Prevision.io Documentation

Gerome Pistre

déc. 21, 2020

Reference documentation

CHAPITRE 1

Getting started

Prevision.io is an automated SaaS machine learning platform that enables you to create and deploy powerful predictive models and business applications in one-click.

Prevision.io has distinguish 2 environments :

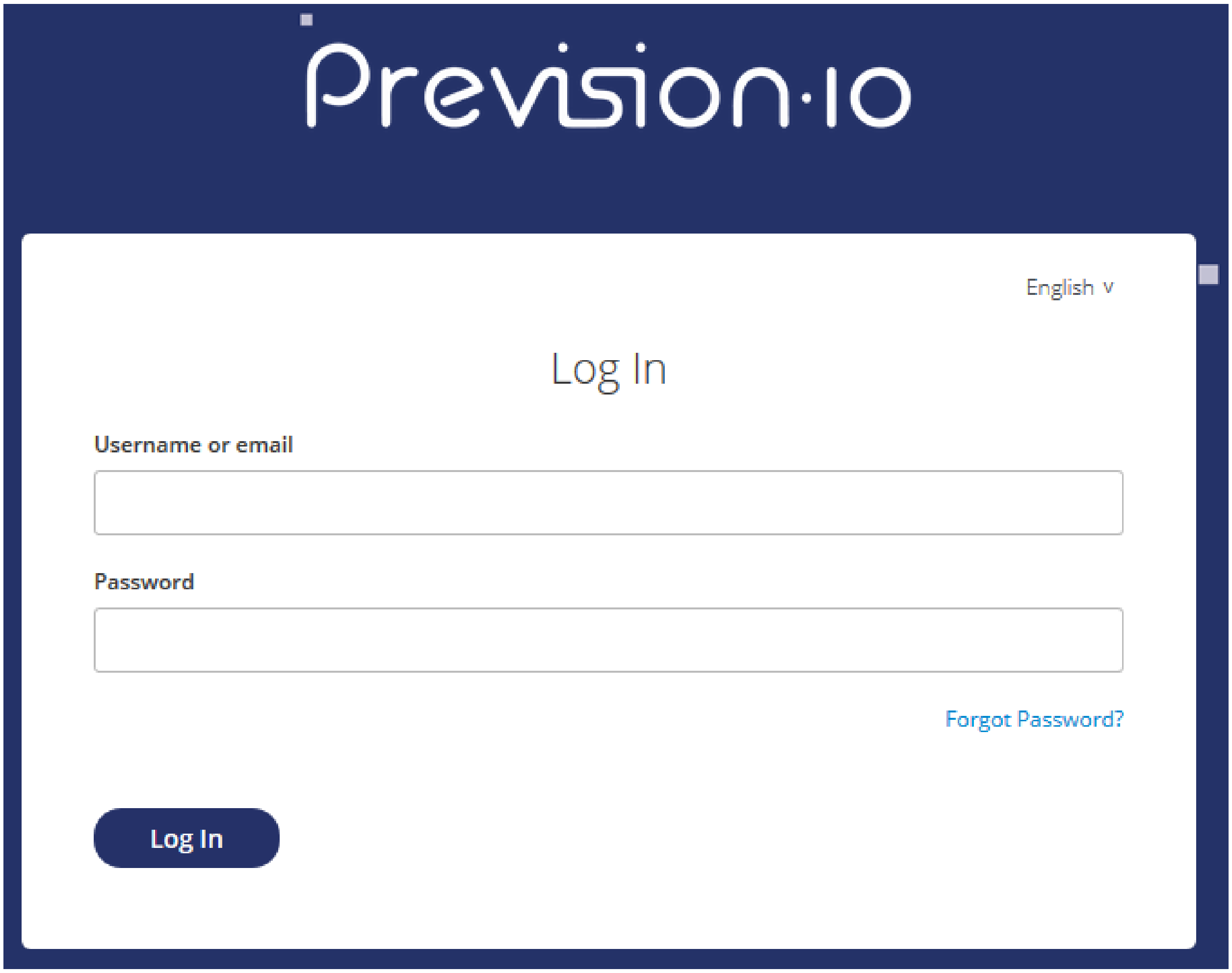
- The `Studio`, in which you can create predictive model powered by automated machine learning
- The `Store`, in which you can deploy ressources (such as model, dashboard or notebooks) directly to your business users

If you have any question or remark, you can reach us through the contact form accessible in the user menu (see below).

1.1 Connection

In order to connect to Prevision.io, you need to open a recent browser (Chrome or Firefox) to the following address : https://xxx.prevision.io [xxx is the subdomain related to your company]

If you are not already logged in, you'll land to the following screen :

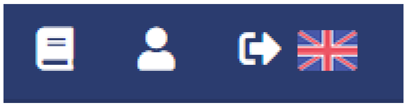


Connection is done with email address and password. If you have forgotten your password, you can reset it by clicking on the « Forgotten password » link.

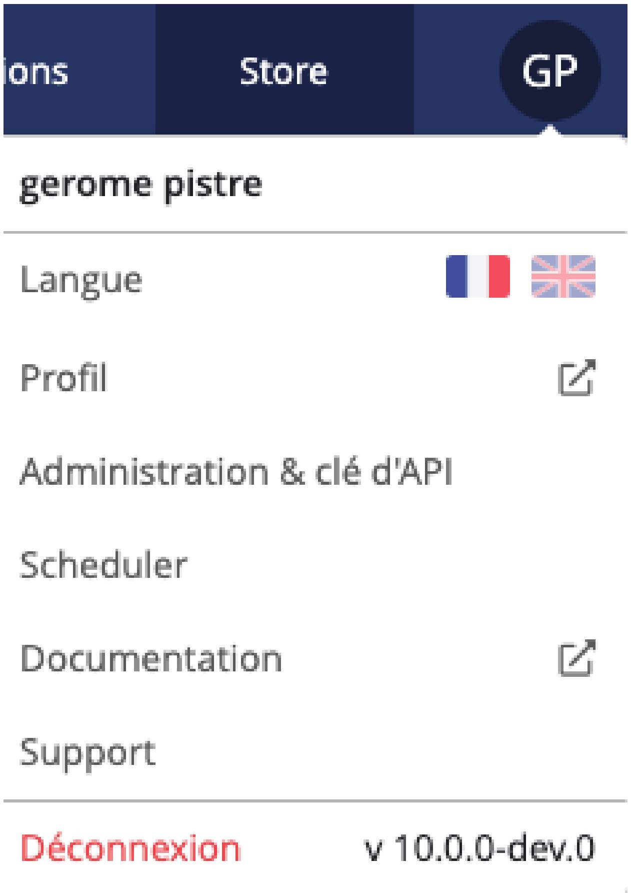
1.2 Navigation

1.2.1 Top navigation bar

Once connected, navigation is done thanks to a bar located at the top of your screen.



1.2.2 User menu



At the top right of the screen, the user menu allows you to :

- Switch displayed language between :
  - English
  - French
- Access users settings
- Access the API key configuration and the admin screen (if you have the admin rights)
- Access the scheduler
- Access the documentation
- Access the support and contact page
- View the current version number
- Logout

CHAPITRE 2

Data

Prevision.io allows you to import data into your environment for further processing. Data can come from remote persistent sources (e.g. SQL database, HIVE. . .) or can be imported directly (e.g. CSV data, ZIP,. . .)

To carry out the import of data, we distinguish between 4 notions :

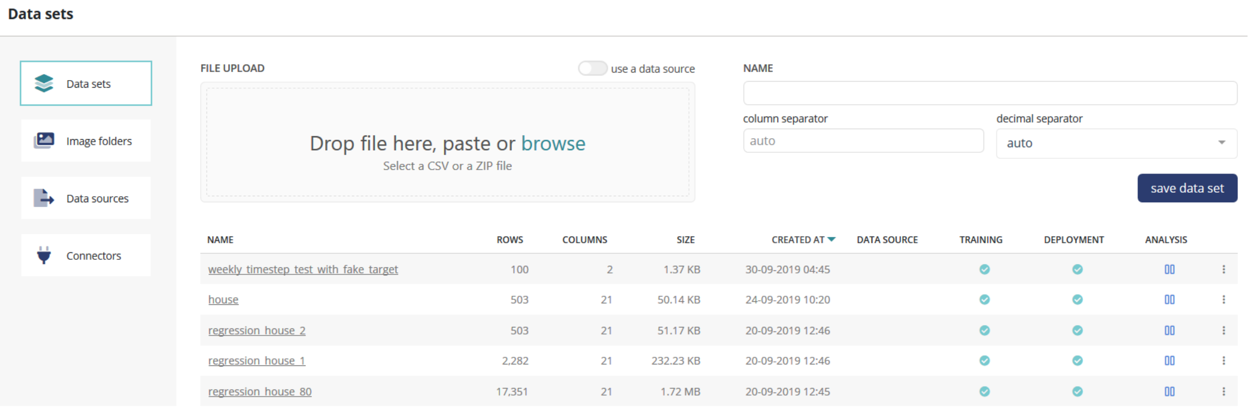
- `Data Sets` : a data set that can be a snapshot of a data source at a given time or simply a CSV or ZIP imported directly
- `Image folders` : a data folder containing images set that can come from a ZIP imported directly
- `Data Sources` : a connector completed by a query, a base / table, a path
- `Connectors` : a pointer to a persistent external environment containing data

2.1 Data Sets

Data Sets are data that can be manipulated in Prevision.io. They are in tabular form and are derived :

- Either from files (CSV, ZIP)
- Either from a Data Source at a given time (snapshot)

All Data Sets are presented in a table with their main characteristics.



NAME

column separator: auto

decimal separator: auto

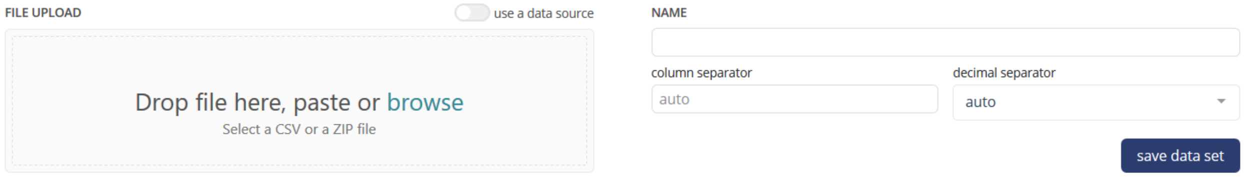
NAME	ROWS	COLUMNS	SIZE	CREATED AT	DATA SOURCE	TRAINING	DEPLOYMENT	ANALYSIS	
weekly timestep test with fake target	100	2	1.37 KB	30-09-2019 04:45		✓	✓	▯▯	⋮
house	503	21	50.14 KB	24-09-2019 10:20		✓	✓	▯▯	⋮
regression house 2	503	21	51.17 KB	20-09-2019 12:46		✓	✓	▯▯	⋮
regression house 1	2,282	21	232.23 KB	20-09-2019 12:46		✓	✓	▯▯	⋮
regression house 80	17,351	21	1.72 MB	20-09-2019 12:45		✓	✓	▯▯	⋮

By clicking on the . . . at the far right of the table, it is also possible to :

- Edit the Data Set name
- Download the Data Set in a ZIP
- Start analysis of the Data Set
- Remove the Data Set

2.1.1 Create a Data Set from a local file

To create a new Data Set from a local, simply drag n" drop a file in the upper left box (or browse for it). CSV file and ZIP containing one CSV file are supported. You can then choose the Data Set name and select separators before saving the Data Set. Please note that the `auto` detection should work in most cases.

