete()
Deletes the card from the server.

classmethod **from_json**(*cls*, *body*, *eureqa=None*)
Construct a new AnalysisCard from the output of 'to_json()'

Parameters

- **cls** (*obj*) – Classmethod object
- **body** (*dict*) – Content of the card
- **eureqa** (Eureqa) – Eureqa-API instance

Returns AnalysisCard

move_above(*other_card*)
Moves this card above another card.

Parameters **other_card** (eureqa.analysis_cards.AnalysisCard) – The other card above which to move this card.

move_below(*other_card*)
Moves this card below another card.

Parameters **other_card** (eureqa.analysis_cards.AnalysisCard) – The other card object below which to move this card.

replace(*other*)
Replace this card's contents with the contents of *other*

Parameters **other** (*eureqa.analysis_card.AnalysisCard*) – Card to replace our contents with

to_json()
Convert this AnalysisCard to a JSON-serializable structure

Returns A representation of this card's contents as primitive Python objects

3.2.2 binned_mean_plot_card

class eureqa.analysis_cards.binned_mean_plot_card.**BinnedMeanPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a binned mean plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.

- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

3.2.3 box_plot_card

class eureqa.analysis_cards.box_plot_card.**BoxPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a box plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

3.2.4 by_row_plot_card

class eureqa.analysis_cards.by_row_plot_card.**ByRowPlotCard**(*datasource=None*, *x_var=None*, *plotted_vars=None*, *title=None*, *description=None*, *focus_variable=None*, *should_center=True*, *should_scale=False*, *collapse=False*)

Represents a variable line graph on the server.

Parameters

- **datasource** (*eureqa.DataSource*) – Data source for the card's data
- **x_var** (*str*) – Name of the variable to plot as the X axis
- **plotted_vars** (*str*) – List of string-names of variables to plot. (To modify a variable's display name, first create the card; then modify the display name directly on it.)
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **focus_variable** (*str*) – Name of the variable in 'plotted_vars' to bring to the foreground
- **should_center** (*bool*) – Should the plot be centered?
- **should_scale** (*bool*) – Should the plot scale?
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***focus_variable*** (*str*) – Focused (foreground) variable for the card. Must be a member of ~eureqa.analysis_cards.ByRowPlotCard.plotted_variables
- ***x_var*** (*str*) – Name of the variable to plot as the X axis
- ***plotted_variables*** (*list*) – Variables to plot. (List of string variable names.)
- ***should_center*** (*bool*) – Should the plot be centered?
- ***should_scale*** (*bool*) – Should the plot scale?

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

datasource

The data source providing data for this card

Returns data source providing data for this card

description

The description of this card.

Returns description of this card

Return type str

focus_variable
The variable that is currently in focus (in the foreground) for this card. Must be a member of 'plotted_variables'.

Returns focus_variable for this card

Return type str

plotted_variables
The plotted variables for this card.

Returns List of the names of the variables being plotted against the X axis

Return type tuple

should_center
The should_center option for this card.

Returns whether this plot should be centered

Return type bool

should_scale
The should_scale option for this card.

Returns whether this plot should be scaled

Return type bool

title
The title of this card.

Returns title of this card

Return type str

x_var
The X-axis variable for this card

Returns the name of the X-axis variable for this card

Return type str

3.2.5 custom_plot_card

class eureqa.analysis_cards.custom_plot_card.**CustomPlotCard**(*plot=None*, *title=None*, *description=None*, *collapse=False*)

Represents a card which can display a custom plot.

Parameters

- **plot** (eureqa.analysis_cards.Plot) – The Plot to be displayed in the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***plot*** (eureqa.analysis_cards.Plot) – the underlying Plot displayed by this card.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

delete()
Deletes the card from the server.

This will delete any data which has been uploaded internally to support plotting.

description
The description of this card.

Returns description of this card

Return type str

plot
The underlying Plot displayed by this card.

title
The title of this card.

Returns title of this card

Return type str

3.2.6 distribution_plot_card

class eureqa.analysis_cards.distribution_plot_card.**DistributionPlotCard**(*datasource=None*, *variable=None*, *title=None*, *description=None*, *collapse=False*)

Represents a distribution plot card on the server.

Parameters

- **datasource** (*eureqa.DataSource*) – The data source to which the variable belongs.
- **variable** (*str*) – The name of the variable that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card title.
- **datasource** (*str*) – The datasource used by the card.
- **variable** (*str*) – The variable plotted by the coard.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

3.2.7 double_histogram_plot_card

class eureqa.analysis_cards.double_histogram_plot_card.**DoubleHistogramPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a double-histogram plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

3.2.8 html_card

class eureqa.analysis_cards.html_card.**HtmlCard**(*html=''*, *title='HTML'*, *description=None*, *collapse=False*)

** BETA ** Represents a text card on the server.

Parameters

- **title** (*str*) – Title of the card
- **description** (*str*) – Description of the card
- **html** (*str*) – Body of the card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***text*** (*str*) – The card text.

collapse
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description
The description of this card.

Returns description of this card

Return type str

html
The body of this card.

Returns body of this card

Return type str

text
The text contents of this card.

Returns text contained in this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

3.2.9 model_card

class eureqa.analysis_cards.model_card.**ModelCard**(*solution=None*, *title=None*, *description=None*, *collapse=False*)

Represents a model card on the server.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **title** (*str*) – The card title.
- **description** (*str*) – The card description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card title.
- ***description*** (*str*) – The card description.

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description

The description of this card.

Returns description of this card

Return type str

solution

Returns Solution that the card is rendering

Return type *Solution*

title

The title of this card.

Returns title of this card

Return type str

3.2.10 model_evaluator_card

class eureqa.analysis_cards.model_evaluator_card.**ModelEvaluatorCard**(*title=None*, *description=None*, *collapse=False*, *solution_infos=None*)

Represents an evaluate model card on the server.

Parameters

- **title** (*str*) – Title of the card. Defaults to 'Evaluate Model'.
- **description** (*str*) – Description of the card.

- **collapse** (*bool*) – Whether the card should default to be collapsed.
- **solution_infos** (*list(eureqa.analysis_cards.ModelEvaluatorCard.SolutionInfo)*) – List of solutions that this card is rendering

Variables

- ***title*** (*str*) – The title of the card
- ***solution_infos*** (*tuple*) – Immutable ordered set of SolutionInfo objects representing the solutions shown on this card.

class SolutionInfo(*datasource=None*, *solution=None*)

The solution information for a single model evaluation ("tab") on ModelEvaluatorCard

Parameters

- **datasource** (eureqa.data_source.DataSource) – Data source for this solution
- **solution** (eureqa.solution.Solution) – Solution

Variables

- ***dataset_id*** (*str*) – ID of the DataSet referenced by this solution-tab
- ***datasource_id*** (*str*) – ID of the DataSource referenced by this solution-tab
- ***search_id*** (*str*) – ID of the Search referenced by this solution-tab
- ***solution_id*** (*str*) – ID of the Solution referenced by this solution-tab
- ***has_target_variable*** (*bool*) – Whether the dataset contains the target variable

dataset_id

ID of the DataSet used for this solution

Returns unique DataSet identifier

datasource_id

ID of the DataSource used for this solution

Returns unique DataSource identifier

has_target_variable

Whether the dataset contains the target variable

Returns Whether the dataset contains the target variable

search_id

ID of the Search used for this solution

Returns unique Search identifier

solution_id

ID of the Solution used for this solution

Returns unique Solution identifier

ModelEvaluatorCard.**add_solution_info**(*datasource*, *solution*)

Add a new (non-default) solution and tab to this card. Once added, this object will show up in the list returned by *solution_infos*.

Parameters

- **datasource** (DataSource) – DataSource used with this solution.
- **solution** (Solution) – Solution associated with the model evaluation.

ModelEvaluatorCard.**collapse**

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

ModelEvaluatorCard.**description**
The description of this card.

Returns description of this card

Return type str

ModelEvaluatorCard.**solution_infos**
The set of all SolutionInfo objects associated with this card. One per solution tab displayed in the UI. Note that *solution_infos[0]* is the default card; it may be treated specially by the UI.

Returns List or tuple of SolutionInfo objects

ModelEvaluatorCard.**title**
The title of this card.

Returns title of this card

Return type str

3.2.11 model_fit_by_row_plot_card

class eureqa.analysis_cards.model_fit_by_row_plot_card.**ModelFitByRowPlotCard**(*solution=None*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Represents a projection card on the server.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***x_axis*** (*str*) – The card's X axis

3.2.12 model_fit_plot_card

class eureqa.analysis_cards.model_fit_plot_card.**ModelFitPlotCard**(*solution=None*, *x_axis=None*, *title=None*, *description=None*, *collapse=False*)

Represents a projection card on the server. Common base class for all projection cards.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***x_axis*** (*str*) – The card's X axis

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description

The description of this card.

Returns description of this card

Return type str

options

The configuration options for this card.

Returns Object representing configuration options

solution

Returns Solution that the card is rendering

Return type *Solution*

title

The title of this card.

Returns title of this card

Return type str

x_axis

The x_axis option for this card.

Returns x axis for this card

Return type str

3.2.13 model_fit_separation_plot_card

class eureqa.analysis_cards.model_fit_separation_plot_card.**ModelFitSeparationPlotCard**(*solution=*, *x_axis=N*, *title=None*, *description=Non*, *collapse=Fa*)

Represents a centipede plot card on the server.

