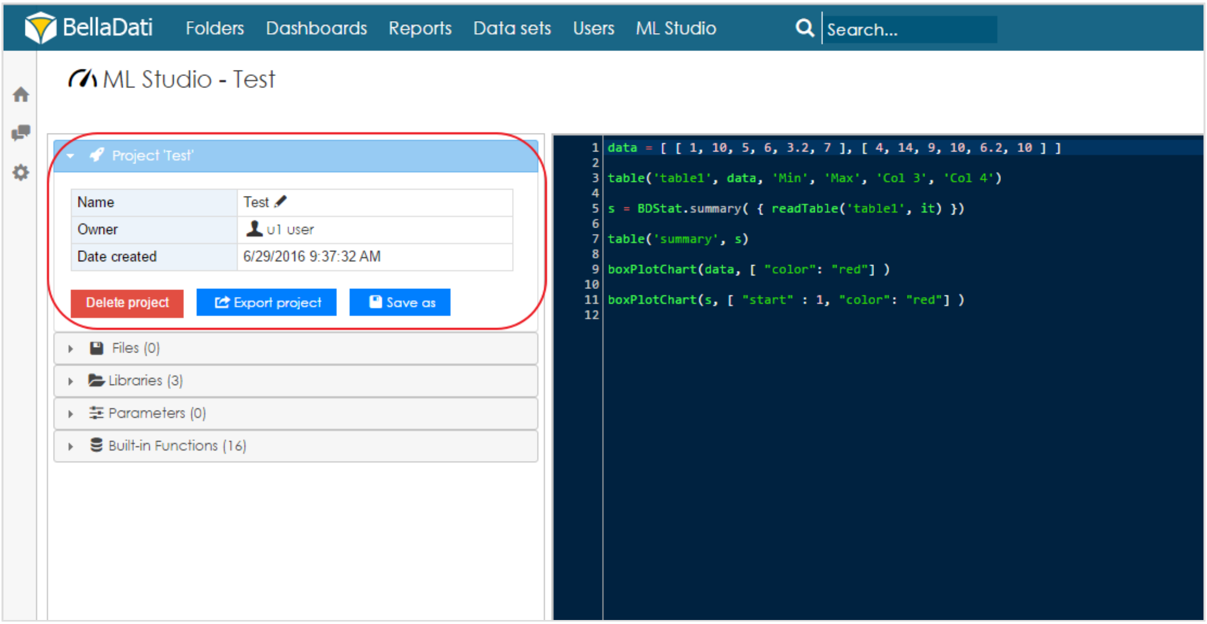
Projects

Projects are the basic building unit of ML Studio. Each project consists of several parts (not all of them necessary):

- Project information
- Files
- Tables
- Charts
- Libraries
- Variables
- Built-in functions

Project Information

Each project has its name, owner, date of creation and description.



Name	Test
Owner	u1 user
Date created	6/29/2016 9:37:32 AM

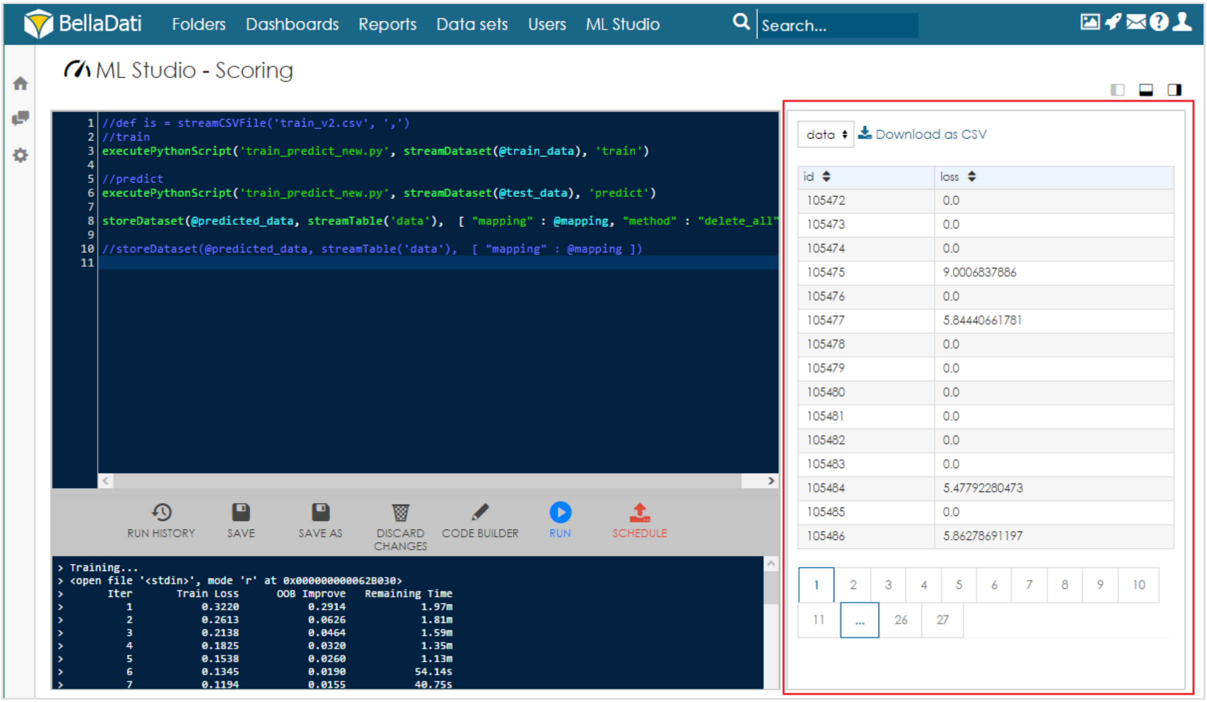
```
data = [ [ 1, 10, 5, 6, 3.2, 7 ], [ 4, 14, 9, 10, 6.2, 10 ] ]

table('table1', data, 'Min', 'Max', 'Col 3', 'Col 4')

s = BDStat.summary( { readTable('table1', it) })

table('summary', s)

boxPlotChart(data, [ "color": "red"] )

boxPlotChart(s, [ "start" : 1, "color": "red"] )
```