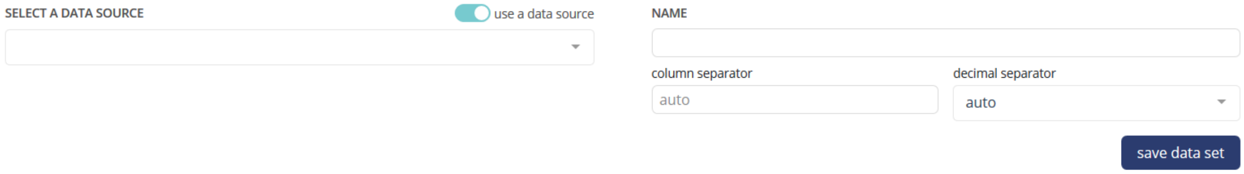


column separator: auto

decimal separator: auto

2.1.2 Create a Data Set from a Data Source

If you want to snapshot a previously created Data Source, simply create a Data Set from it. No additional information is required, except the name you wish to give to the Data Set.



column separator: auto

decimal separator: auto

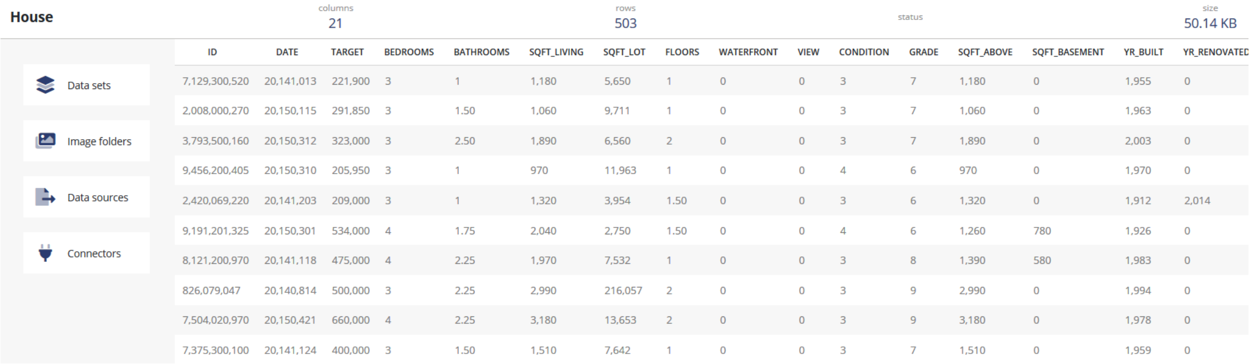
2.1.3 Once created

The Data Set name will appears in the bottom table. We will display :

- Data Set name
- Number of rows
- Number of columns
- Size
- Date of creation
- If it is linked to a Data Source
- A parsed indicator :
  - Spinning : Data Set is beeing processed for training / prediction (checking it is tabular, checking data types, . . .)
  - Green : Data Set is ready for beeing trained / predicted on
  - Red : Data Set can't be trained / predicted on. This is blocking and indicates a structural error on it
- A deployment indicator :
  - Spinning : Data Set is beeing processed for deployment (calculating drift)
  - Green : Data Set is ready for beeing deployed once a model has been trained on it
  - Red : Data Set drift can't be monitored if a use case linked to it is deployed (= non blocking)
- An analysis indicator :
  - Paused : Data Set has no analysis done on it (default behavior - it can't be seen in the Data Explorer)
  - Spinning : Data Set analysis is beeing computed
  - Green : Data Set can be analysed in the Data Explorer
  - Red : Data Set can't be analysed in the Data Explorer because of an error

- A ... menu allowing you to :
  - Edit the Data Set name
  - Create a use case from the Data Set
  - Start / Stop / Explore the Data Set
  - Download the Data Set
  - Remove the Data Set

Also, a click on the Data Set name will display the top 10 rows of it :



House

columns 21 | rows 503 | status | size 50.14 KB

ID	DATE	TARGET	BEDROOMS	BATHROOMS	SQFT_LIVING	SQFT_LOT	FLOORS	WATERFRONT	VIEW	CONDITION	GRADE	SQFT_ABOVE	SQFT_BASEMENT	YR_BUILT	YR_RENOVATED
7,129,300,520	20,141,013	221,900	3	1	1,180	5,650	1	0	0	3	7	1,180	0	1,955	0
2,008,000,270	20,150,115	291,850	3	1.50	1,060	9,711	1	0	0	3	7	1,060	0	1,963	0
3,793,500,160	20,150,312	323,000	3	2.50	1,890	6,560	2	0	0	3	7	1,890	0	2,003	0
9,456,200,405	20,150,310	205,950	3	1	970	11,963	1	0	0	4	6	970	0	1,970	0
2,420,069,220	20,141,203	209,000	3	1	1,320	3,954	1.50	0	0	3	6	1,320	0	1,912	2,014
9,191,201,325	20,150,301	534,000	4	1.75	2,040	2,750	1.50	0	0	4	6	1,260	780	1,926	0
8,121,200,970	20,141,118	475,000	4	2.25	1,970	7,532	1	0	0	3	8	1,390	580	1,983	0
826,079,047	20,140,814	500,000	3	2.25	2,990	216,057	2	0	0	3	9	2,990	0	1,994	0
7,504,020,970	20,150,421	660,000	4	2.25	3,180	13,653	2	0	0	3	9	3,180	0	1,978	0
7,375,300,100	20,141,124	400,000	3	1.50	1,510	7,642	1	0	0	3	7	1,510	0	1,959	0

2.2 Dataset page

In this screen you'll find :

- General information about the dataset :
  - Number of features
  - Number of rows
  - Number of cells (rows * features)
  - Number of missing value (%)
- Feature type distribution
- Correlation matrix (top correlation only)
- Macro analysis of features
  - Feature name
  - Feature type
  - % of missing values in the dataset

2.2.1 Data Explorer

The Data Explorer is a specific module that aim to detect similarities between samples of your dataset. It uses a combination of Dimension reduction algorithms for representing your dataset into a vector space, sometimes called **embedding** By using it, you're being able to :

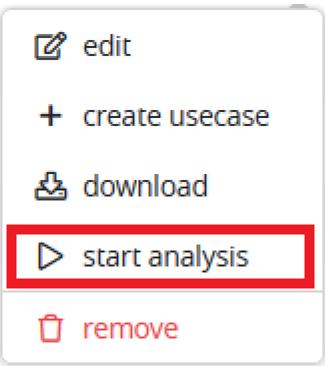
- Visually observe cluster
- see wich samples are the most similar to a selected one, for exemple a Customer in his buying habits
- See in wich population a given feature, like expenses, is present or higher
- Have a global view of your data

The Data explorer is often use as a pre-analysis of dataset, as it uses unsupervised alogorithm, but it can be uses as a standalone feature. Once the *embedding* has been generated you can request them by API or download them for use in a thrd party tool like Excel.

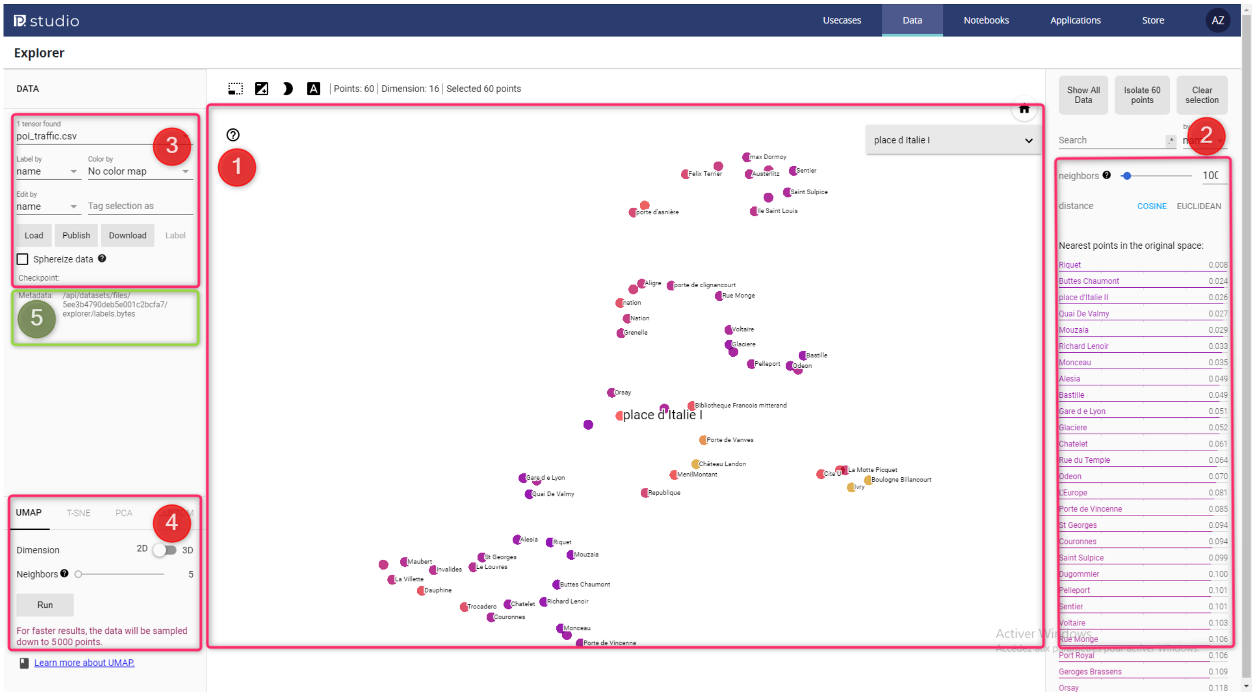
Launching the analysis of a Data set

Embedding are generated from any dataset and should be manually launched, as they require computing power.

Once a tabular dataset have been uploaded in the data tab, your can launch an analysis. To do so, click on the ... icon located at the right of the row describing the selected Data set and click on `start analysis`



The analysis indicator will start spinning. Once green, the Data set can be seen in the Data Explorer. To do so, click on the ... icon located at the right of the row describing the selected Data set and click on `explorer`



Points: 60 | Dimension: 16 | Selected 60 points

1 tensor found
poi_traffic.csv

Label by: name
Color by: No color map
Edit by: name
Tag selection as

□ Sphereize data
Checkpoint:
Metadata: /api/datasets/files/5ee3b4790deb5e001c2bcfa7/explorer/labels.bytes

Dimension 2D 3D
Neighbors 5

For faster results, the data will be sampled down to 5000 points.

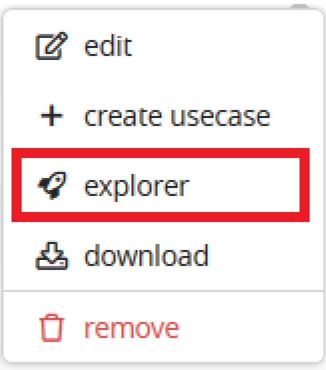
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

Riquet	0.008
Buttes Chaumont	0.024
place d'Italie II	0.026
Quai De Valmy	0.027
Mouzaia	0.029
Richard Lenoir	0.033
Monceau	0.035
Alesia	0.049
Bastille	0.049
Gare d e Lyon	0.051
Glaciere	0.052
Chatelet	0.061
Rue du Temple	0.064
Odeon	0.070
L'Europe	0.081
Porte de Vincenne	0.085
St Georges	0.094
Couronnes	0.094
Saint Sulpice	0.099
Dugommier	0.100
Pelleport	0.101
Sentier	0.101
Voltaire	0.103
Rue Monge	0.106
Port Royal	0.106
Geroges Brassens	0.109
Orsay	0.118

The Explorer

The Data Explorer is now accessible and will give you a visual representation in 2 or 3 dimensions of the selected Data Set. This representation is a dimention reduction constrained to 2 or 3 dimension, apply on the embedded vectors, that may be of an higher dimension. There are five important section in the data explorer.



(1) Graphical projection

The main screen is a visual representation of the dataset. Each point is a sample of your dataset (up to 5000). You can pan and zoom and if you click on a point, or use the selecting box tool, some more info are displayed.

In this representation, point are group by similarities as much as possible, meaning that if two points are near in this space, the samples share some important similarities.

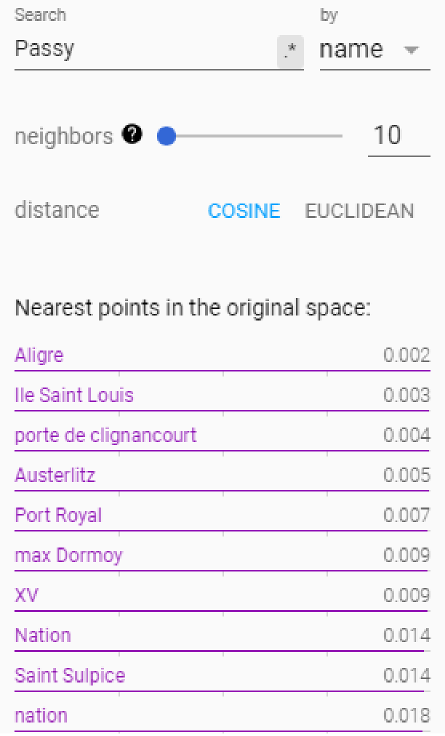
The nature of the displayed information are selected on the section (3)

(2) Search and similarities

The second section is a dynamic list of similar sample.

You can search any sample from any feature. For example if your dataset has an index with name, you can search a sample by using its name but you can too search all the sample that have « RPG » as type or « 5 » for size.