Parameters

- **solution** (*Solution*) – The Solution object for which this card is being created
- **x_axis** (*str*) – The card's X axis label
- **title** (*str*) – The card's title.
- **description** (*str*) – The card's description.
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The card's title.
- ***x_axis*** (*str*) – The card's X axis

3.2.14 model_summary_card

class eureqa.analysis_cards.model_summary_card.**ModelSummaryCard**(*solution=None*, *title=''*, *description=None*, *collapse=False*)

Represents a model summary card on the server.

Parameters

- **solution** (*eureqa.Solution*) – The solution that will be displayed on the card.
- **title** (*str*) – Title of the card
- **description** (*str*) – Title of the card
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables ***title*** (*str*) – The card title.

collapse

Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

description

The description of this card.

Returns description of this card

Return type str

solution

Returns Solution that the card is rendering

Return type *Solution*

title

The title of this card.

Returns title of this card

Return type str

3.2.15 plot

class eureqa.analysis_cards.plot.**Plot**(*width=None*, *height='400px'*, *x_axis_label=None*, *y_axis_label=None*, *show_legend=True*, *zero_x_line=False*, *zero_y_line=False*, *x_tick_format=None*, *y_tick_format=None*, *num_x_ticks=None*, *num_y_ticks=None*, *max_displayed_points=200*, *guides_type='XY'*)

Beta

Represents a plot to be displayed in an analysis plot card.

By calling the plot() method multiple times on one instance of this class, multiple lines and scatter plots can be superimposed on this object.

Parameters

- **width** (*str*) – Set manual dimensions for the plot in "px" units (e.g. 350px). Defaults to full panel width.
- **height** (*str*) – Set manual dimensions for the plot in "px" units. Defaults to a constant height of 400px.
- **x_axis_label** (*str*) – The label to use on the x axis of the plot.
- **y_axis_label** (*str*) – The label to use on the y axis of the plot.
- **show_legend** (*bool*) – Controls whether or not a legend will be used in the plot.
- **zero_x_line** (*bool*) – Draws a horizontal line through the origin
- **zero_y_line** (*bool*) – Draws a vertical line through the origin
- **x_tick_format** (*str*) – Format for x axis tick labels. Valid values are "date" or anything supported by D3: https://github.com/mbostock/d3/wiki/Formatting#d3_format. Defaults to our internal number format
- **y_tick_format** (*str*) – Format for y axis tick labels. Valid values are "date" or anything supported by D3: https://github.com/mbostock/d3/wiki/Formatting#d3_format. Defaults to our internal number format
- **num_x_ticks** (*int*) – Number of x axis tick labels. Defaults to as many as fit comfortably without overlap
- **num_y_ticks** (*int*) – Number of y axis tick labels. Defaults to as many as fit comfortably without overlap
- **max_displayed_points** (*int*) – Maximum number of data points to sample for each component

- **guides_type** (*str*) – The type of value-guides to show when hovering over a point in the graph. Valid values are "XY", "YY" or False. XY guides will show the x and y values for the point under the cursor. YY guides will show the x and y values of each component for the data point closest to the cursor. False will turn off value guides.

delete()
Delete the Plot and any associated data which has been uploaded.

plot(*x*, *y*, *datasource=None*, *style='-'*, *color='black'*, *line_width=1*, *circle_radius=1*, *use_in_plot_range=True*, *error_bars_upper_values=None*, *error_bars_lower_values=None*, *legend_label=None*, *tooltip_template=None*)
Add new data to the plot with the specified options.

If a datasource is specified, x and y can be provided as expressions in terms of the variables in that datasource. Otherwise, x and y must be lists of data of identical length.

Parameters

- **x** (*list*) – X axis data. Either a list of input values, or if a datasource is specified, an expression in terms of that datasource's variables.
- **y** (*list*) – Y axis data. Either a list of input values, or if a datasource is specified, an expression in terms of that datasource's variables.
- **datasource** (DataSource) – If provided, a specific DataSource to use as a base for the variables to be plotted.
- **style** (*str*) – A matplotlib-like style string of 'o' or '-' or '-o' to indicate circle, line, or line-circle, respectively.
- **color** (*str*) – CSS color.
- **line_width** (*int*) – The width of the line, if applicable.
- **circle_radius** (*int*) – The radius of each circle, if applicable.
- **use_in_plot_range** (*bool*) – The chart auto-computes the x and y axis ranges based on the data for each component. If you want to add a component, but have its data not be used to compute the x and y axis ranges, set this value to False. For example, if you want to make a scatter plot, with a trend line going through it, then setting this field to False for the trend-line component will make the chart snugly fit the points, with the trend line extending beyond.
- **error_bars_upper_values** (*list*) – The column to use for the tops of error bars
- **error_bars_lower_values** (*list*) – The column to use for the bottoms of error bars
- **legend_label** (*str*) – A string label to be used in the legend. If unspecified, this plot will not appear in the legend.
- **tooltip_template** (*str*) – Any valid Handlebars template. The template will be provided the x and y values as "x_value" and "y_value", so you can use these to format a tooltip for each point. For example, " When age is {{x_value}}, salary is expected to be {{y_value}} "

upload_data(*eureqa*)
Upload the plot data to eureqa. This is required before the plot can be viewed.

Parameters **eureqa** (Eureqa) – a eureqa connection

3.2.16 scatter_plot_card

class eureqa.analysis_cards.scatter_plot_card.**ScatterPlotCard**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a scatter plot card on the server.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- ***title*** (*str*) – The title of the card
- ***x_var*** (*str*) – The X-axis variable for the card's plot
- ***y_var*** (*str*) – The Y-axis variable for the card's plot
- ***needs_guides*** (*bool*) – Whether the card needs guides
- ***axis_labels*** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- ***label_format*** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

3.2.17 text_card

class eureqa.analysis_cards.text_card.**TextCard**(*text=''*, *title='Text'*, *description=None*, *collapse=False*)

Represents a text card on the server.

Parameters

- **title** (*str*) – Title of the card
- **description** (*str*) – Description of the card
- **text** (*str*) – Body of the card. (Markdown-formatted text.)

- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables **text** (*str*) – The card text.

delete()
Deletes the card from the server.

If the card contains any images, they will be deleted.

description
The description of this card.

Returns description of this card

Return type str

text
The text contents of this card.

Returns text contained in this card

Return type str

title
The title of this card.

Returns title of this card

Return type str

3.2.18 two_variable_plot

class eureqa.analysis_cards.two_variable_plot.**TwoVariablePlot**(*datasource=None*, *x_var=None*, *y_var=None*, *title=None*, *description=None*, *needs_guides=False*, *axis_labels=None*, *label_format=None*, *collapse=False*)

Represents a two-variable plot. Common base class for two-variable plots.

Parameters

- **datasource** (DataSource) – The data source containing the data to be plotted.
- **x_var** (*str*) – The X-axis variable for the card's plot.
- **y_var** (*str*) – The Y-axis variable for the card's plot.
- **title** (*str*) – The title of the card.
- **description** (*str*) – A textual description of the card.
- **needs_guides** (*bool*) – Whether the card needs guides.
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".
- **collapse** (*bool*) – Whether the card should default to be collapsed.

Variables

- **title** (*str*) – The title of the card
- **x_var** (*str*) – The X-axis variable for the card's plot
- **y_var** (*str*) – The Y-axis variable for the card's plot
- **needs_guides** (*bool*) – Whether the card needs guides
- **axis_labels** (XYMap) – Axis labels for this card's plot. Set member fields "x" and "y" to set the X and Y axis labels.
- **label_format** (XYMap) – Label format for this card. Set member fields "x" and "y" to set the X and Y axis printf-style format-strings; for example, ".3s".

class XYMap (*tvp*, *dic*)
A named tuple with keys 'x' and 'y'

Parameters

- **tvp** (*TwoVariablePlotCard*) – The TwoVariablePlotCard object.
- **dic** (*dict*) – X and Y parameters as dictionary.

x
X value

y
Y value

TwoVariablePlot.**axis_labels**
The axis labels for this card

Returns Axis labels for this card

Return type *TwoVariablePlot.XYMap*

TwoVariablePlot.**collapse**
Whether the card is collapsed by default.

Returns whether the card is collapsed by default

Return type str

TwoVariablePlot.**datasource**
The data source providing data for this card

Returns data source providing data for this card

TwoVariablePlot.**description**
The description of this card.

Returns description of this card

Return type str

TwoVariablePlot.**label_format**
The label format for this card

Returns A dictionary mapping axis names to printf-style format-string specifiers (for example, ".3s")

Return type dict

TwoVariablePlot.**needs_guides**
Does this card need guides?

Returns Whether this card needs guides

Return type bool

`TwoVariablePlot.`**`title`**
The title of this card.

Returns title of this card

Return type str

`TwoVariablePlot.`**`x_var`**
The X variable for this card.

Returns X variable for this card

Return type str

`TwoVariablePlot.`**`y_var`**
The Y variable for this card.

Returns Y variable for this card

Return type str

3.3 analysis_templates

class `eureqa.analysis_templates.`**`AnalysisTemplate`**(*body*, *eureqa*)
Represents an analysis template on the server.

Parameters

- **`body`** (*dict*) – Class metadata as dictionary
- **`eureqa`** (Eureqa) – A eureqa connection

Variables

- **`name`** (*str*) – The name of the analysis template.
- **`description`** (*str*) – The description of the analysis template.
- **`icon_url`** (*str*) – The url of the icon for the analysis template.
- **`parameters`** (*list*) – The list of parameters for the analysis template.

`delete`()
Delete the analysis template.

`description`
Analysis Template's extended description

`execute`(*values*)
Start execution of an analysis template with given parameters.

Parameters **`values`** (*dict*) – Analysis template settings as dictionary

`get_execution`(*execution_id*)
Get a specific execution of the analysis template.