Files

Files can be attached to a project. They can contain Python scripts, source data, testing data etc. These files can be accessed by using functions, such as *readXLSFile()*, *readCSVFile()* and *executePythonScript()*.

Tables

Tables can be used together with charts as temporary output of the project. It can be used during the modelling to see the results.



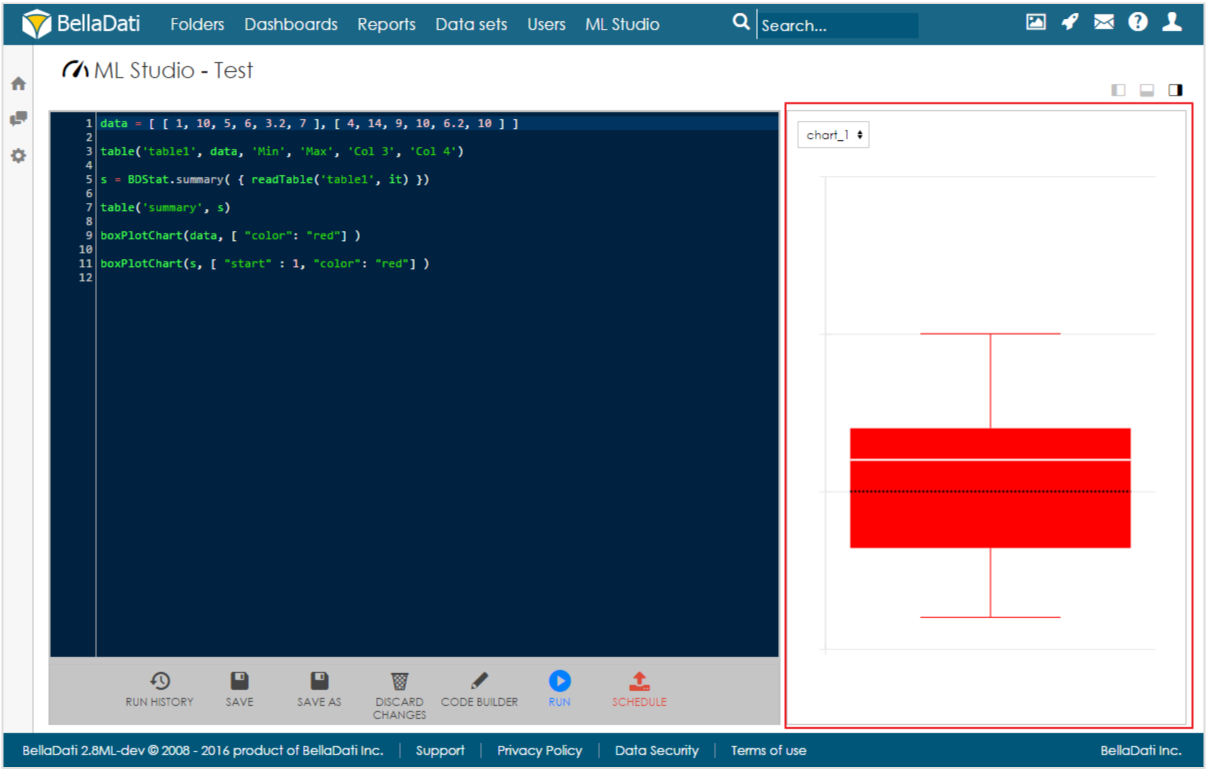
```
//def is = streamCSVFile('train_v2.csv', ',')
//train
executePythonScript('train_predict_new.py', streamDataset(@train_data), 'train')

//predict
executePythonScript('train_predict_new.py', streamDataset(@test_data), 'predict')

storeDataset(@predicted_data, streamTable('data'),  [ "mapping" : @mapping, "method" : "delete_all"

//storeDataset(@predicted_data, streamTable('data'),  [ "mapping" : @mapping ])
```

```
> Training...
> <open file '<stdin>', mode 'r' at 0x000000000062B030>
>       Iter       Train Loss      OOB Improve   Remaining Time
>          1           0.3220           0.2914            1.97m
>          2           0.2613           0.0626            1.81m
>          3           0.2138           0.0464            1.59m
>          4           0.1825           0.0320            1.35m
>          5           0.1538           0.0260            1.13m
>          6           0.1345           0.0190           54.14s
>          7           0.1194           0.0155           40.75s
```