Once a sample is selected, it and a list of similar are is highlighted in the main section. They can be further isolated by clicking on the « isolate N points » button on top of the section..



Search Passy .* by name

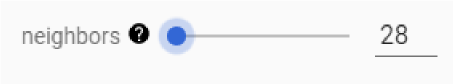
neighbors 10

distance COSINE EUCLIDEAN

Nearest points in the original space:

Aligre	0.002
Ile Saint Louis	0.003
porte de clignancourt	0.004
Austerlitz	0.005
Port Royal	0.007
max Dormoy	0.009
XV	0.009
Nation	0.014
Saint Sulpice	0.014
nation	0.018

The number of similar samples to display can be choosen with the « neighbors » slider

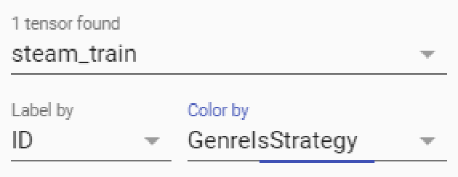


neighbors 28

(3) Labels

Section 3 main purpose is to set labels displayed in section 1. Here you can set :

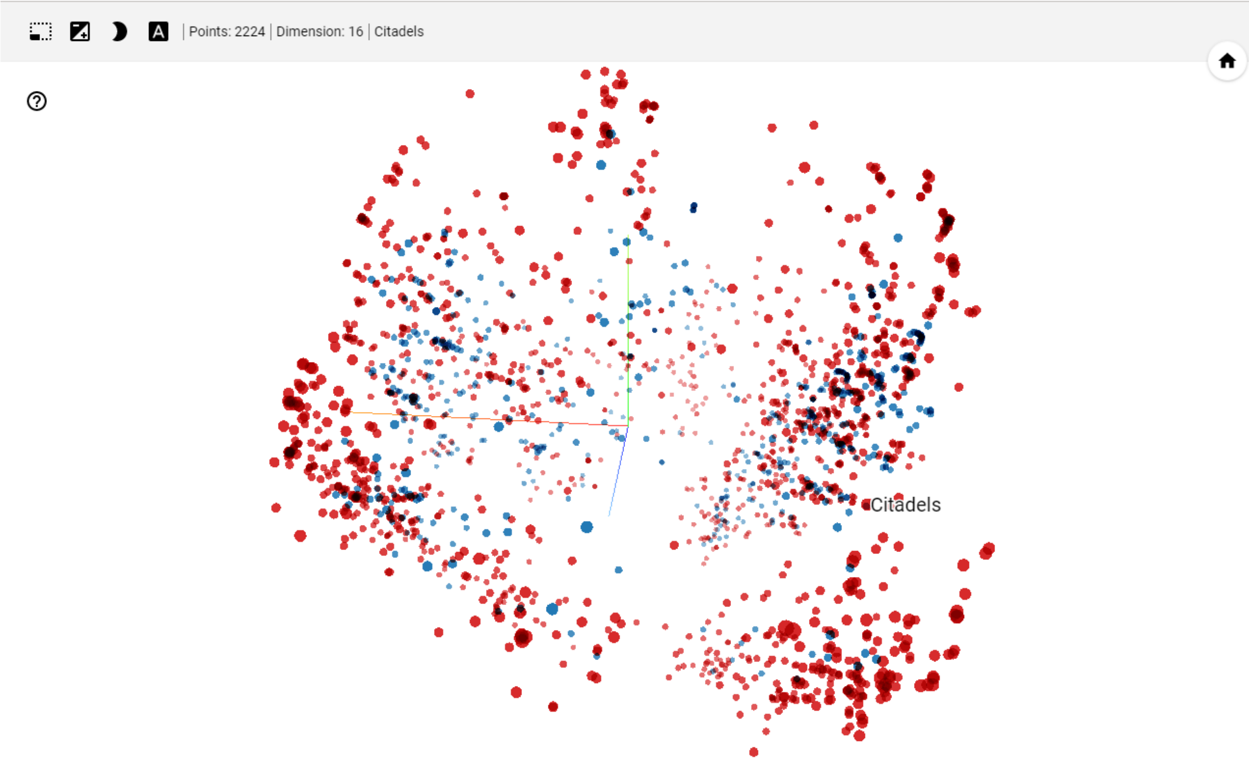
- the label displayed above each point
- the feature use for coloring each point :



1 tensor found: steam_train

Label by: ID

Color by: GenreIsStrategy



(4) Segmenting and clustering

Section 4 is all about Segmenting and clustering your samples.

Here you can choose an algorithm and tune its parameter to display the most similar point together. Thus, you can start to observe sample cluster, or segment of data that represent big group that share important similarities.

Yet, as we try to project a lot of dimension in a smaller space (3D or 2D), note that this algorithm are just for display and shaping human decision. A lot of the process is a little bit subjective and further conclusion should be driven by a supervised algorithm.

Anyway, here you can choose between 3 algorithms :

- PCA : the quickest and simplest algorithm. Clicking on the PCA tab immediately led to a 3D representation of your samples. Yet, this is a very simple algorithm that only show sample variability along 3 axes.
- T-SNE : once you click on the T-SNE tab, a process of convergence is launched. T-SNE is a very time consuming algorithm but that can lead to very accurate segmentation. You can change its parameters and click on « Stop » button then « Re-run » it. But in most of case it's better to already know this algorithm to use it.
- UMAP : UMAP is a good alternative to T-SNE and PCA. Quicker than T-SNE , it offers better result thant PCA. THe only parameters is « Neighbors », that change the size of clusters. The more neighbors you ask for, the bigger the cluster.

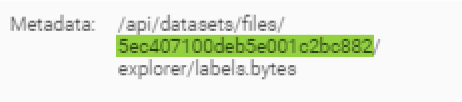
We recommend to use UMAP in most of case.

(5) API informations

The 5th part is only about API informations.

When launching a dataset Analysis, the platform built an embedding of dataset, namely, it projects each sample of the dataset to a vector. This embedding is attached to the dataset and can be retreived wit the dataset ID. Then you can use it for running any mathematical operation, in most of case a distance, that can be ran on vectors.

Section 5 of the tools gives you the Id of your dataset :



Metadata: /api/datasets/files/5ec407100deb5e001c2bc882/explorer/labels.bytes

With it you can access several URL :

- GET https://<YOUR_DOMAIN>.prevision.io/api/datasets/files/<DATASET_ID>/download : get the original dataset
- GET https://<YOUR_DOMAIN>.prevision.io/api/datasets/files/<DATASET_ID> : JSON info about your dataset
- GET https://<YOUR_DOMAIN>.prevision.io/api/datasets/files/<DATASET_ID>/explorer : JSON info about the embeddding
- GET https://<YOUR_DOMAIN>.prevision.io/api/datasets/files/<DATASET_ID>/explorer/tensors.bytes : numpy files of embeddings
- GET https://<YOUR_DOMAIN>.prevision.io/api/datasets/files/<DATASET_ID>/explorer/labels.bytes : tsv files of labels

The embedding files (tensor.bytes) is a numpy float 32 file whom shape is in the json file if explorer URL. You can read it with the following python code for example :

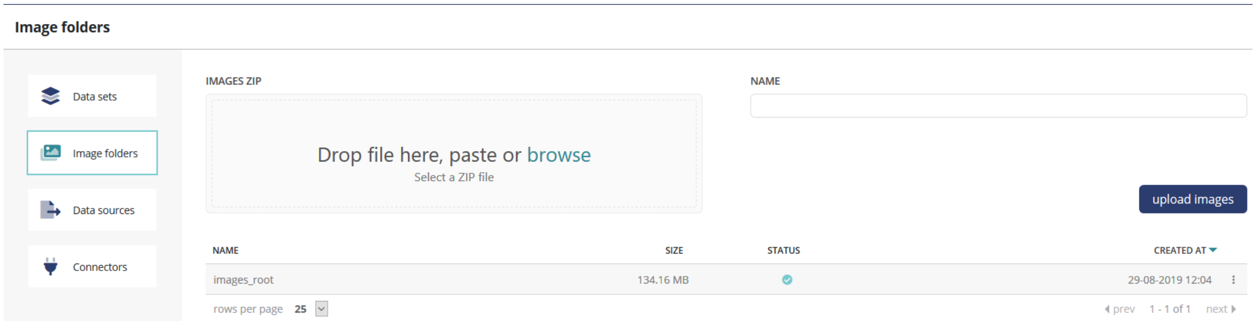
```
req = Request('https://<YOUR_DOMAIN>.prevision.io/ext/v1/datasets/files/<DATASET_ID>/explorer/tensors.bytes')
req.add_header('Authorization',<YOUR_TOKEN> ) #get YOUR_TOKEN in the admin page
content = urlopen(req).read()
vec = np.frombuffer(BytesIO(content).read(), dtype="float32").reshape(u,v) # u,v is the shape gotten in /ext/v1/datasets/files/<DATASET_ID>/explorer
print(vec.shape)
```

Please note that you can use SDK's functions in order to simplifies this process.

2.3 Image Folders

An Image Folder is a ZIP containing 1 to n images that will be needed for training image use cases.

The process of creating an Image Folder is similar compared to a Data Set. However, less analytics will be computed on it because of its nature.



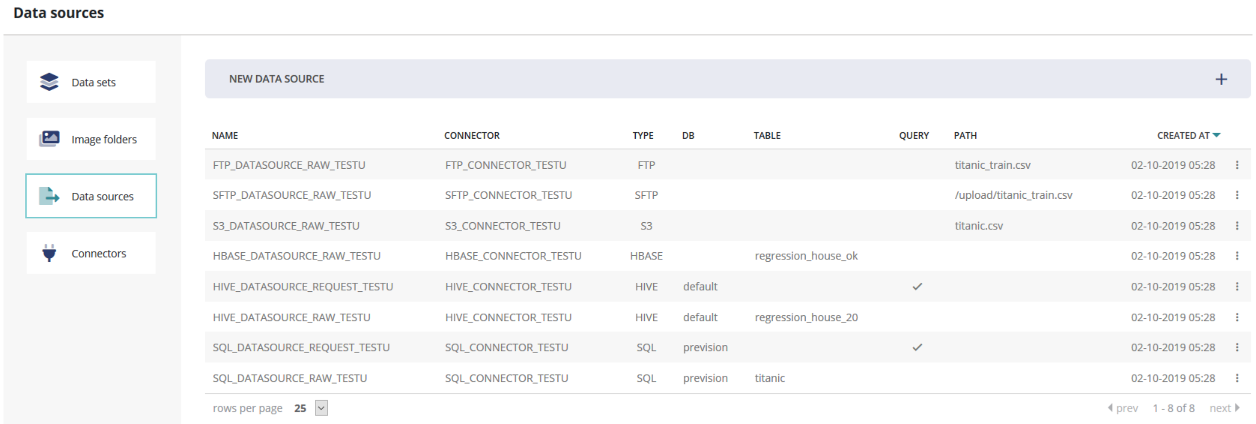
NAME	SIZE	STATUS	CREATED AT
images_root	134.16 MB	✓	29-08-2019 12:04

rows per page 25 | prev 1 - 1 of 1 next

2.4 Data Sources

Data Sources require the existence of a connector, which is supplemented by information in order to point to a specific data source (query, database + table, file name, . . .)

All Data Sources are presented in a table with their main characteristics.