Parameters **`execution_id`** (*str*) – Execution identifier for this execution instance

`get_executions`()
Get all executions of the analysis template.

get_module(*output_filename*)
Download the Python code which implementes this analysis template, as a .zip package, to the specified file on the local filesystem. Returns the main_module's name

Parameters **output_filename** (*str*) – The filename to be used for the resulting zip file containing the analysis template code.

name
Analysis Template's name

parameters
Parameters object representing this template

set_module(*main_module_name*, *module_fs_path*, *ignore_files=None*, *additional_modules_paths=[]*)
Set the the python module and function that specifies the execution of this analysis template.

Parameters

- **main_module_name** (*str*) – Absolute Python-import name of the main module to run. For example, "example_module".
- **module_fs_path** (*str*) – Filesystem path where the module containing the function lives. For example, "C:Userseureqaexample_module". If omitted, this is inferred by importing the function above and taking the parent directory of the file that contains it.
- **ignore_files** (*list[str]*) – List of names of files that, if encountered, will not be uploaded to the Eureqa server.
- **additional_modules_paths** (*list*) – List of other directory names, that, if encountered, will be uploaded to the Eureqa server as well

class eureqa.analysis_templates.**ComboBoxParameter**(*id*, *label*, *items*)
Combo box parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **items** (*list[str]*) – The items to populate the combo box with.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **items** (*list[str]*) – The items to populate the combo box with.

class eureqa.analysis_templates.**ComboBoxParameterValue**(*eureqa*, *id*, *value*)
Combo box parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **eureqa** (Eureqa) – A eureqa connection
- **value** (*str*) – The parameter value.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

class eureqa.analysis_templates.**DataFileParameter**(*id*, *label*, *description*, *filetypes*)
Data file parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **description** (*str*) – A description that will be shown in UI.
- **filetypes** (*list[str]*) – The list of accepted filetypes.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- ***description*** (*str*) – A description that will be shown in UI.
- **filetypes** (*list[str]*) – The list of accepted filetypes.

class eureqa.analysis_templates.**DataFileParameterValue**(*eureqa*, *id*, *value*, *object_store_bucket*, *object_store_key*)
Combo box parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **object_store_bucket** (*int*) – The specific collection in the object store.
- **object_store_key** (*str*) – The reference into the object store bucket for this id.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

class eureqa.analysis_templates.**DataSourceParameter**(*id*, *label*, *variables*)
Data source parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **variables** (*list[VariableParameter]*) – The parameters for variables that belong to the data set described by this parameter.

class eureqa.analysis_templates.**DataSourceParameterValue**(*eureqa*, *id*, *value*, *variables*)
Data source parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **variables** (*list[VariableParameterValue]*) – The parameters values for variables that belong to this data source.

class eureqa.analysis_templates.**Execution**(*body*, *template_id*, *eureqa*)

Represents an analysis template execution on the server.

Parameters

- **body** (*dict*) – Class metadata as dictionary
- **template_id** (*str*) – The id of the analysis_template the execution belongs to.
- **eureqa** (Eureqa) – A eureqa connection.

Variables

- **template_id** (*str*) – The id of the analysis_template the execution belongs to.
- **analysis_id** (*str*) – The id of the analysis the execution belongs to.
- **state** (*str*) – The current state of the execution.
- **parameters** (*list*) – The list of parameter values for the execution.
- ***progress_updates*** (*list*) – The list of updates for the execution.

get_analysis()

Retrieves the analysis that belongs to the execution.

get_analysis_template()

Retrieves the analysis that belongs to the execution.

progress_updates

Get all progress updates for an execution of an analysis template.

report_fatal_error(*error*)

Notifies the server that an error occurred during the execution and terminates the script execution.

Parameters **error** (*str*) – The error that occurred during the execution.

report_validation_result(*type*, *message=None*, *details=None*, *parameter_id=None*)

Report an info/warning/error message about the specified parameter to be shown to the user in validation review

Parameters

- **type** (*str*) – If this result is INFO, WARNING, or ERROR
- **message** (*str*) – The result message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

update_progress(*message*)

Create a progress update for an execution of an analysis template.

Parameters **message** (*str*) – The progress message

validation_results

Get all validation results for the execution of an analysis template.

class eureqa.analysis_templates.**Parameters**(*parameters=None*)
Analysis template parameters definition

Parameters **parameters** (*list[Parameter]*) – The list of parameters for the template, whether text, variable or datasource.

class eureqa.analysis_templates.**ParametersValues**(*parameters=None*)
Analysis template parameters values

Parameters **parameters** (*list[Parameter]*) – The list of parameter values for the template, whether text, variable or datasource value.

class eureqa.analysis_templates.**ProgressUpdate**(*body*)
Represents an analysis template execution progress update on the server.

Parameters **body** (*dict*) – Class metadata as dictionary

Variables

- **message** (*str*) – The the message about the execution's status.
- **time_stamp** (*datetime*) – The time the progress update was created.

class eureqa.analysis_templates.**TextParameter**(*id*, *label*, *text_multiline=False*)
Text parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **text_multiline** (*bool*) – Indicates that the text should be split across multiple lines.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

class eureqa.analysis_templates.**TextParameterValue**(*eureqa*, *id*, *value*, *text_multiline=False*)
Text parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **text_multiline** (*bool*) – Whether to display as multiline text

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

class eureqa.analysis_templates.**TopLevelModelParameter**(*id*, *label*, *custom_disabled*)
TopLevelModel parameter description for analysis template. For selecting a single model (either an existing one within a dataset and search, or a custom one).

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **custom_disabled** (*bool*) – Whether to block the user from setting a custom expression

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **custom_disabled** (*bool*) – Whether to block the user from setting a custom expression

class eureqa.analysis_templates.**TopLevelModelParameterValue**(*eureqa*, *id*, *value*, *datasource_id*, *search_id*, *solution_id*)

TopLevelModel parameter value for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **datasource_id** (*str*) – The parameter value for the datasource the expression belongs to.
- **search_id** (*str*) – The parameter value for the search the expression belongs to.
- **solution_id** (*str*) – The parameter value for the solution the expression belongs to.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

class eureqa.analysis_templates.**VariableParameter**(*id*, *label*)

Variable parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

class eureqa.analysis_templates.**VariableParameterValue**(*eureqa*, *id*, *value*)

Variable parameter value for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

class eureqa.analysis_templates.**ParameterValidationResult**(*type*, *message=None*, *details=None*, *parameter_id=None*)

Represents an analysis template execution validation result to the server

Parameters

- **type** (*str*) – If this result is INFO, WARNING, or ERROR
- **message** (*str*) – The the message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

Variables

- **type** (*str*) – If this result is INFO, WARNING, or ERROR
- **message** (*str*) – The the message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

3.3.1 analysis_template

class eureqa.analysis_templates.analysis_template.**AnalysisTemplate**(*body*, *eureqa*)

Represents an analysis template on the server.

Parameters

- **body** (*dict*) – Class metadata as dictionary
- **eureqa** (Eureqa) – A eureqa connection

Variables

- **name** (*str*) – The name of the analysis template.
- **description** (*str*) – The description of the analysis template.
- **icon_url** (*str*) – The url of the icon for the analysis template.
- **parameters** (*list*) – The list of parameters for the analysis template.

delete()

Delete the analysis template.

description

Analysis Template's extended description

execute(*values*)

Start execution of an analysis template with given parameters.

Parameters **values** (*dict*) – Analysis template settings as dictionary

get_execution(*execution_id*)
Get a specific execution of the analysis template.

Parameters **execution_id** (*str*) – Execution identifier for this execution instance

get_executions()
Get all executions of the analysis template.

get_module(*output_filename*)
Download the Python code which implementes this analysis template, as a .zip package, to the specified file on the local filesystem. Returns the main_module's name

Parameters **output_filename** (*str*) – The filename to be used for the resulting zip file containing the analysis template code.

name
Analysis Template's name

parameters
Parameters object representing this template

set_module(*main_module_name*, *module_fs_path*, *ignore_files=None*, *additional_modules_paths=[]*)
Set the the python module and function that specifies the execution of this analysis template.

Parameters

- **main_module_name** (*str*) – Absolute Python-import name of the main module to run. For example, "example_module".
- **module_fs_path** (*str*) – Filesystem path where the module containing the function lives. For example, "C:Userseureqaexample_module". If omitted, this is inferred by importing the function above and taking the parent directory of the file that contains it.
- **ignore_files** (*list[str]*) – List of names of files that, if encountered, will not be uploaded to the Eureqa server.
- **additional_modules_paths** (*list*) – List of other directory names, that, if encountered, will be uploaded to the Eureqa server as well

3.3.2 combo_box_parameter

class eureqa.analysis_templates.combo_box_parameter.**ComboBoxParameter**(*id*, *label*, *items*)
Combo box parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **items** (*list[str]*) – The items to populate the combo box with.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **items** (*list[str]*) – The items to populate the combo box with.

3.3.3 combo_box_parameter_value

class eureqa.analysis_templates.combo_box_parameter_value.**ComboBoxParameterValue**(*eureqa*, *id*, *value*)

Combo box parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **eureqa** (Eureqa) – A eureqa connection
- **value** (*str*) – The parameter value.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

3.3.4 data_file_parameter

class eureqa.analysis_templates.data_file_parameter.**DataFileParameter**(*id*, *label*, *description*, *filetypes*)

Data file parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **description** (*str*) – A description that will be shown in UI.
- **filetypes** (*list[str]*) – The list of accepted filetypes.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- ***description*** (*str*) – A description that will be shown in UI.
- **filetypes** (*list[str]*) – The list of accepted filetypes.