id	loss
105472	0.0
105473	0.0
105474	0.0
105475	9.0006837886
105476	0.0
105477	5.84440661781
105478	0.0
105479	0.0
105480	0.0
105481	0.0
105482	0.0
105483	0.0
105484	5.47792280473
105485	0.0
105486	5.86278691197

Charts

Charts are another option how to see results of the modelling.



```
data = [ [ 1, 10, 5, 6, 3.2, 7 ], [ 4, 14, 9, 10, 6.2, 10 ] ]

table('table1', data, 'Min', 'Max', 'Col 3', 'Col 4')

s = BDStat.summary( { readTable('table1', it) })

table('summary', s)

boxPlotChart(data, [ "color": "red"] )

boxPlotChart(s, [ "start" : 1, "color": "red"] )
```

Libraries

Libraries are ready-made packs of mathematical and statistical components and functions which can be used in the project. By default ML Studio contains these libraries:

- Math3
- Javaml

- Javastat

Variables

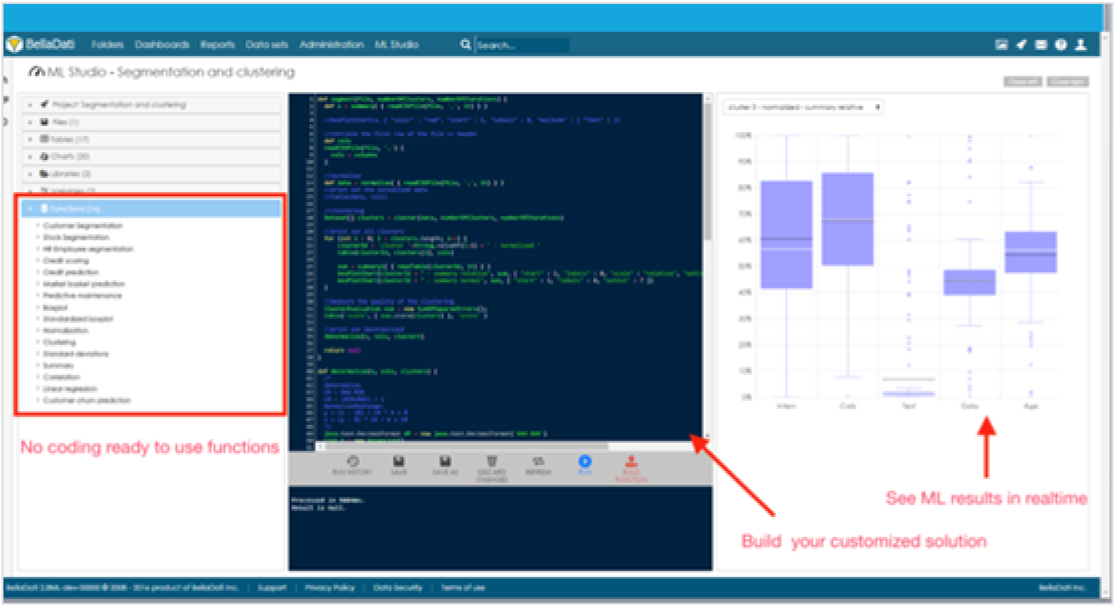
Variables are used for parametrization of projects. This enables the project to be more universal. For more information see Parameters in ML Studio.

Prebuilt functions

ML Studio has been used by BellaDati to prebuilt functions for the most frequent IoT tasks as predictive maintenance and situational intelligence as customer segmentation or market basket analytics. These can be used without the programming, only using graphical interface for the specification of parameters. At the same time it allows partners and organizations to build their own statistical and ML cases.