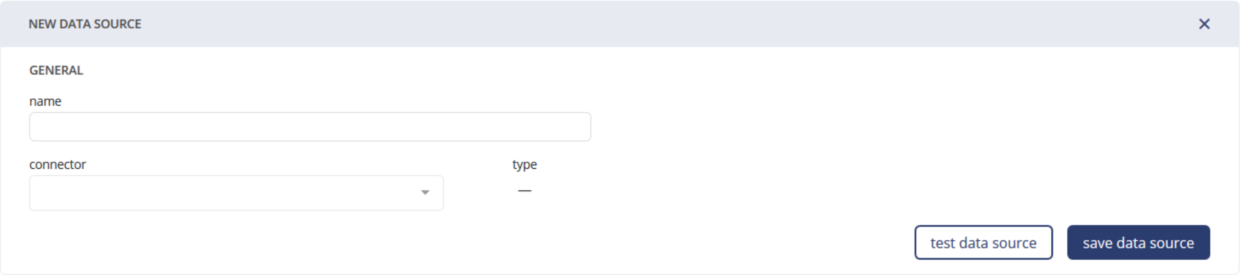
NAME	CONNECTOR	TYPE	DB	TABLE	QUERY	PATH	CREATED AT
FTP_DATASOURCE_RAW_TESTU	FTP_CONNECTOR_TESTU	FTP				titanic_train.csv	02-10-2019 05:28
SFTP_DATASOURCE_RAW_TESTU	SFTP_CONNECTOR_TESTU	SFTP				/upload/titanic_train.csv	02-10-2019 05:28
S3_DATASOURCE_RAW_TESTU	S3_CONNECTOR_TESTU	S3				titanic.csv	02-10-2019 05:28
HBASE_DATASOURCE_RAW_TESTU	HBASE_CONNECTOR_TESTU	HBASE		regression_house_ok			02-10-2019 05:28
HIVE_DATASOURCE_REQUEST_TESTU	HIVE_CONNECTOR_TESTU	HIVE	default		✓		02-10-2019 05:28
HIVE_DATASOURCE_RAW_TESTU	HIVE_CONNECTOR_TESTU	HIVE	default	regression_house_20			02-10-2019 05:28
SQL_DATASOURCE_REQUEST_TESTU	SQL_CONNECTOR_TESTU	SQL	prevision		✓		02-10-2019 05:28
SQL_DATASOURCE_RAW_TESTU	SQL_CONNECTOR_TESTU	SQL	prevision	titanic			02-10-2019 05:28

rows per page 25 ◂ prev 1 - 8 of 8 next ▸

This table is completed by 3 possible actions :

- Test a Data Source
- Edit a Data Source
- Remove a Data Source

To create a Data Source, simply click on the « New Data Source » button.

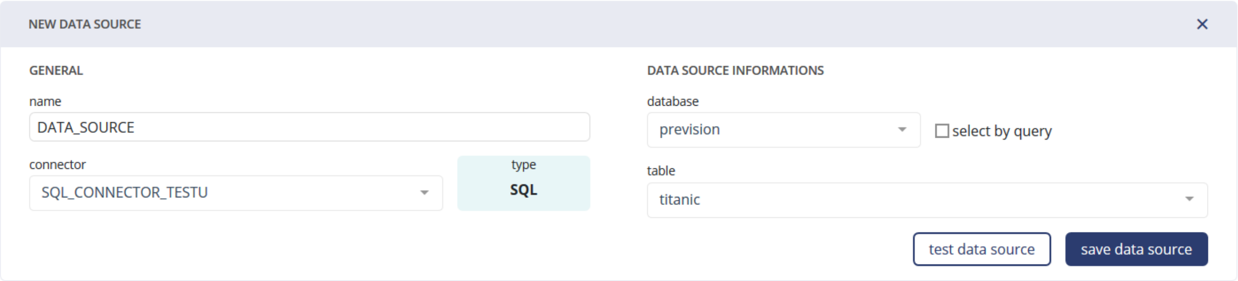


name:

connector:

type: —

For Data Sources from database connectors, it is possible to request via drop-down lists the database and table of your choice



name: DATA_SOURCE

connector: SQL_CONNECTOR_TESTU

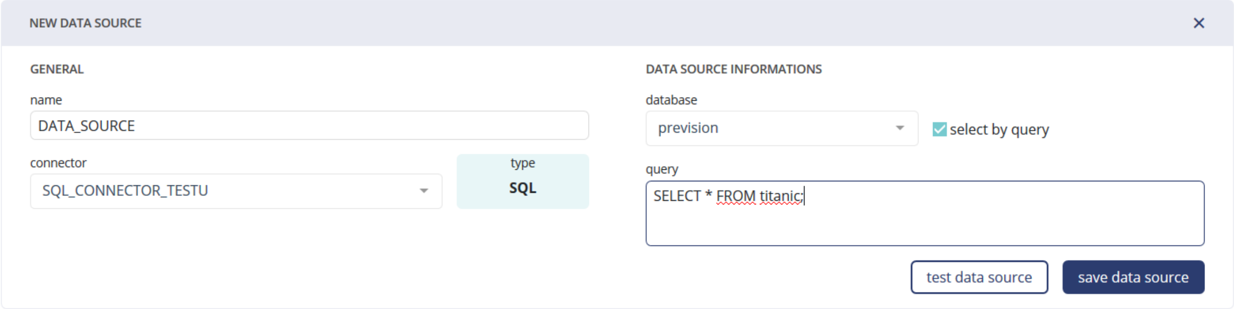
type: SQL

database: prevision

□ select by query

table: titanic

For Data Sources from SQL and HIVE connectors, it is also possible to choose a database and write a SQL query if you require more modularity :



NEW DATA SOURCE ✕

GENERAL

name: DATA_SOURCE

connector: SQL_CONNECTOR_TESTU

type: SQL

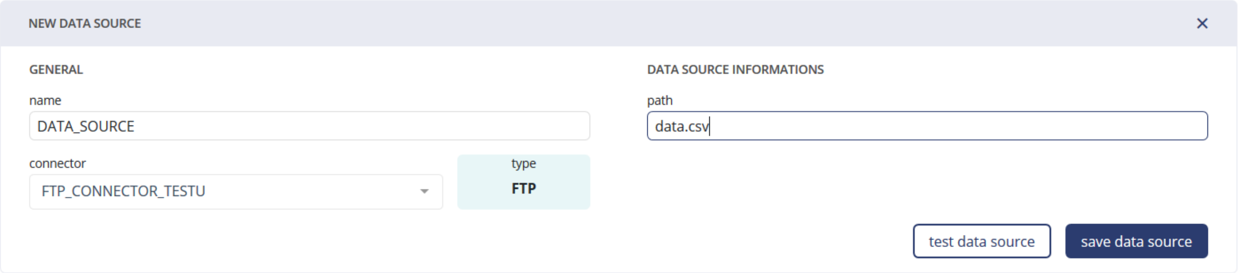
DATA SOURCE INFORMATIONS

database: prevision

☑ select by query

query: SELECT * FROM titanic;

For Data Sources from FTP connectors, a file path must be filled in



NEW DATA SOURCE ✕

GENERAL

name: DATA_SOURCE

connector: FTP_CONNECTOR_TESTU

type: FTP

DATA SOURCE INFORMATIONS

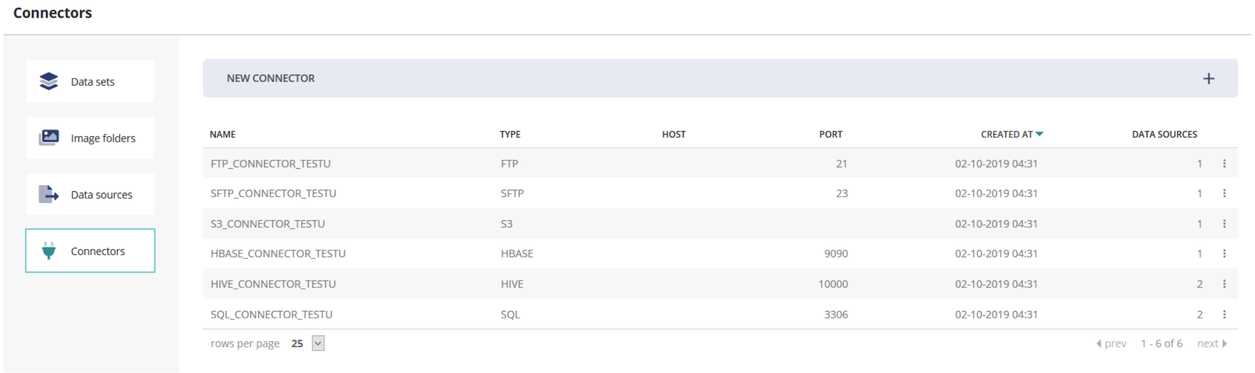
path: data.csv

2.5 Connectors

Connectors are pointers to persistent data sources. Several types of connectors are currently supported by Prevision. To know :

- SQL
- HIVE
- HBASE
- FTP
- SFTP
- S3

All connectors are presented in a table with their main characteristics (please note that host have been remove in the following screenshot)



Connectors

NAME	TYPE	HOST	PORT	CREATED AT	DATA SOURCES
FTP_CONNECTOR_TESTU	FTP		21	02-10-2019 04:31	1
SFTP_CONNECTOR_TESTU	SFTP		23	02-10-2019 04:31	1
S3_CONNECTOR_TESTU	S3			02-10-2019 04:31	1
HBASE_CONNECTOR_TESTU	HBASE		9090	02-10-2019 04:31	1
HIVE_CONNECTOR_TESTU	HIVE		10000	02-10-2019 04:31	2
SQL_CONNECTOR_TESTU	SQL		3306	02-10-2019 04:31	2

rows per page 25 · prev 1 - 6 of 6 next

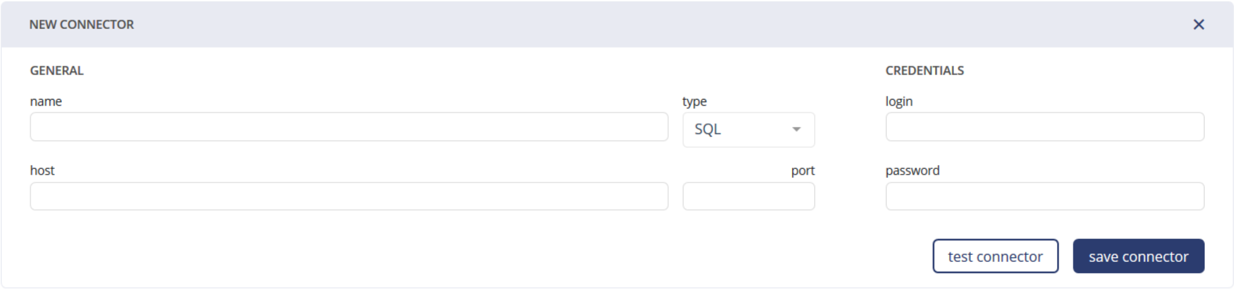
This table is completed by 3 possible actions :

- Test the connector
- Edit the connector
- Remove the connector

To create a connector, simply click on the « New Connector » button.

Connection to databases such as SQL - HIVE - HBASE - HBASE - FTP - SFTP is possible with the following information :

- Name : The name of your connector
- Type : The desired connector type
- Host : The URL to your environment
- Port : The port to your environment
- Login : The login allowing you to connect to your environment
- Password : The password to connect to your environment



NEW CONNECTOR

GENERAL

name

type: SQL

host

port

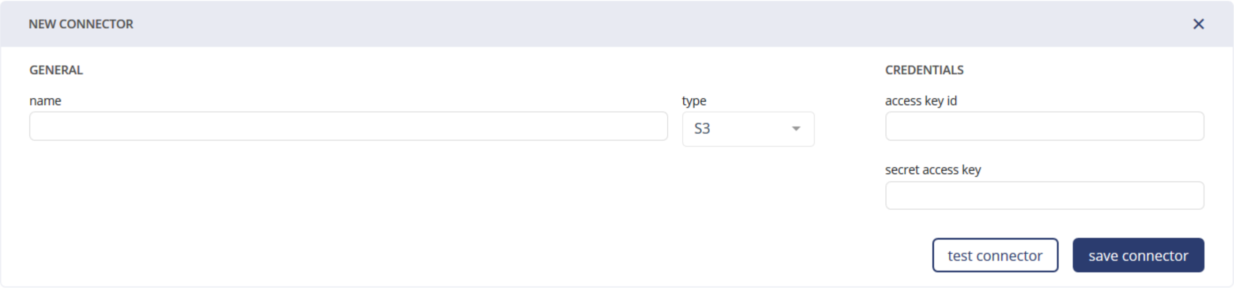
CREDENTIALS

login

password

Connection to data stored on Amazon S3 is possible with the following information :

- Login : Your Access_Key_ID
- Password : Your Access_Key



NEW CONNECTOR

GENERAL

name

type: S3

CREDENTIALS

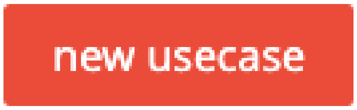
access key id

secret access key

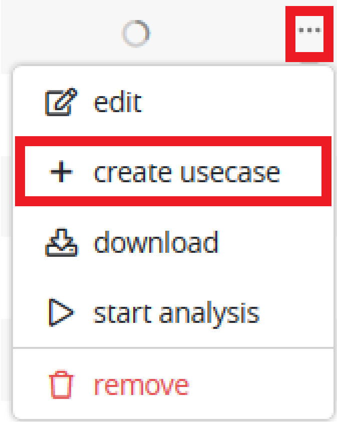
CHAPITRE 3

New usecase

In order to create a new usecase, from the usecase tab, you need to click on the « new usecase » button :



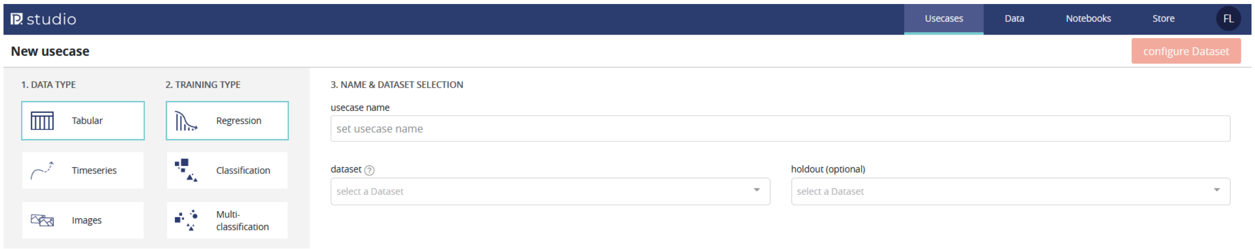
You can also create directly a usecase by clicking the . . . icon near a given Data set in the data screen :



When creating a new usecase, you should first specify a `DATA TYPE` among :

- Tabular (including textual)
- Time series
- Images

Then, you can specify a usecase name linked to a previously created Data Set.