3.3.5 data_file_parameter_value

class eureqa.analysis_templates.data_file_parameter_value.**DataFileParameterValue**(*eureqa*, *id*, *value*, *object_store_bucket*, *object_store_key*)

Combo box parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **object_store_bucket** (*int*) – The specific collection in the object store.
- **object_store_key** (*str*) – The reference into the object store bucket for this id.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

3.3.6 data_source_parameter

class eureqa.analysis_templates.data_source_parameter.**DataSourceParameter**(*id*, *label*, *variables*)

Data source parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **variables** (*list[VariableParameter]*) – The parameters for variables that belong to the data set described by this parameter.

3.3.7 data_source_parameter_value

class eureqa.analysis_templates.data_source_parameter_value.**DataSourceParameterValue**(*eureqa*, *id*, *value*, *variables*)

Data source parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **variables** (*list[VariableParameterValue]*) – The parameters values for variables that belong to this data source.

3.3.8 execution

class eureqa.analysis_templates.execution.**Execution**(*body*, *template_id*, *eureqa*)

Represents an analysis template execution on the server.

Parameters

- **body** (*dict*) – Class metadata as dictionary
- **template_id** (*str*) – The id of the analysis_template the execution belongs to.
- **eureqa** (Eureqa) – A eureqa connection.

Variables

- **template_id** (*str*) – The id of the analysis_template the execution belongs to.
- **analysis_id** (*str*) – The id of the analysis the execution belongs to.
- **state** (*str*) – The current state of the execution.
- **parameters** (*list*) – The list of parameter values for the execution.
- ***progress_updates*** (*list*) – The list of updates for the execution.

get_analysis()

Retrieves the analysis that belongs to the execution.

get_analysis_template()

Retrieves the analysis that belongs to the execution.

progress_updates

Get all progress updates for an execution of an analysis template.

report_fatal_error(*error*)

Notifies the server that an error occurred during the execution and terminates the script execution.

Parameters **error** (*str*) – The error that occurred during the execution.

report_validation_result(*type*, *message=None*, *details=None*, *parameter_id=None*)

Report an info/warning/error message about the specified parameter to be shown to the user in validation review

Parameters

- **type** (*str*) – If this result is INFO, WARNING, or ERROR
- **message** (*str*) – The result message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

update_progress(*message*)

Create a progress update for an execution of an analysis template.

Parameters **message** (*str*) – The progress message

validation_results
Get all validation results for the execution of an analysis template.

3.3.9 parameter

class eureqa.analysis_templates.parameter.**Parameter**(*id*, *label*, *_type*)
Base class for all analysis templates parameters

Parameters

- **id** (*str*) – The unique identifier for this Parameter.
- **label** (*str*) – The human-readable label to be associated with this Parameter.

3.3.10 parameters

class eureqa.analysis_templates.parameters.**Parameters**(*parameters=None*)
Analysis template parameters definition

Parameters **parameters** (*list[Parameter]*) – The list of parameters for the template, whether text, variable or datasource.

3.3.11 parameters_values

class eureqa.analysis_templates.parameters_values.**ParametersValues**(*parameters=None*)
Analysis template parameters values

Parameters **parameters** (*list[Parameter]*) – The list of parameter values for the template, whether text, variable or datasource value.

3.3.12 parameter_validation_result

class eureqa.analysis_templates.parameter_validation_result.**ParameterValidationResult**(*type*, *message=None*, *details=None*, *parameter_id=None*)
Represents an analysis template execution validation result to the server

Parameters

- **type** (*str*) – If this result is INFO, WARNING, or ERROR
- **message** (*str*) – The the message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

Variables

- **type** (*str*) – If this result is INFO, WARNING, or ERROR

- **message** (*str*) – The the message
- **details** (*str*) – (optional) Detailed message about the result
- **parameter_id** (*str*) – (optional) the analysis template parameter that this progress update refers to

3.3.13 parameter_value

class eureqa.analysis_templates.parameter_value.**ParameterValue**(*id*, *value*, *_type*)
Base class for all analysis templates parameters values

Parameters

- **id** (*str*) – The unique identifier for this parameter.
- **value** (*str*) – The value of this parameter.

3.3.14 progress_update

class eureqa.analysis_templates.progress_update.**ProgressUpdate**(*body*)
Represents an analysis template execution progress update on the server.

Parameters **body** (*dict*) – Class metadata as dictionary

Variables

- **message** (*str*) – The the message about the execution's status.
- **time_stamp** (*datetime*) – The time the progress update was created.

3.3.15 runner

class eureqa.analysis_templates.runner.**analysis_template_runner**
Bootstrapper for analysis templates. Sets up the environment required to invoke the analysis template and then invokes it.

get_analysis_module_from_template(*eureqa*, *template*)
Retrieves the bytes that represent an analysis template from the eureqa server (as base64-encoded .zip file module); then function, raising an Exception if an error occurs.

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **template** (*Template*) – The Template object containing the desired analysis.

run_args(*arguments*)
Run a template with the provided arguments.

Parameters **arguments** (*str*) – Command line-style argument string.

analysis_template_runner

exception eureqa.analysis_templates.runner.analysis_template_runner.**LocalFatalException**
Class that represents fatal exceptions from the local runner

class eureqa.analysis_templates.runner.analysis_template_runner.**Local_analysis_template_execu**

Stubs out the analysis_template_execution to provide the same interface while running locally.

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **parameters** (*str*) – JSON string of parameters.

report_fatal_error(*error*)
Report a fatal error of the running analysis.

Parameters **error** (*str*) – Message to print indicating progress.

throw_if_fatal_exception()
Throw any error encountered as a fatal exception

update_progress(*message*)
Update progress of the running analysis.

Parameters **message** (*str*) – Message to print indicating progress.

class eureqa.analysis_templates.runner.analysis_template_runner.**analysis_template_runner**
Bootstrapper for analysis templates. Sets up the environment required to invoke the analysis template and then invokes it.

get_analysis_module_from_template(*eureqa*, *template*)
Retrieves the bytes that represent an analysis template from the eureqa server (as base64-encoded .zip file module); then function, raising an Exception if an error occurs.

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **template** (*Template*) – The Template object containing the desired analysis.

run_args(*arguments*)
Run a template with the provided arguments.

Parameters **arguments** (*str*) – Command line-style argument string.

client

eureqa.analysis_templates.runner.client.**main**(*argv*)
Handle executing analysis template code without actually uploading it to the server. Mostly a wrapper around ‘eureqa.analysis_templates.runner.analysis_template_runner. The latter is called directly by the server; this one is a little more user-friendly and respects eureqa_config.json.

Parameters **argv** (*list*) – Arguments

3.3.16 text_parameter

class eureqa.analysis_templates.text_parameter.**TextParameter**(*id*, *label*, *text_multiline=False*)
Text parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **text_multiline** (*bool*) – Indicates that the text should be split across multiple lines.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

3.3.17 text_parameter_value

class eureqa.analysis_templates.text_parameter_value.**TextParameterValue**(*eureqa*, *id*, *value*, *text_multiline=False*)

Text parameter description for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **text_multiline** (*bool*) – Whether to display as multiline text

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

3.3.18 top_level_model_parameter

class eureqa.analysis_templates.top_level_model_parameter.**TopLevelModelParameter**(*id*, *label*, *custom_disabled*)

TopLevelModel parameter description for analysis template. For selecting a single model (either an existing one within a dataset and search, or a custom one).

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.
- **custom_disabled** (*bool*) – Whether to block the user from setting a custom expression

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

- **custom_disabled** (*bool*) – Whether to block the user from setting a custom expression

3.3.19 top_level_model_parameter_value

class eureqa.analysis_templates.top_level_model_parameter_value.**TopLevelModelParameterValue**(*e*
i
v
a
s
s
s
l
t

TopLevelModel parameter value for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.
- **datasource_id** (*str*) – The parameter value for the datasource the expression belongs to.
- **search_id** (*str*) – The parameter value for the search the expression belongs to.
- **solution_id** (*str*) – The parameter value for the solution the expression belongs to.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

3.3.20 variable_parameter

class eureqa.analysis_templates.variable_parameter.**VariableParameter**(*id*, *label*)

Variable parameter description for analysis template

Parameters

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

Variables

- **id** (*str*) – The id of the parameter that will be passed together with its value to an analysis script.
- **label** (*str*) – The parameter label that will be shown in UI.

3.3.21 variable_parameter_value

class eureqa.analysis_templates.variable_parameter_value.**VariableParameterValue**(*eureqa*, *id*, *value*)

Variable parameter value for analysis template

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

Variables

- **id** (*str*) – The id of the parameter.
- **value** (*str*) – The parameter value.

3.4 api_version

3.5 complexity_weights

class eureqa.complexity_weights.**ComplexityWeights**(*continuous_variable=2*, *boolean_variable=0*, *nest_depth=0*)

Represents complexity penalties applied to each model to reward for increased interpretability.

Parameters

- **continuous_variable** (*int*) – additional complexity added for each continuous variable in a term after the first continuous variable.
- **boolean_variable** (*int*) – additional complexity added for each binary variable in a model.
- **nest_depth** (*int*) – additional complexity added for each term with nested functions, like log(sin(x)).

3.6 data_source

class eureqa.data_source.**DataSource**(*eureqa*, *body*)

Represents an interface to a data source on the server

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **body** (*dict*) – Class metadata as dictionary.

Variables

- **name** (*str*) – The data source name.
- **number_columns** (*int*) – The number of columns in the data source.
- **number_rows** (*int*) – The number of rows (variables) in the data source.

create_search(*search_settings*)
Creates a new search with input as a SearchSettings object.