All of the following functions are supported and solutions can be built:

- Max, min, standard deviation, mean, median, 1st quartile, 3rd quartile, in the predefined function called summary. Graphic forms is available in the form of box plot, relative box plot, outliers and histogram.
- Confidence interval and hypothesis testing using basic statistical capabilities (e.g. T Test, Chi-Square) are available.
- The distribution of objects into predefined classes or categorical variables is enabled.
- The prediction of the value of an unknown, continuous variable is available.
- Available algoritmus allows to identify logical groups in which to place similar objects.
- Classification, estimation or clustering about a value or behaviour are available.
- Functions to determine which objects can be expected to co-occur with other objects (e.g. market basket analysis) are available.
- Functions that are able to identify the independent variables that have the biggest impact on a dependent variable are available.
- ML studio functions are able to parse unstructured content to create structured data objects (i.e. columns) that can be used for predictive modelling.
- ML studio supports support real-time scoring from models that use historical and real-time data.
- ML studio supports user-defined algorithms in a plug-and-play fashion and import / export via PMML.
- ML studio provides advanced search capabilities, for example given a registration number, the solution can determine the entity (ROC, ROB, LLP, etc.), or suggest the closest match. Search by keyword should also be supported, for example given a partial name.
- Decision trees and neural networks are available.



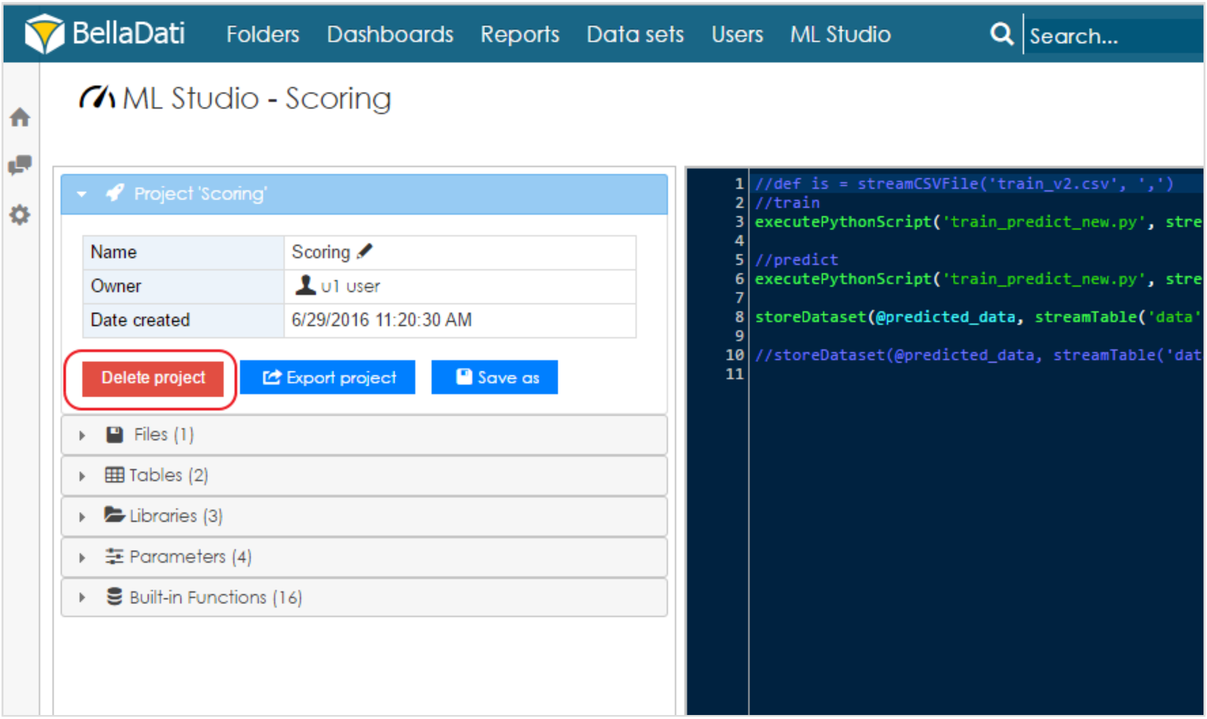
No coding ready to use functions

See ML results in realtime

Build your customized solution

Deleting Project

A project can be deleted by clicking on the **Delete project** button under Project information. Please note that this process is IRREVERSIBLE.



Name	Scoring
Owner	u1 user
Date created	6/29/2016 11:20:30 AM

```
//def is = streamCSVFile('train_v2.csv', ',')
//train
executePythonScript('train_predict_new.py', stre

//predict
executePythonScript('train_predict_new.py', stre

storeDataset(@predicted_data, streamTable('data'

//storeDataset(@predicted_data, streamTable('dat

```