New usecase

1. DATA TYPE

Tabular · Timeseries · Images

2. TRAINING TYPE

Regression · Classification · Multi-classification

3. NAME & DATASET SELECTION

usecase name: set usecase name

dataset: select a Dataset

holdout (optional): select a Dataset

Depending on the TRAINING TYPE, some options are displayed :

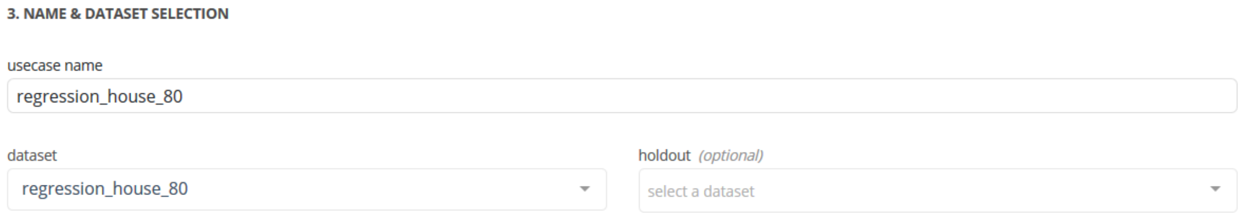
- Hold out : only for Tabular usecases. It is a Data Set that will be predicted for each model trained and the performance will be compute on it
- Image folder : only for Images usecases. It is a Data Set labelled as a folder containing images linked to a tabular Data Set

We offer 4 differents TRAINING TYPE :

TYPE	TABULAR	TIME-SERIES	IMAGE	DEFINITION	EXAMPLE
Regression	OK	OK	OK	Prediction of a quantitative feature	2.39 / 3.98 / 18.39
Classification	OK		OK	Prediction of a binary quantitative feature	« Yes » / « No » ou 0 / 1
Multi classification	OK		OK	Prediction of a qualitative feature whose cardinality is > 2	« Victory » / « Defeat » / « Tie game »
Object detection			OK	Detection from 1 to n objects per image + location	Is there a train on this image ? If so, where ?

3.1 Tabular

The screens for these 3 types of usecases are extremely similar. Only metrics, detailed below, change according to the type of project. First, you should give your usecase a name and attach a previously created Data Set :



3. NAME & DATASET SELECTION

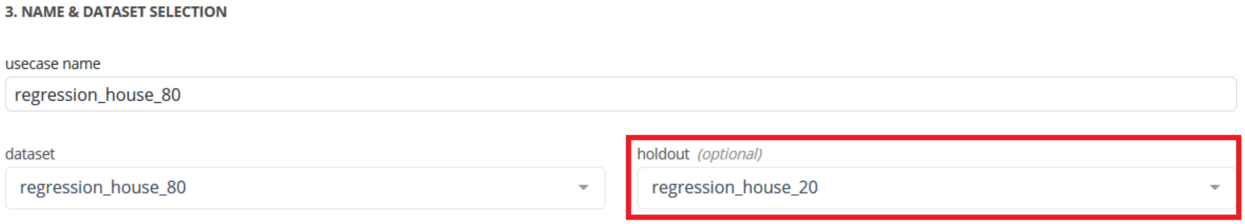
usecase name: regression_house_80

dataset: regression_house_80

holdout *(optional)*: select a dataset

Note that only tabular Data Sets with an OK parsed status (✓ icon on the Data Set screen on the PARSED column) are selected.

It is also possible, but not mandatory, to add a Data Set for comparison (hold out) :



3. NAME & DATASET SELECTION

usecase name: regression_house_80

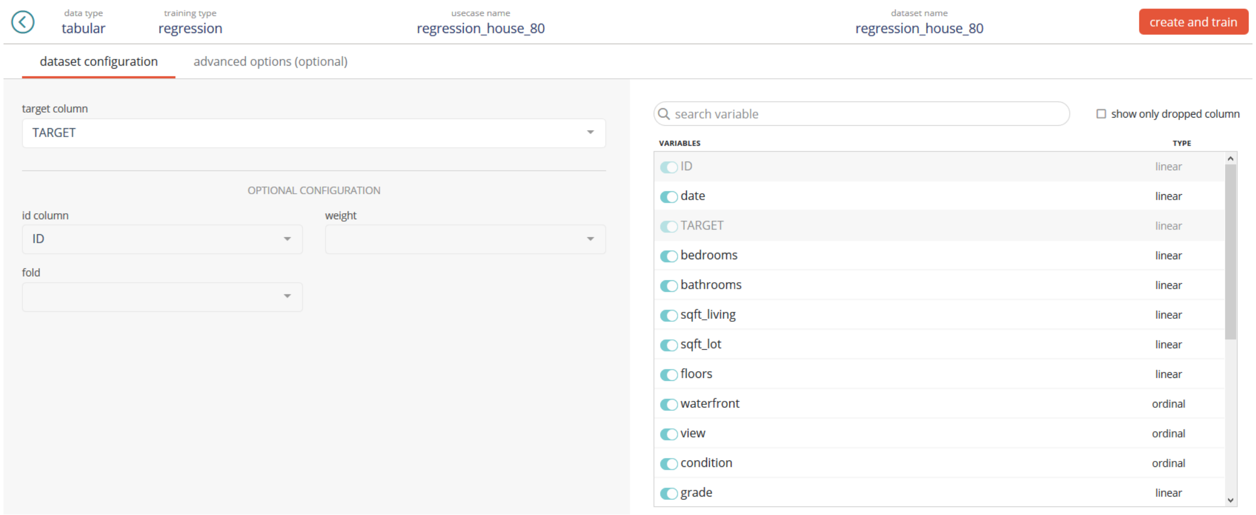
dataset: regression_house_80

holdout *(optional)*: regression_house_20

Typically, the addition of such a Data Set is useful in a study context in which we want to compare the quality of the actual prediction (and no longer only the performance estimators) on a set of models. This Data Set must have the same structure as the original set (same column name).

Once this step done you can proceed on cliquing the `configure dataset` button, located on the top right of the screen :

3.1.1 Data Set configuration



data type: tabular
training type: regression
usecase name: regression_house_80
dataset name: regression_house_80

target column: TARGET

id column: ID

weight:

fold:

search variable

□ show only dropped column

VARIABLES	TYPE
ID	linear
date	linear
TARGET	linear
bedrooms	linear
bathrooms	linear
sqft_living	linear
sqft_lot	linear
floors	linear
waterfront	ordinal
view	ordinal
condition	ordinal
grade	linear

On the left part of the screen, you will be able to fill :

- The target column (mandatory). This column is the one we want to predict on.
- The id column (optionnal). This column has typically no predictive power and is used to make join on other Data Sets later on.
- The fold (optionnal). Typically, this column will contain a feature of 1, 2, . . . n (n being the maximum number of folds). If fed, the CV stratification will be based on this column and won't be stratified to the target which is Prevision.io's default behavior.
- The weight (optionnal). Typically, this column contains a linear feature indicating the importance of a given row. The higher the weight, the more important the row is. If not fed, all rows are considered equally important (which is the case in most usecases).

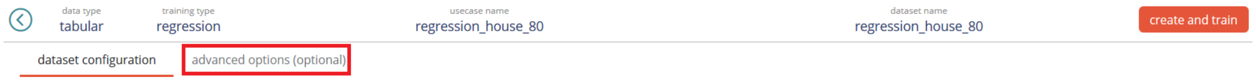
Note : If your Data Set contains a column named `ID` or `TARGET`, these will automatically be detected and selected from the corresponding menus

On the right part of the screen, you will be able to :

- Filter columns by names
- Shows only dropped (removed) columns
- Drop (remove) columns for the training phase. This means that every dropped column won't be use in the learning process

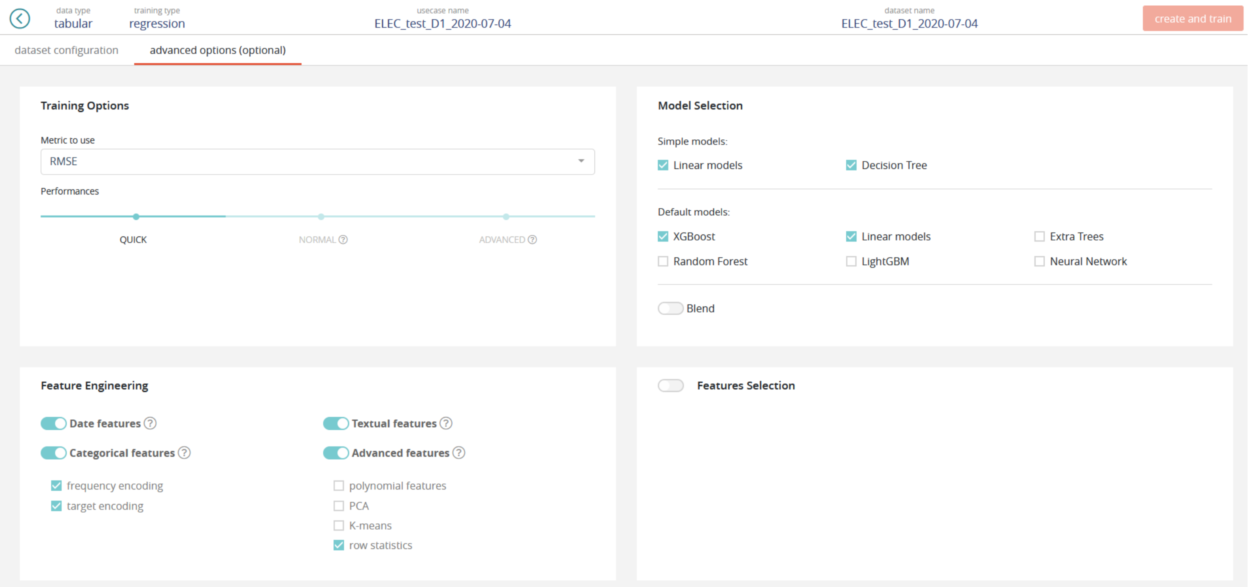
Once done, you can launch the training by clicking on the `create and train` button, located on the top right of the screen :

Optionnally, there are advanced options reachable by clicking the tab in the top bar :



data type: tabular
training type: regression
usecase name: regression_house_80
dataset name: regression_house_80

3.1.2 Advanced options



data type: tabular
training type: regression
usecase name: ELEC_test_D1_2020-07-04
dataset name: ELEC_test_D1_2020-07-04

Training Options

Metric to use: RMSE

Performances: QUICK / NORMAL / ADVANCED

Model Selection

Simple models:

- ☑ Linear models
- ☑ Decision Tree

Default models:

- ☑ XGBoost
- ☑ Linear models
- □ Extra Trees
- □ Random Forest
- □ LightGBM
- □ Neural Network