Parameters **search_settings** (*SearchSettings*) – the settings for creating a new search.

Returns A Search object which represents a newly create search on the server.

Return type Search

create_variable(*expression*, *variable_name*)
Adds a new variable to the data_source with values from evaluating the given expression.

Parameters

- **expression** (*str*) – the expression to evaluate to fill in the values
- **variable_name** (*str*) – what to name the new variable

delete()
Deletes the data source from the server.

Raises **Exception** – If the data source is already deleted.

download_data_file(*file_path*)
Downloads the originally uploaded data file from the server.

Parameters **file_path** (*str*) – the filepath at which to save the data

get_searches()
Retrieves from the server a list of searches associated with the data source.

Returns The list of all searches associated with the data source.

Return type list of Search

get_variable_details(*variable_name*)
Retrieves the details for the requested variable from the data_source.

Parameters **variable_name** (*str*) – the name of the variable to get the details for

Returns The object representing the variable details

Return type *VariableDetails*

get_variables()
Retrieves from the server a list of variables in a data set.

Returns A list of the same variables as visible in Eureqa UI. Including all derived variables.

Return type list of str

3.7 data_splitting

class eureqa.data_splitting.**DataSplitting**(*shuffle*, *training_data_percentage=None*, *validation_data_percentage=None*, *training_selection_expression=None*, *validation_selection_expression=None*, *training_selection_expression_type=None*, *validation_selection_expression_type=None*)
Represents a data splitting settings for the genetic algorithm.

Parameters

- **shuffle** (*bool*) – Indicates whether data are shuffled or not before the training.
- **training_data_percentage** (*float*) – The percentage of data used for training.
- **validation_data_percentage** (*float*) – The percentage of data used for validation.
- **training_selection_expression** (*str*) – The expression used for selecting the training set.
- **validation_selection_expression** (*str*) – The expression used for selecting the validation set.
- **training_selection_expression_type** (*str*) – The type of expression used for selecting the training set. ('EXPRESSION','VARIABLE',None)
- **validation_selection_expression_type** (*str*) – The type of expression used for selecting the validation set. ('EXPRESSION','VARIABLE',None)

Variables

- **shuffle** (*bool*) – Indicates whether data are shuffled or not before the training.
- **training_data_percentage** (*float*) – The percentage of data used for training.
- **validation_data_percentage** (*float*) – The percentage of data used for validation.
- **training_selection_expression** (*str*) – The expression used for selecting the training set.
- **validation_selection_expression** (*str*) – The expression used for selecting the validation set.
- **training_selection_expression_type** (*str*) – The type of expression used for selecting the training set. ('EXPRESSION','VARIABLE',None)
- **validation_selection_expression_type** (*str*) – The type of expression used for selecting the validation set. ('EXPRESSION','VARIABLE',None)

3.8 error_metric

class eureqa.error_metric.**ErrorMetrics**(*mean_absolute_error=None*, *r2_goodness_of_fit=None*, *correlation_coefficient=None*, *maximum_absolute_error=None*, *signed_difference_between_lhs_and_rhs=None*, *area_under_roc_error=None*, *log_loss_error=None*, *rank_correlation_1_minus_r=None*, *mean_square_error=None*, *mean_squared_error_auc_hybrid=None*)

Stores the value of the error of an expression as compared to a model by a variety of different types of error-evaluation methods.

Parameters

- **mean_absolute_error** (*float*) – Mean Absolute Error
- **r2_goodness_of_fit** (*float*) – R^2 Goodness of Fit
- **correlation_coefficient** (*float*) – Correlation Coefficient

- **maximum_absolute_error** (*float*) – Maximum Absolute Error
- **signed_difference_between_lhs_and_rhs** (*float*) – Signed Difference Between LHS and RHS
- **area_under_roc_error** (*float*) – Area Under ROC Curve
- **log_loss_error** (*float*) – Log Loss Error
- **rank_correlation_1_minus_r** (*float*) – Rank Correlation
- **mean_square_error** (*float*) – Mean Squared Error
- **mean_squared_error_auc_hybrid** (*float*) – Mean Squared Error for Classification

Variables

- ***mean_absolute_error*** (*float*) – Mean Absolute Error
- ***r2_goodness_of_fit*** (*float*) – R^2 Goodness of Fit
- ***correlation_coefficient*** (*float*) – Correlation Coefficient
- ***maximum_absolute_error*** (*float*) – Maximum Absolute Error
- ***signed_difference_between_lhs_and_rhs*** (*float*) – Signed Difference Between LHS and RHS
- ***area_under_roc_error*** (*float*) – Area Under ROC Curve
- ***log_loss_error*** (*float*) – Log Loss Error
- ***rank_correlation_1_minus_r*** (*float*) – Rank Correlation
- ***mean_square_error*** (*float*) – Mean Squared Error
- ***mean_squared_error_auc_hybrid*** (*float*) – Mean Squared Error for Classification

eureqa.error_metric.**area_under_roc_error**()
Area Under ROC Curve error metric

eureqa.error_metric.**correlation_coefficient**()
Correlation Coefficient error metric

eureqa.error_metric.**log_loss_error**()
Log Loss Error error metric

eureqa.error_metric.**maximum_absolute_error**()
Maximum Absolute Error error metric

eureqa.error_metric.**mean_absolute_error**()
Mean Absolute Error error metric

eureqa.error_metric.**mean_square_error**()
Mean Squared Error error metric

eureqa.error_metric.**mean_squared_error_auc_hybrid**()
Mean Squared Error for Classification error metric

eureqa.error_metric.**r2_goodness_of_fit**()
R^2 Goodness of Fit error metric

eureqa.error_metric.**rank_correlation_1_minus_r**()
Rank Correlation error metric

eureqa.error_metric.**signed_difference_between_lhs_and_rhs**()
Signed Difference Between LHS and RHS error metric

3.9 eureqa

class eureqa.eureqa.**Eureqa**(*url='https://rds.nutonian.com'*, *user_name=None*, *password=None*, *organization=None*, *interactive_mode=False*, *save_credentials=False*, *verify_ssl_certificate=True*, *verify_version=True*, *verbose=False*, *retries=5*, *session_key=None*, *timeout_seconds=None*, *key=None*)
Represents an interface to the Eureqa server. All interactions with the server should start from this class.

Variables ***search_templates*** (*Solution*) – Provides access to predefined search templates.

Parameters

- **url** (*str*) – The URL of the eureqa server. It should be the same URL as used to access the web UI.
- **user_name** (*str*) – The user name to login into the server. It should be the same user name as used to login into the web UI. If the user name is not provided and the interactive mode is enabled, the user name will be requested during the script execution.
- **password** (*str*) – The password. If the password is not provided and the interactive mode is enabled, the password will be requested during the script execution.
- **organization** (*str*) – The name of the organization. All request to API will be executed in the context of this organization. If the organization name is not provided and the user is assigned to only one organization, then that organization will be used by default.
- **interactive_mode** (*bool*) – If set to True, enables interactive mode. In the interactive mode the script will request user name, password, and potentially other information if it is not provided or incorrect. If set to False (default), throws an exception if a login or password is incorrect, or if the two factor authentication is enabled. This is the default behaviour which prevents scripts from indefinitely waiting for the user input if they are executed automatically by a CRON job or a Windows Scheduler task.
- **save_credentials** (*bool*) – If set to True, saves user name and password to the '.eureqa_passwd' in the user directory. If after that any script on the same machine is trying to connect to the same server and does not provide credentials, it reuses the saved credentials. It does not save a temporary password when the two-factor authentication is enabled. When used with the two-factor authentication, the interactive_mode parameter should also be enabled.
- **verify_ssl_certificate** (*bool*) – If set to False will not verify SSL certificate authenticity while connecting to Eureqa.
- **verify_version** (*bool*) – If set to False will allow to use Python API library with incompatible version of Eureqa. Should only be used for the diagnostic purpose.
- **verbose** (*bool*) – If set to True will print to the console the detailed information for each request to the server. Should only be used for the diagnostic purpose.
- **retries** (*int*) – The number of attempts to establish a session before an Exception is raised
- **session_key** (*str*) – The session identifier.

- **timeout_seconds** (*int*) – The HTTP connection and transmission timeout. Any request to Eureqa API will abort with an exception if it takes more time, than set in the timeout, to either connect to the server or to receive a next data package.
- **key** (*str*) – Authentication key. Provide either this field or *password*.

Raises **Exception** – If the authentication fails or cannot be completed.

compute_error_metrics(*datasource*, *target_variable*, *model_expression*, *template_search=None*, *variable_options=[]*, *row_weight='1.0'*, *row_weight_type='uniform'*, *data_split='all'*)

Compute the *eureqa.error_metric.ErrorMetrics* for the specified model, against the specified target_variable

Parameters

- **datasource** (DataSource) – DataSource to compute error against
- **target_variable** (*str*) – Variable (or expression) to compare the model to
- **model_expression** (*str*) – Model whose error is to be computed
- **template_search** (Search) – If specified, inherit variable options from the specified search. Values specified in :variable_options: take precedence over values in this search; use it for finer-grained control instead of or on top of this argument.
- **variable_options** (VariableOptionsDict) – Override any default behavior for the specified variables. If the data contains nulls and no null-handling policy is specified, this method will return an error. A list of *eureqa.variable_options.VariableOptions* may also be provided.
- **row_weight** (*str*) – Expression to compute the weight of a row (how much that row contributes to the computed error)
- **row_weight_type** (*str*) – The type of expression to use to compute row weight (uniform, target_frequency, variable, or custom_expr)
- **data_split** (*str*) – For internal use

Returns The computed error metrics

Return type *ErrorMetrics*

create_analysis(*name*, *description=None*)

Creates an analysis.