Blend

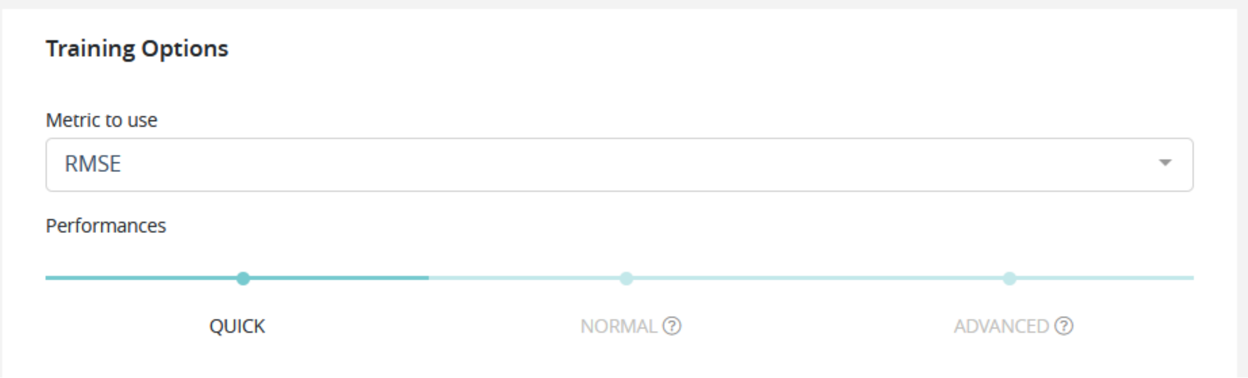
Feature Engineering

- Date features
- Textual features
- Categorical features
- Advanced features

- ☑ frequency encoding
- ☑ target encoding
- □ polynomial features
- □ PCA
- □ K-means
- ☑ row statistics

Features Selection

Training options



Training Options

Metric to use: RMSE

Performances: QUICK / NORMAL / ADVANCED

In this part of the screen, you can tune the following :

Metric (will differ depending of the training type) :

TYPE	METRIC	DEFINITION	DEFAULT ?
Regression	RMSE	Root mean squared error	YES
Regression	MSE	Mean squared error	
Regression	RMSLE	Root mean squared logarithmic error	
Regression	RMSPE	Root mean squared percentage error	
Regression	MAE	Mean absolute error	
Regression	MAPE	Mean absolute percentage error	

Suite sur la page suivante

Tableau 1 – suite de la page précédente

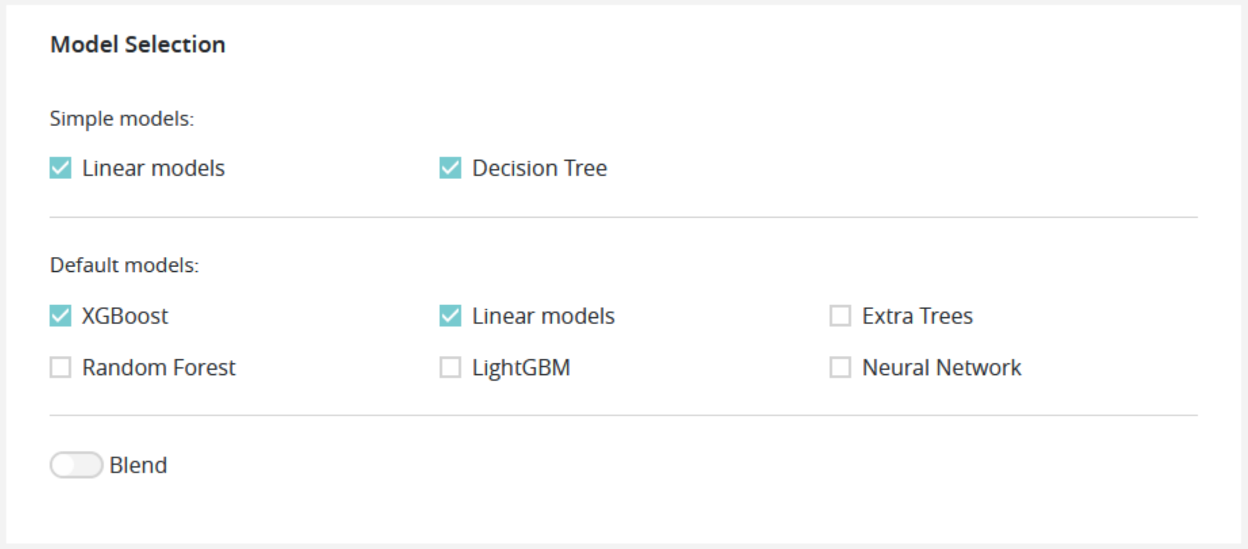
Regression	MER	Median absolute error	
Regression	R2	Coefficient of determination	
Regression	SMAPE	Symetric mean absolute percentage error	
Classification	AUC	Area under ROC curve	YES
Classification	ERROR RATE	Error rate	
Classification	LOGLOSS	Logarithmic loss	
Classification	ACCURACY	Accuracy	
Classification	F05	F-0.5 Score	
Classification	F1	F-1 Score	
Classification	F2	F-2 Score	
Classification	F3	F-3 Score	
Classification	F4	F-4 Score	
Classification	MCC	Matthews" correlation coefficient	
Classification	GINI	Gini's coefficient	
Classification	AUPCR	Area under precision-recall curve	
Classification	LIFT_AT_0.1	Lift @ 10%	
Classification	LIFT_AT_0.2	Lift @ 20%	
Classification	LIFT_AT_0.3	Lift @ 30%	
Classification	LIFT_AT_0.4	Lift @ 40%	
Classification	LIFT_AT_0.5	Lift @ 50%	
Classification	LIFT_AT_0.6	Lift @ 60%	
Classification	LIFT_AT_0.7	Lift @ 70%	
Classification	LIFT_AT_0.8	Lift @ 80%	
Classification	LIFT_AT_0.9	Lift @ 90%	
Multi classification	LOGLOSS	Logarithmic loss	YES
Multi classification	ERROR_RATE	Error rate	
Multi classification	AUC	Area under ROC cure (mean of AUC by class)	
Multi classification	MACROF1	Macro F1-Score (mean of F1 by class)	
Multi classification	ACCURACY	Accuracy	
Multi classification	QKAPPA	Quadratic weighted Kappa	
Multi classification	MAP_AT_3	Mean average precision @ 3	
Multi classification	MAP_AT_5	Mean average precision @ 5	
Multi classification	MAP_AT_10	Mean average precision @ 10	

All technicals formulas are available here : https://previsionio.readthedocs.io/fr/latest/_static/ressources/formula.pdf

Performances :

- QUICK : Training is done faster but performance may be slightly lower. Ideal in iterative phase.
- NORMAL : Intermediate value, suitable for most usecases on a later stage.
- ADVANCED : The training is done in an optimal way. Though the performance will be more stable, the calculations will take longer to process. This is ideal when the model is put into production and the performance is discriminating.

Model Selection



Model Selection

Simple models:

☑ Linear models ☑ Decision Tree

Default models:

☑ XGBoost ☑ Linear models ☐ Extra Trees

☐ Random Forest ☐ LightGBM ☐ Neural Network

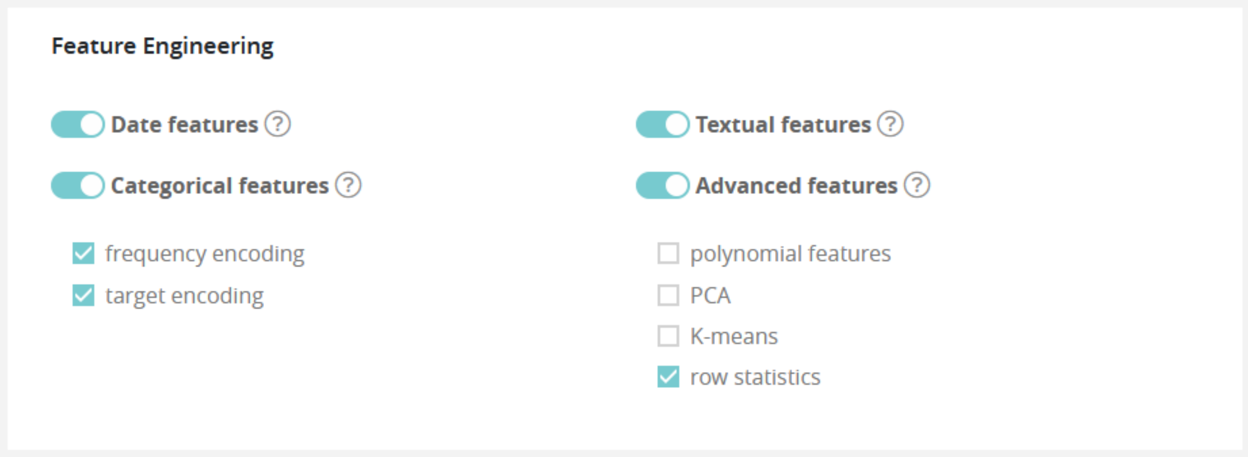
○ Blend

In this part of the screen you can enable or disable model types, such as :

- Simple models (with simple feature engineering)
  - Linear models (https://en.wikipedia.org/wiki/Linear_regression or https://en.wikipedia.org/wiki/Logistic_regression)
  - Decition tree (https://en.wikipedia.org/wiki/Decision_tree_learning)
- Default models (with advanced feature engineering)
  - XGBoost (https://github.com/dmlc/xgboost)
  - Linear models (https://en.wikipedia.org/wiki/Linear_regression or https://en.wikipedia.org/wiki/Logistic_regression)
  - Random Forest (https://en.wikipedia.org/wiki/Random_forest)
  - LightGBM (https://github.com/Microsoft/LightGBM)
  - Extra Trees (https://en.wikipedia.org/wiki/Random_forest#ExtraTrees)
  - Neural Network (https://en.wikipedia.org/wiki/Artificial_neural_network)
- Blend (https://mlwave.com/kaggle-ensembling-guide/)

Note : The more model types you add in the training, the longer it will be.

Feature Engineering



Feature Engineering

● Date features ? ● Textual features ?

● Categorical features ? ● Advanced features ?

☑ frequency encoding

☑ target encoding

☐ polynomial features

☐ PCA

☐ K-means

☑ row statistics

In this part of the screen you can enable or disable feature engineering, such as :

- Date features : dates are detected and operations such as information extraction (day, month, year, day of the week, etc.) and differences (if at least 2 dates are present) are automatically performed
- Textual features : Textual features : textual features are detected and automatically converted into numbers using 3 techniques :

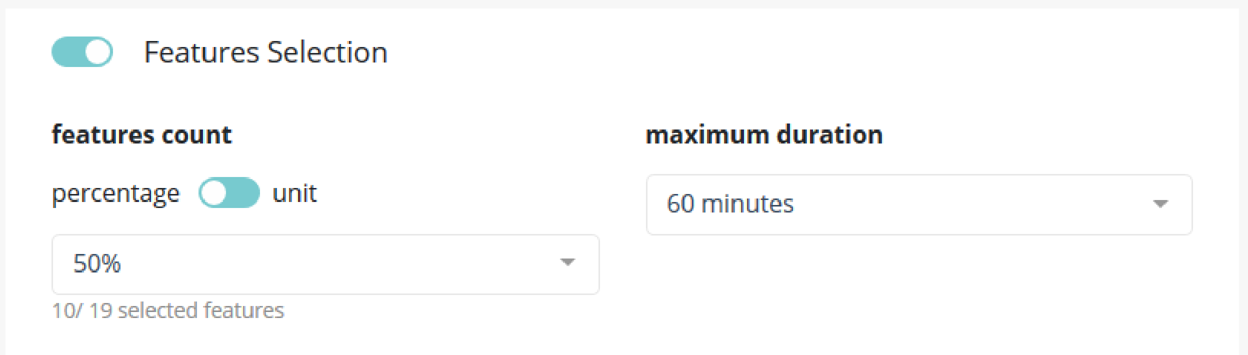
By default, only TF-IDF approach is used.

Note : For better performance, it is advisable to check the word embedding and sentence embedding options. Checking its additional options will increase the time required for feature engineering, modeling, and prediction

- Categorical features :
  - Frequency encoding : modalities are converted to their respective frequencies
  - Target encoding : modalities are replaced by the average (TARGET, grouped by modality) for a regression and by the proportion of the modality for the target's modalities in the context of a classification
- Advanced features :
  - Polynomial features : features based on products of existing features are created. This can greatly help linear models since they do not naturally take interactions into account but are less usefull on tree based models
  - PCA : main components of the PCA
  - K-means : Cluster number comming from a K-means methode are added as new features
  - Row statistics : features based on row by row counts are added as new features (number of 0, number of missing values, . . .)

Note : The more feature engineering you add in the training, the longer it will be.