Parameters

- **name** (*str*) – The analysis name. It will be used as the Analysis title.
- **description** (*str*) – The analysis description.

Returns An Analysis object that represents a newly created data source on the server.

Return type Analysis

create_analysis_template(*name*, *description*, *parameters*, *icon=None*)

Create a new Analysis Template on the Eureqa server.

Parameters

- **name** (*str*) – The analysis template's name. Will be used to identify the template.
- **description** (*str*) – The analysis template's description. Will be used where more space is available for an expanded description.

- **parameters** (*Parameters*) – Object describing the parameters that a user must fill in via the UI in order to specify the template's behavior
- **icon** (*str*) – The path to an icon to use in the UI for this analysis.

Returns An AnalysisTemplate object representing the template on the server

Return type AnalysisTemplate

create_data_source(*name*, *file_or_path*)

Creates a new data source on the server.

Uploads data and performs all the same preproccessing that happen when a new data set is created through the UI.

Parameters

- **name** (*str*) – A name for the new data source.
- **file_or_path** (*str*) – A path to a local CSV file with data for the data source. It can be either an absolute path or path relative to the current working directory. Alternatively, a Python file-like object.

Returns A DataSource object that represents a newly created data source on the server.

Return type DataSource

Raises **Exception** – If data source with the same name already exists and get_existing is set to False.

evaluate_expression(*datasource*, *expressions*, *template_search=None*, *variable_options=[]*, *data_split='all'*)

Evaluates the provided expression against the specified datasource. Returns the value of the evaluated computation.

Example:

values = eureqa.evaluate_expression(['x','y','x^2']) # where

values['x'] –> [1,2,3,4]

values['y'] –> [5,6,7,8]

values['x^2'] –> [1,4,9,16]

data = pandas.DataFrame(values) # convert to pandas.DataFrame

Parameters

- **datasource** (DataSource) – DataSource to perform the computation against
- **expressions** (*str*) – If only one expression is to be evaluated, that expression. If multiple expressions are to be evaluated, a list of those expressions.
- **template_search** (Search) – If specified, inherit variable options from the specified search. Values specified in :variable_options: take precedence over values in this search; use it for finer-grained control instead of or on top of this argument.
- **variable_options** (VariableOptionsDict) – Override default variable options directly for particular variables. Set interpretation of NaN values, outliers, etc. default behavior is to make no changes to the original data. By default, missing values are not filled; missing values in input data may result in missing values in corresponding computed values. A list of *eureqa.variable_options.VariableOptions* may also be provided.
- **data_split** (*str*) – For internal use only

get_all_analysis_templates()
Get a list of all Analysis Template objects currently available to this connection

Return type list of AnalysisTemplate

get_all_data_sources()
Get all data sources from the server

Returns A list of DataSource objects for all data sources within the organization.

get_analyses()
Return the list of all analyses from the server.

Return type list of Analysis

get_analysis(*analysis_id*)
Return a specific analysis from the server, by id

Parameters **analysis_id** (*str*) – The id of the analysis to return

Return type Analysis

get_data_source(*data_source_name*)
Get a data source by its name.

Searches on the server for a data source given its name.

Parameters **data_source_name** (*str*) – The name of the data source.

Returns A DataSource object if such data source exists, otherwise None.

Return type DataSource

get_data_source_by_id(*data_source_id*)
Get a data source by its id.

Searches on the server for a data source given its id.

Parameters **data_source_id** (*str*) – The ID of the data source.

Returns A DataSource object if such data source exists, otherwise None.

Return type DataSource

search_templates
Return all Search Templates available to the current connection

Return type SearchTemplates

class eureqa.eureqa.**EureqaLocal**(*instance=1*)
Represents an interface to the Eureqa Local server.

Works exactly the same as Eureqa, its parent class, except as documented.

Parameters **instance** (*int*) – Instance number. Default, 1, points to the first Eureqa launched on the machine. The second is 2, etc. Use this instead of the 'url' argument on the parent class.

All constructor arguments for the Eureqa class are also supported.

3.10 html

3.10.1 button

class eureqa.html.button.**Button**(*title*)

BETA An HTML button inside an HtmlCard

Parameters **title** (*str*) – Title of the card

class Events

Constants that identify the types of events that can trigger an action

Button.**add_action**(*card_action*, *args=None*, *kwargs=None*, *event='click'*)

Add an action to this button, to be performed when the specified event happens

Parameters

- **card_action** (*instancemethod*) – Method on an analysis-card instance to invoke. For example, *card.replace*.
- **args** (*tuple*) – Arguments to the *card_action* method
- **kwargs** (*dict*) – Keyword arguments to the *card_action* method
- **event** (*str*) – Event that triggers this action

Button.**add_replace_action**(*target_card*, *replacement_card*, *event='click'*)

Add a 'Replace' action to the specified card. Equivalent to *add_action(target_card.replace, (replacement_card,), event=event)*.

Parameters

- **target_card** (*eureqa.analysis_card.AnalysisCard*) – Card to be replaced
- **replacement_card** (*eureqa.analysis_card.AnalysisCard*) – Card to replace *target_card* with
- **event** (Events) – Event to trigger the replacement

Button.**to_html**(*html_tag='a'*)

Render this button as an HTML tag

For example: to_html('Go!', html_tag='button', style='color: blue;') would return "<button style='color: blue;' (...)>Go!</button>"

Parameters

- **html_tag** (*str*) – Tag name. For example, 'a', 'input', 'button', 'div'
- ****kwargs** – Tag parameters. For example, "style='color: blue;"'

Returns HTML string representation of this button

3.11 install_analysis_template

eureqa.install_analysis_template.**apply_module**(*module_dir*, *module_name*, *organization*, *id_*, *name*, *description*, *parameters*, *url*, *username*, *verify_ssl_certificate*, *dependency_dir*, *quiet=False*, *password=False*, *save_credentials=False*)

Create and/or fetch the analysis template as necessary, and upload the specified module to it.

Parameters

- **module_dir** (*str*) – The directory where the module to apply can be found.
- **module_name** (*str*) – The name of the module to apply.
- **organization** (*str*) – The organization containing the data.
- **id** (*str*) – The id for the module.
- **name** (*str*) – A name for the template.
- **description** (*str*) – A user-defined description for the template.
- **parameters** (*dict*) – Template parameters as dictionary.
- **url** (*str*) – The url defining the endpoint.
- **username** (*str*) – The username for authentication.
- **verify_ssl_certificate** (*bool*) – Whether to use certificate-based authentication.
- **dependency_dir** (*str*) – A directory where required modules will be found.
- **quiet** (*bool*) – Whether to use interactive mode.
- **password** (*bool*) – Whether password authentication is to be used.
- **save_credentials** (*bool*) – Whether the credentials should be saved by Eureqa.

3.12 math_block

Available math blocks:

class eureqa.math_block.**MathBlock**(*name*, *complexity*, *notation*)

Represents the building blocks which compose the mathematical models.

Parameters

- **name** (*str*) – The name
- **complexity** (*int*) – The complexity weight of the operations. The algorithm will try to avoid operations with higher complexity.
- **notation** (*str*) – The actual notation of the model block

complexity

MathBlock's complexity (settable)

disable()

Don't allow this MathBlock to be used for modeling

enable(*complexity=None*)

Allow this MathBlock to be used for modeling :param int complexity: The level of complexity of the MathBlock

enabled

Can this MathBlock be used for modeling?

name

MathBlock's name (read-only)

3.13 math_block_set

class eureqa.math_block_set.**MathBlockSet**

Tracks which known MathBlock objects are currently available in the system

Usage:

```
>>> # enable the pow building block and set its complexity to 3
>>> math_blocks.pow.enable(complexity=3)
>>> # disable the pow building block in searches
>>> math_blocks.pow.disable()
>>> # set the complexity of log to 5 (do not change enabled or disabled)
>>> math_blocks.log.complexity = 5
>>>
>>> # enable a list of blocks
>>> for block in [settings.math_blocks.pow, settings.math_blocks.exp]:
>>>     block.enable(complexity=3)
>>>
>>> # Get a list of enabled blocks w/ complexity
>>> for block in settings.math_blocks:
>>>     if block.enabled:
>>>         print block.complexity
```

abs

Absolute Value math block

Return type *MathBlock*

acos

Arccosine math block

Return type *MathBlock*

acosh

Inverse Hyperbolic Cosine math block

Return type *MathBlock*

add

Addition math block

Return type *MathBlock*

and_op

Logical And math block

Return type *MathBlock*

asin

Arcsine math block

Return type *MathBlock*

asinh

Inverse Hyperbolic Sine math block

Return type *MathBlock*

atan

Arctangent math block

Return type *MathBlock*

atanh
Inverse Hyperbolic Tangent math block

Return type *MathBlock*

ceiling
Ceiling math block

Return type *MathBlock*

complementary_error
Complementary Error Function math block

Return type *MathBlock*

const
Returns math block for Constant

Return type *MathBlock*

cos
Cosine math block

Return type *MathBlock*

cosh
Hyperbolic Cosine math block

Return type *MathBlock*

delay
Delayed Variable math block

Return type *MathBlock*

div
Division math block

Return type *MathBlock*

equal
Equal-To math block

Return type *MathBlock*

error
Error Function math block

Return type *MathBlock*

exp
Exponential math block

Return type *MathBlock*

fact
Factorial math block

Return type *MathBlock*

floor
Floor math block

Return type *MathBlock*

gauss
Gaussian Function math block

Return type *MathBlock*

greater
Greater-Than math block

Return type *MathBlock*

greater_equal
Greater-Than-Or-Equal math block

Return type *MathBlock*

if_op
If-Then-Else math block

Return type *MathBlock*

int_const
Integer Constant math block

Return type *MathBlock*

less
Less-Than math block

Return type *MathBlock*

less_equal
Less-Than-Or-Equal math block

Return type *MathBlock*

log
Natural Logarithm math block

Return type *MathBlock*

logistic
Logistic Function math block

Return type *MathBlock*

max
Maximum math block

Return type *MathBlock*

min
Minimum math block

Return type *MathBlock*

mod
Modulo math block

Return type *MathBlock*

mult
Multiplication math block

Return type *MathBlock*

neg
Negation math block

Return type *MathBlock*

not_op
Logical Not math block

Return type *MathBlock*

or_op
Logical Or math block

Return type *MathBlock*

pow
Power math block

Return type *MathBlock*

round
Round math block

Return type *MathBlock*

sign
Sign Function math block

Return type *MathBlock*

simple_moving_average
Simple Moving Average math block

Return type *MathBlock*

simple_moving_median
Simple Moving Median math block

Return type *MathBlock*

sin
Sine math block

Return type *MathBlock*

sinh
Hyperbolic Sine math block

Return type *MathBlock*

sqrt
Square Root math block

Return type *MathBlock*

step
Step Function math block

Return type *MathBlock*

sub
Subtraction math block

Return type *MathBlock*

tan
Tangent math block

Return type *MathBlock*

tanh
Hyperbolic Tangent math block

Return type *MathBlock*

two_args_atan
Two-Argument Arctangent math block

Return type *MathBlock*

var
Input Variable math block

Return type *MathBlock*

weighted_moving_average
Weighted Moving Average math block

Return type *MathBlock*

xor
Logical Xor math block

Return type *MathBlock*

3.14 missing_value_policies

3.15 organization

3.16 passwords

3.17 run_analysis_template

3.18 search

class eureqa.search.**Search**(*body*, *eureqa*)
Represents a search on the server. It should be created using the *eureqa.data_source.DataSource.create_search()* method.

Parameters

- **eureqa** (Eureqa) – A eureqa connection
- **body** (*dict*) – Class metadata as a dictionary

Variables

- **name** (*str*) – The name of the search.
- **math_blocks** (MathBlock) – The set of the *eureqa.math_block.MathBlock* objects which represents mathematical operations allowed to be used by the search algorithm. Use member properties to access the blocks; for example, *eureqa.math_block_set.MathBlockSet.add* to access the "add" MathBlock, etc. Use *eureqa.math_block.MathBlock.enable* and *eureqa.math_block.MathBlock.disable* to enable or disable a specific MathBlock.
- ***data_splitting*** (DataSplitting) – The data splitting settings for the search algorithm.

- ***error_metric*** (*str*) – One of the error metrics from *eureqa.error_metric*.
- **maximum_history_absolute_rows** (*int*) – The maximum number of rows that can be used in range based functions.
- **prior_solutions** (*list*) – The list of prior solutions.
- **row_weight** (*str*) – The row weight expression.
- ***target_expression*** (*str*) – The target expression.
- ***variable_options*** (VariableOptionsDict) – Override default behavior for the specified variables. A list of *eureqa.variable_options.VariableOptions* may also be provided.

create_solution(*solution_string*, *use_all_data=False*)

Creates a custom solution for the search. Use this if you want to compute error metrics and other statistics of a specified expression. It is also useful to compare how well a known model does against one found by Eureqa

Parameters

- **solution_string** (*string*) – the right hand side of the expression.
- **use_all_data** (*bool*) – whether to use all data or just validation data when calculating the metrics for the solution.

Return type *Solution*

delete()

Deletes the search from the server.

Raises **Exception** – search is already deleted.

evaluate_expression(*expressions*, *data_split='all'*)

Deprecated. Use *Eureqa.evaluate_expression()* instead

Parameters

- **expressions** (*str*) – Deprecated. Do not use.
- **data_split** (*str*) – Deprecated. Do not use.

get_best_solution()

Retrieves from the server the best solution found so far.

Return type *Solution*

get_data_source()

Retrieves from the server the data source information for this search.

Return type DataSource

get_most_accurate_solution()

Retrieves from the server the most accurate solution found so far.

Return type *Solution*

get_solutions()

Retrieves from the server the list of solutions found so far.

This method can be called while the search is running to check what searches are already found and make a decision whether continue the search.

Returns list of Solution objects.

Return type list

is_running
Indicates if the search currently running.

Return type bool

rename (*new_search_name*)
Change search display name.

Parameters **new_search_name** (*std*) – New search name.

stop()
Stops running the search.

submit (*time_seconds*)
Submit the search to the server to run for the specified amount of time.

This method does not guarantee to start the search immediately. The search can be queued for some time before it will start producing any results.

Parameters **time_seconds** (*int*) – The maximum amount of time to run the search. The server will stop running the search once the running time will reach this limit.

wait_until_done (*show_progress=False*, *poll_seconds=5*, *print_callback=None*)
Waits until the search stops running.

Parameters

- **show_progress** (*bool*) – whether to print the search progress while waiting.
- **poll_seconds** (*int*) – number of seconds to wait between checking progress.
- **print_callback** (*function*) – method to invoke to print the progress (sys.stdout.write by default).

3.19 search_settings

Templates for search settings:

class eureqa.search_settings.**SearchSettings**
A set of settings which should be passed into `create_search()` Use one of the templates from `search_templates()` to create an instance of this class.

Variables

- **name** (*str*) – The name of the search.
- **search_template_id** (*str*) – The type of the search, can only be one of "generic", "timeseries" or "classification".
- **target_variable** (*str*) – The target variable.
- **input_variables** (*list*) – The list of input variables.
- **math_blocks** (*list*) – The list of the `MathBlock` objects which represents mathematical operations allowed to be used by the search algorithm.
- ***data_splitting*** (`DataSplitting`) – The data splitting settings for the search algorithm.
- ***error_metric*** (*str*) – One of the error metrics from *eureqa.error_metric*.
- **maximum_history_absolute_rows** (*int*) – The maximum number of rows that can be used in range based functions.

- **prior_solutions** (*list*) – The list of prior solutions.
- **row_weight** (*str*) – The row weight expression.
- **row_weight_type** (*str*) – The row weight type expression. The default, 'uniform', requires *~eureqa.search_settings.SearchSettings.row_weight* to be unspecified. 'custom' allows arbitrary expressions.
- ***target_expression*** (*str*) – The target expression.

target_expression
The target expression to optimize.

Only set if a custom expression is required. If set, the resulting search will be treated as an "Advanced" search in the UI; the variable chooser will be replaced with an expression editor.

3.20 search_templates

class eureqa.search_templates.**SearchTemplates**(*eureqa*)
Provides a set of search settings for well known search scenarios.

Parameters **eureqa** (eureqa.Eureqa) – A eureqa connection.

classification(*name*, *target_variable*, *input_variables*)
The classification search settings template.

Parameters

- **name** (*str*) – The search name.
- **target_variable** (*str*) – The target variable.
- **input_variables** (*list*) – The list (str) of input variables.

Return type *SearchSettings*

numeric(*name*, *target_variable*, *input_variables*)
The numeric search settings template.

Parameters

- **name** (*str*) – The search name.
- **target_variable** (*str*) – The target variable.
- **input_variables** (*list*) – The list (str) of input variables.

Return type *SearchSettings*

time_series(*name*, *target_variable*, *input_variables*, *min_delay=1*, *data_custom_history_fraction=0.1*, *max_delays_per_variable=0*)
The time series search settings template.

Parameters

- **name** (*str*) – The search name.
- **target_variable** (*str*) – The target variable.
- **input_variables** (*list*) – The list (str) of input variables.
- **min_delay** (*int*) – Optionally specify the minimum number of rows used in the range functions.

- **data_custom_history_fraction** (*float*) – Optionally specify the percentage of the data to be withheld from history blocks. Specifies the maximum possible delay for a history block.
- **max_delays_per_variable** (*int*) – Optionally overrides data_custom_history_fraction to directly set the maximum possible delay for a history block.

Return type *SearchSettings*

time_series_classification(*name*, *target_variable*, *input_variables*, *min_delay=1*, *data_custom_history_fraction=0.1*, *max_delays_per_variable=0*)

The time series classification search settings template.

Parameters

- **name** (*str*) – The search name.
- **target_variable** (*str*) – The target variable.
- **input_variables** (*list*) – The list (str) of input variables.
- **min_delay** (*int*) – Optionally specify the minimum number of rows used in the range functions.
- **data_custom_history_fraction** (*float*) – Optionally specify the percentage of the data to be withheld from history blocks. Specifies the maximum possible delay for a history block.
- **max_delays_per_variable** (*int*) – Optionally overrides data_custom_history_fraction to directly set the maximum possible delay for a history block.

Return type *SearchSettings*

3.21 session

3.22 solution

class eureqa.solution.**Solution**(*body*, *search*)

Represents one of the solution found by the system for a particular search.

Parameters

- **search** (*eureqa.Search*) – The search object for this solution
- **body** (*dict*) – Class metadata as dictionary

Variables

- **target** (*str*) – The target variable.
- **model** (*str*) – The model expression.
- **complexity** (*int*) – The model complexity based on the complexity weights.
- **is_best** (*bool*) – An indicator whether the solution is considered to be the best based on its complexity and precision. The system can pick only one solution as best.
- **is_most_accurate** (*bool*) – An indicator whether the solution is the most accurate for the optimized error metric. The system can pick only one solution as most accurate.