Feature Selection



Features Selection

features count

percentage unit

50%

10/ 19 selected features

maximum duration

60 minutes

In this part of the screen you can chose to enable feature selection (off by default).

This operation is important when you have a high number of features (a couple hundreds) and can be critical when the number of features is above 1000 since the full Data Set won't be able to hold in RAM.

You can chose to keep a percentage or a count of feature and you can give a time budget to Prevision.io's to perform the search of optimal features given the TARGET and all other parameters. In this time, Prevision.io will subset the feature of the Data Set then start the classical process.

3.2 Time series

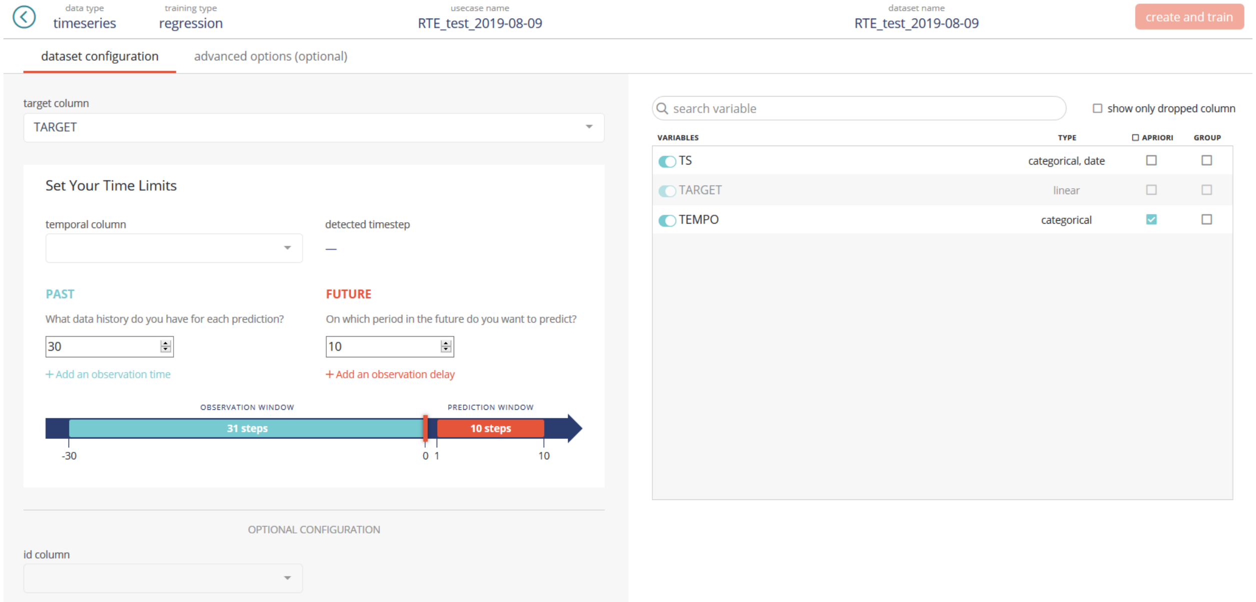
Time series is very similar to tabular usecase except :

- There is no hold out

- There is no weight
- There is no fold (in this case, Prevision.io use temporal stratification)

However, you will find some new notions :

- Temporal column : the feature that contain the time reference of the time series. Since date formats can be complex, Prevision.io supports ISO 8601 (https://fr.wikipedia. org/wiki/ISO_8601) as well as standard formats (e.g. DD/MM/YYYY or DD-MM-YYYY hh :mm).
- Time step : period between 2 events (within the same group) from the temporal column (automatically detected)
- Observation window : illustrate the period in the past that you have for each prediction
  - Start of observation window : the maximum time step multiple in the past that you'll have data from for each prediction (inclusive, 30 by default)
  - Enf of the observation window : the last time step multiple in the past that you'll have data from for each prediction (inclusive, 0 by default that means that the immediate values before the prediction time step is known)
- Prediction window : illustrate the period in the future that you want to predict
  - Start of the prediction window : the first time step multiple you want to predict (inclusive, 1 by default which means we will predict starting at the next value)
  - End of the prediction window : the last time stemp multiple you want to predict (inclusive, 10 by default which means we will predict up to the 10th next value)
- A priori features : features whose value is known in the future (customer number, calendar, public holidays, weather. . .)
- Group features : features that identify a unique time serie (e.g. you want to predict your sales by store and by product. If you have 2 stores selling 3 products, there are 6 time series in your file. Selecting features « store » and « product in the group column allows Prevision.io to take into account these multiple series)



data type: timeseries
training type: regression
usecase name: RTE_test_2019-08-09
dataset name: RTE_test_2019-08-09

target column: TARGET

Set Your Time Limits

temporal column:

detected timestep: —

PAST

What data history do you have for each prediction? 30

+ Add an observation time

FUTURE

On which period in the future do you want to predict? 10

+ Add an observation delay

OPTIONAL CONFIGURATION

id column:

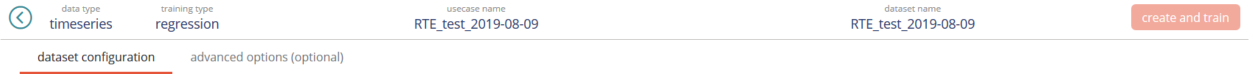
search variable

☐ show only dropped column

VARIABLES	TYPE	☐ APRIORI	GROUP
TS	categorical, date	☐	☐
TARGET	linear	☐	☐
TEMPO	categorical	☑	☐

Once eveything set up, you can launch the training by clicking on the « create and train » button, located on the top right of the screen :

Optionnally, there are advanced options reachable by clicking the tab in the top bar :



data type: timeseries
training type: regression
usecase name: RTE_test_2019-08-09
dataset name: RTE_test_2019-08-09

Example 1 : You want to predict day ahead value per hour and you have all data available 1 week in the past for each value

Time step = 1 hour

Start of observation window = 7 (days) * 24 (hours / day) - 1 (because this value is inclusive) = 167

End of observation window = 0 (we have the last known value before each prediction)

Start of prediction window = 1 (we predict the next immediate value)

End of prediction window = 1 (day) * 24 (hours) (we predict the next day, on a hour level)

Example 2 : You want to predict from day+2 to day+7 (= week ahead minus the first day) per day and you have all data available 4 weeks in the past for each value with a 1 week delay (which means you don't know the last week value)

Time step = 1 day

Start of observation window = 4 (weeks) * 7 (days / week) - 1 (because this value is inclusive) = 27

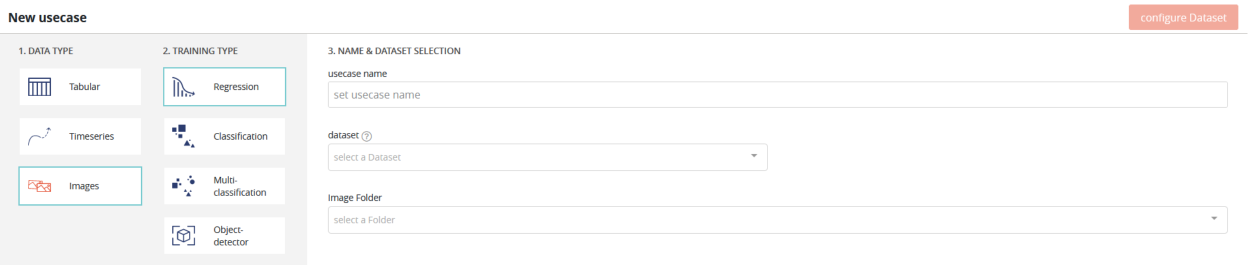
End of observation window = 1 (week) * 7 (days / week) = 7 (we miss the last known week)

Start of prediction window = 2 (we predict the second immediat value)

End of prediction window = 7 (we predict up to the next 7th day)

Notes : The wider the window is, the longer the compute time will be. Also, please make sure to provide an observation window of reasonnable size. It most usecases, it should be a reasonnable multiple of the prediction window. (e.g. if you predict day ahead, don't use more that a couple of weeks in the observation window).

3.3 Images



3.3.1 Regression / classification / multi classification

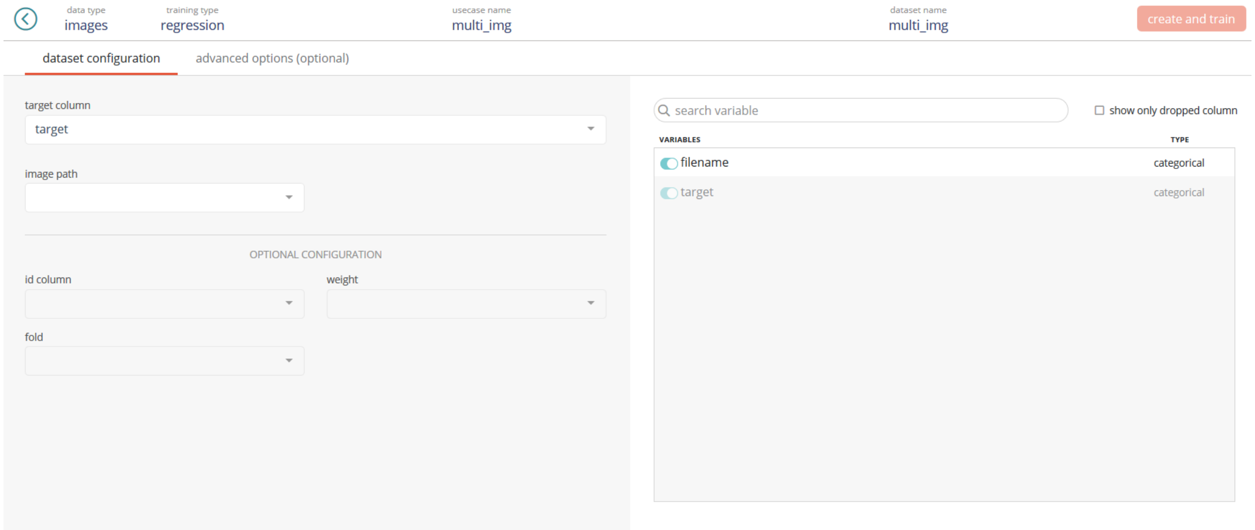
To launch a regression / classification / multiclass classification project, the method is identical to tabular usecases with the exception that you need to :

- Add in the tabular Data Set a relative path to the image, which will be specified in the interface.
- Provide an image type Data Set whose paths correspond to those indicated in the previous Data Set.

It should be noted that the tabular Data Set may or may not contain exogenous features (e.g. geographical position of the camera, temperature, weather, etc.)

Once this step done you can proceed on cliquing the `configure dataset` button, located on the top right of the screen :

Data Set configuration



data type: images
training type: regression
usecase name: multi_img
dataset name: multi_img

target column: target
image path:

OPTIONAL CONFIGURATION

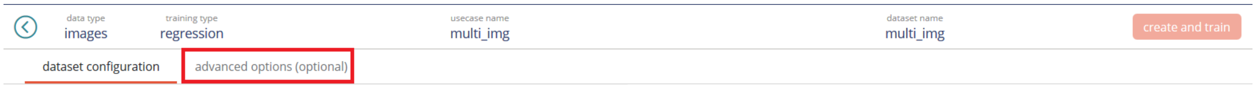
id column:
weight:
fold:

VARIABLES	TYPE
filename	categorical
target	categorical

On the left part of the screen, you will be able to fill the same columns than in tabular usecase but you'll need to add the « image path » feature which link the tabular Data Set and the images folder.

Once done, you can launch the training by clicking on the « create and train » button, located on the top right of the screen :

Optionnally, there are advanced options reachable by clicking the tab in the top bar :

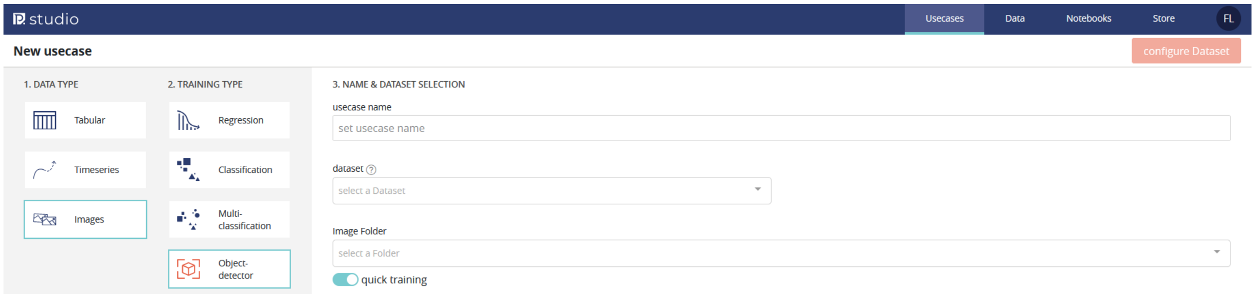


data type: images
training type: regression
usecase name: multi_img
dataset name: multi_img

Advanced options

Advanced options do work exactly like for tabular usecases.