- **optimized_error_metric** (*str*) – The error metric for which the solution was optimized. It is one of the error metrics from *error_metric*.
- **optimized_error_metric_value** (*float*) – The value of the optimized error metric.
- **search** (*Search*) – The search object to which this solution belongs.

get_all_series_error_metrics(*data_split='all'*)
Returns the metric values for each series in the dataset.

Parameters **data_split** (*str*) – For internal use only.

Returns list of :class: *~eureqa.ErrorMetrics* objects.

Return type list

get_error_metric_value(*name*)
Returns the value for the specified error metric.

Parameters **name** (*str*) – One of the error metrics from *error_metric*.

Return type float

get_single_series_error_metrics(*series_index*, *data_split='all'*)
Returns the metric values for the specified series in the dataset.

Parameters

- **series_index** (*int*) – The series to compute the metrics on
- **data_split** (*str*) – For internal use only.

Return type *~eureqa.ErrorMetrics*

3.22.1 channel

3.22.2 data_holder

class eureqa.utils.data_holder.**DataHolder**
For internal use only.

A helper class for holding arbitrary data in a 2d list, then writing to a CSV in a StringIO buffer.

add_column(*name*, *column*)
Add a column of data with the specified name to the 2d array.

Parameters

- **name** (*str*) – The name of the column.
- **column** (*list*) – The column of data.

If the name is already used in this instance, the old column will be overwritten.

get_csv_file()
Translate the accumulated 2d array of data into a CSV file inside a StringIO buffer.

Returns a StringIO buffer containing the CSV

Return type StringIO

3.22.3 image

class eureqa.utils.image.**Image**(*eureqa*, *image_filename*, *image_key*)
Represents an image which has been uploaded to the server

Parameters

- **eureqa** (Eureqa) – A eureqa connection.
- **image_filename** (*str*) – The name of the image file.
- **image_key** (*str*) – The identifying key of the image file in the objectstore image_uploads bucket.

delete()
Removes the image from the server. Caller is expected to throw away this object afterwards.

classmethod upload_from_file(*cls*, *eureqa*, *image_path*)
Uploads an image from file.

Parameters

- **cls** (*str*) – Classpath object.
- **eureqa** (Eureqa) – A eureqa connection.
- **image_path** (*str*) – The path to an image file.

Returns An Image object representing the newly uploaded image.

Return type eureqa.utils.Image

3.22.4 modulefiles

3.22.5 objectstore

class eureqa.utils.objectstore.**ObjectStore**(*eureqa*)
For internal use only. Used to get/put/delete objects from the object store.

Parameters **eureqa** (Eureqa) – a eureqa connection

classmethod construct_python_uri_full(*cls*, *organization*, *bucket*, *key=None*, *download=False*)
Given an organization, bucket and optional key, construct the absolute uri associated with those values, starting with '/api/v2'. This uri will be accessible from a python. It may not work from a Web browser.

Parameters

- **cls** (*obj*) – Classmethod object
- **organization** (*str*) – The logical organization containing the data.
- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key
- **download** (*bool*) – Whether to enable download from this URI

Returns the full objectstore uri associated with the specified bucket/key

Return type str

classmethod construct_uri(*cls*, *bucket*, *key=None*)
Given a bucket and optional key, construct the objectstore uri associated with those values.

Parameters

- **cls** (*obj*) – Classmethod object
- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key

Returns the objectstore uri associated with the specified bucket/key

Return type str

classmethod construct_uri_full (*cls*, *organization*, *bucket*, *key=None*, *download=False*)
Given an organization, bucket and optional key, construct the absolute uri associated with those values, starting with '/api'. This uri will be accessible from a Web browser. It may not work from Python.

Parameters

- **cls** (*obj*) – Classmethod object
- **organization** (*str*) – The logical organization containing the data.
- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key
- **download** (*bool*) – Whether to enable download from this URI

Returns the full objectstore uri associated with the specified bucket/key

Return type str

delete (*bucket*, *key=None*)
Delete contents of a bucket or key.

If no key is specified, we'll delete the entire bucket.

Parameters

- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key

get (*bucket*, *key=None*, *filepath=None*)
Get an object from a bucket/key in the objectstore.

If no key is specified, the response will be a list of keys in the specified bucket

If filename is specified, result is also saved to file.

Parameters

- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key
- **filepath** (*str*) – The (optional) key for this data URI

Returns the contents of the bucket/key

Return type str

classmethod get_uri_parts (*cls*, *uri*)
Given a uri of the form /this/is/a/uri, return a list of the uri components.

Parameters

- **cls** (*obj*) – Classmethod object
- **uri** (*str*) – The URI to process.

Returns a list of the uri components

Return type list

put(*bucket*, *key=None*, *raw_data=None*, *filepath=None*)
Put contents into a bucket/key.

If a key is specified, we'll overwrite the stored value for the bucket/key. Otherwise we'll create a new key in the bucket.

Parameters

- **bucket** (*str*) – Objectstore logical container
- **key** (*str*) – Objectstore key
- **raw_data** (*bool*) – The raw data if provided
- **filepath** (*str*) – The filepath if desired

Returns the response from the objectstore

Return type str

3.22.6 utils

class eureqa.utils.utils.**Throttle**
Decorator that ensures the wrapped function is not called twice within one seconds. sleeps for a second if it is called within 1 second of the last call must wrap a function that takes at least one argument

eureqa.utils.utils.**remove_files_in_directory**(*dir_path*)
Removes all files in the specified directory.

Parameters **dir_path** (*str*) – Path to files to be removed.

3.23 variable_details

class eureqa.variable_details.**VariableDetails**(*body*, *data_source*)
Represents a variable_details object on the server.

Parameters

- **data_source** (DataSource) – The data source definition for the variables.
- **body** (*dict*) – Class metadata as dictionary

Variables

- **name** (*str*) – The name of the variable.
- **display_name** (*str*) – The name to display in the ui‘.
- **expression** (*str*) – The expression used to create the variable if it is a custom variable
- **min_value** (*int*) – The smallest value.
- **max_value** (*int*) – The largest value.
- **mean_values** (*int*) – The mean value.
- **standard_deviation** (*int*) – The standard deviation of the values.
- **distinct_values** (*int*) – The number of distinct values.

- **missing_values** (*int*) – The number of missing values.
- **error_metric** (*str*) – One of the error metrics from *eureqa.error_metric*.
- **datatype** (*str*) – Whether it is a boolean or numeric variable
- **num_zeros** (*int*) – The number of zero values.
- **num_ones** (*int*) – The number of one values if it is a boolean variable.
- **categories** (*list*) – The categories.
- **data_source** (*DataSource*) – The DataSource the VariableDetails belongs to

update_categories(*categories*)
Update the display name of the variable

Parameters **categories** (*list*) – The categories to set for the variable

update_display_name(*display_name*)
Update the display name of the variable

Parameters **display_name** (*str*) – The name to show for the variable in the ui

3.24 variable_options

class eureqa.variable_options.**VariableOptions**(*name=None*, *smoothing_enabled=False*, *smoothing_along=None*, *smoothing_percent=None*, *smoothing_weight=None*, *missing_value_policy='column_iqm'*, *remove_outliers_enabled=False*, *outlier_threshold=None*, *normalize_enabled=False*, *normalization_offset=None*, *normalization_scale=None*)
Represents a set of settings for a variable that can be included into a search.

Parameters

- **name** (*str*) – The name of the variable. It is the same as a column name in the list of data source columns.
- **smoothing_enabled** (*bool*) – Whether the smoothing capability is enabled or not
- **smoothing_along** (*int*) – The x variable used to apply smoothing.
- **smoothing_percent** (*int*) – The amount of smoothing to apply, where 0% indicates none and 100% indicates max smoothing.
- **smoothing_weight** (*int*) – A variable or expression which indicates the weights to apply to each row during smoothing.
- **missing_value_policy** (*str*) – A policy name for the processing of missing values. One of the values from missing_value_policies module can be used.
- **remove_outliers_enabled** (*bool*) – Enables removal of rows which contain outlier.
- **outlier_threshold** (*float*) – The threshold at which a point is considered an outlier and will be removed.
- **normalize_enabled** (*bool*) – Enables normalization of the variable.

- **normalization_offset** (*int*) – A constant offset to subtract from each value.
- **normalization_scale** (*int*) – A constant factor to divide each value by before adding the offset.

Variables

- ***name*** (*str*) – The name of the variable. It is the same as a column name in the list of data source columns.
- **missing_values_enabled** (*bool*) – Enables additional processing of missing values.
- **missing_value_policy** (*str*) – A policy name for the processing of missing values. One of the values from missing_value_policies module can be used. missing_value_policies.column_mean is used if the additional processing of missing values is enabled but no policy name provided.
- **remove_outliers_enabled** (*bool*) – Enables normalization of the variable.
- **outlier_threshold** (*float*) – The threshold at which a point is considered an outlier and will be removed.

3.25 variable_options_dict

class eureqa.variable_options_dict.**VariableOptionsDict**(...)

Create a dictionary of *eureqa.variable_options.VariableOptions* objects in which the key is the *eureqa.variable_options.VariableOptions* name

This class is used primarily to hold variable parameter overrides for *eureqa.eureqa.Eureqa* and *eureqa.search.Search* methods.

add(*variable_options*)

Add an existing *eureqa.variable_options.VariableOptions* to this dictionary.

Parameters **variable_options** (VariableOptions) – *eureqa.variable_options.VariableOptions* object to add

e

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Y