3.3.2 Object detection



New usecase

1. DATA TYPE: Images
2. TRAINING TYPE: Object-detector
3. NAME & DATASET SELECTION

usecase name: set usecase name
dataset: select a Dataset
Image Folder: select a Folder
quick training

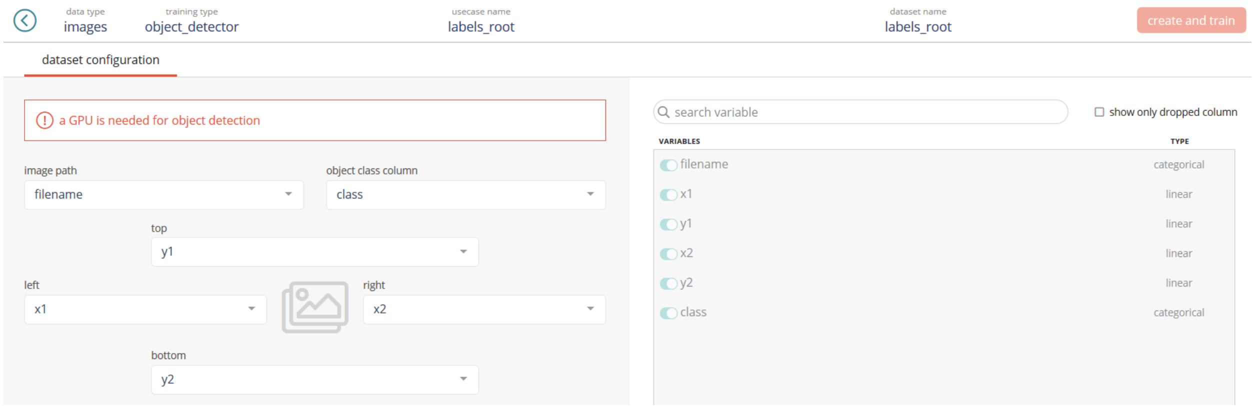
Like any other images usecase, you need to specify 2 Data Sets (one tabular and one images).

There is a « quick » button that will allow to train a model faster (typically by a factor 5-10) with a little bit less of performance.

Note : While object detection use case can run on CPU's, the training time will be very long. That's why we recommand you to have a instance that has GPU in it.

Once this step done you can proceed on cliquing the « configure dataset » button, located on the top right of the screen :

Data Set configuration



data type: images
training type: object_detector
usecase name: labels_root
dataset name: labels_root

a GPU is needed for object detection

image path: filename
object class column: class
top: y1
left: x1
right: x2
bottom: y2

search variable

☐ show only dropped column

VARIABLES	TYPE
filename	categorical
x1	linear
y1	linear
x2	linear
y2	linear
class	categorical

In this usecase type, you'll need to provide :

- image path : the feature that link the tabular Data Set to the image folder
- object class column : the feature that indicates the category of the object to detect
- top : the top ordinate of the pixel that indicates the bounding boxe in which the object is
- right : the right abscissa of the pixel that indicates the bounding boxe in which the object is
- bottom : the bottom ordinate of the pixel that indicates the bounding boxe in which the object is
- left : the left abscissa of the pixel that indicates the bounding boxe in which the object is

Note : The Data Set shouldn't contains any other columns than the one required to launche the training

Once done, you can launch the training by clicking on the « create and train » button, located on the top right of the screen :

CHAPITRE 4

Usecase versioning

4.1 How to keep track of your experiments ?

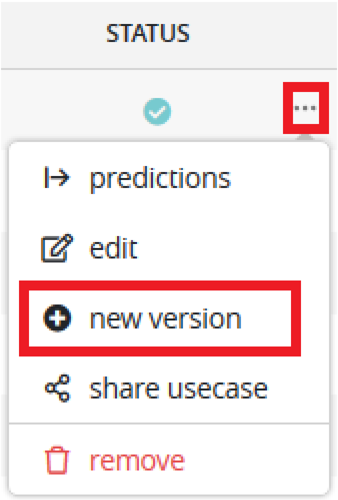
4.1.1 Creating versions

From version 10.1 onwards, it is possible to create multiple versions of a usecase, preserving settings while possibly modifying them.

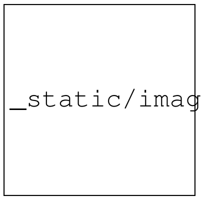
This allows to easily try different settings, approaches and datasets for a given problem, and keep track of the results of the experiments in a single location.

You can create a new version of a usecase from three different locations :

— the main usecase listing : click on the `...` on the usecase for which you want to create a new version, then `New version` :

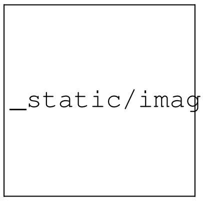


— the usecase « general » page : on the bottom, the « New version » will take you to the usecase creation screen :



_static/images/new_version_general.png

— the « versions » page : here you will see all versions of a usecase and you will also be able to create a new one :

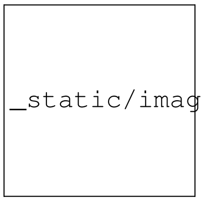


_static/images/new_version_infos_versions.png

When a new version is created, you will be taken to the usecase creation screen where all settings will be exactly as you set them up for the first usecase.

You can modify all of the settings : problem type, dataset, model settings, etc, although it is necessary to keep in mind that some previous might not be applicable anymore once you change the problem type (as an example, if the first usecase was an image classification usecase, changing the problem type to a tabular classification will mean that the « image folder » setting will not be applicable). It is up to you to keep your experiments consistent or to create a new usecase when it is needed.

If your experiments deviate too far from the original goal, you can create a completely new usecase while still preserving some settings by using the « duplicate usecase » button on a usecase « general » tab :

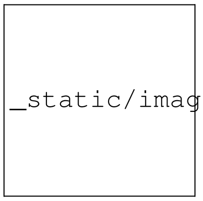


_static/images/duplicate_usecase.png

Doing will take you to the usecase creation screen, but the new created usecase will be independent of the previous one, resetting the version history.

4.1.2 Versioning info

From every version of a usecase, you can access the listing of all the versions that were created in the « versions » tab.



_static/images/listing_versions.png

Here you can compare versions scores and create new versions from each of the existing iterations.

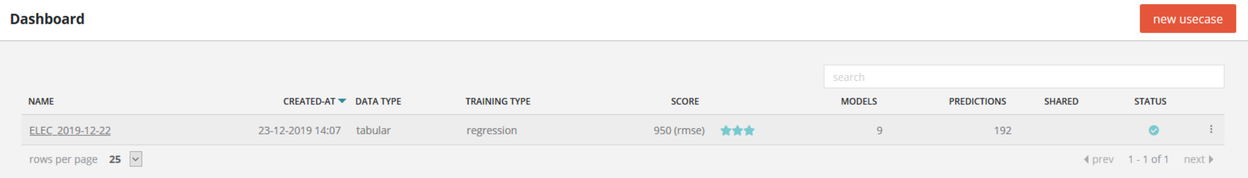
Note : When you create a new version from an existing version, the setting from this version will be applied in the usecase creation screen. However, the version number will always be set to increment the highest existing version number. For example, on a usecase with 4 existing versions, you want to create a new version using version nº3 (because, for example, the 4th experiment was a mistake). The new version, even if created from version nº3, will be nº5.

CHAPITRE 5

Use cases

5.1 Dashboard use cases

This dashboard allows you to see all usecases that you have created. You can access it by clicking on :



NAME	CREATED-AT	DATA TYPE	TRAINING TYPE	SCORE		MODELS	PREDICTIONS	SHARED	STATUS	
ELEC_2019-12-22	23-12-2019 14:07	tabular	regression	950 (rmse)	★★★	9	192		✓	⋮

You can create a new usecase by clicking on top right button or you can check all informations on previously done usecases in the table. This table is sortable and filterable by usecase name / usecase type. Also, names can be clicked and will redirect you on the detailed view of the selected usecase.

On each line of the table, you can see :

- The usecase name
- The creation date
- The data type (among « tabular », « time series », « images »)
- The training type (among « regression », « classification », « multiclassification », « object-detection »)
- The actual score, the optimized metric and a star system allowing you to quickly know if a model is well performing
- The number of trained models
- The number of predictions done
- The number of people you have share the use case with (none by default)
- The status of the usecase (running, done, crashed)
- Actions linked to the usecase :
  - Pause or resume a running usecase

- Share a use case
- Make predictions
- Stop or delete a usecase

5.1.1 Star system

By default, when no model is available, 3 gray stars will be displayed (and 0 blue). As soon as at least 1 model is available, the number of stars may change (up to 3 blue stars will be displayed according to the current performance of the modelisation).

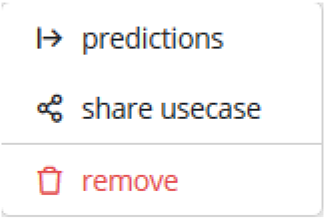
The detail of the computation is explained here :

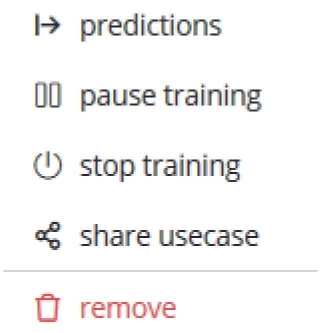
METRIC	3 STARS	2 STARS	1 STAR
MSE	[0 ; 0.01 * VAR[	[0.01 * VAR ; 0.1 * VAR[	[0.1 * VAR ; VAR[
RMSE	[0 ; 0.1 * STD[	[0.1 * STD ; 0.3 * STD[	[0.3 * STD ; STD[
MAE	[0 ; 0.1 * STD[	[0.1 * STD ; 0.3 * STD[	[0.3 * STD ; STD[
MAPE	[0 ; 10[	[10 ; 30[	[30 ; 1[
RMSLE	[0 ; 0.095[	[0.095 ; 0.262[	[0.262 ; 1[
RMSPE	[0 ; 0.1[	[0.1 ; 0.3[	[0.3 ; 1[
SMAPE	[0 ; 0.1[	[0.1 ; 0.3[	[0.3 ; 1[
MER	[0 ; 0.1[	[0.1 ; 0.3[	[0.3 ; 1[
R2	]0.9 ; 1]	]0.7 ; 0.9]	]0.5 ; 0.7]
AUC	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
LOGLOSS	[0 ; 0.223[	[0.223 ; 0.693[	[0.693 ; +inf[
ERROR RATE	[0 ; 0.125[	[0.125 ; 0.25[	[0.25 ; +inf[
mAP	[0 ; 45[	[45 ; 60[	[60 ; 100[
AUCPR	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
ACCURACY	]0.875 ; 1]	]0.75 ; 0.875]	]0.5 ; 0.75]
F1	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
MCC	]0.9 ; 1]	]0.7 ; 0.9]	]0.5 ; 0.7]
GINI	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
F05	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
F2	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
F3	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
F4	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
MACRO F1	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
MACRO AUC	]0.85 ; 1]	]0.65 ; 0.85]	]0.5 ; 0.65]
MACRO ACCURACY	]0.875 ; 1]	]0.75 ; 0.875]	]0.5 ; 0.75]
QUADRATIC KAPPA	]0.8 ; 1]	]0.6 ; 0.8]	]0.2 ; 0.6]
MAP @ k	]0.875 ; 1]	]0.75 ; 0.875]	]0.5 ; 0.75]
LIFT @ k	]1 + 7 * (1-k) ; 1]	]1 + 3 * (1-k) ; 1 + 7 * (1-k)]	]1 + 1 * (1-k) ; 1 + 3 * (1-k)]

VAR is the variance of the target feature STD is the standard deviation of the target feature

5.1.2 Actions links

Actions links are available when clicking the top right « … » icon on the usecase dashboard..





Actions will change depending of the usecase status (running, paused, finished).

Pausing a usecase

As long as a use case is running, you can pause it by clicking on the « pause » action link.

The pause isn't immediate. All running calculus are beeing finished before the actual pause which can take a couple of time depending on the data set size.

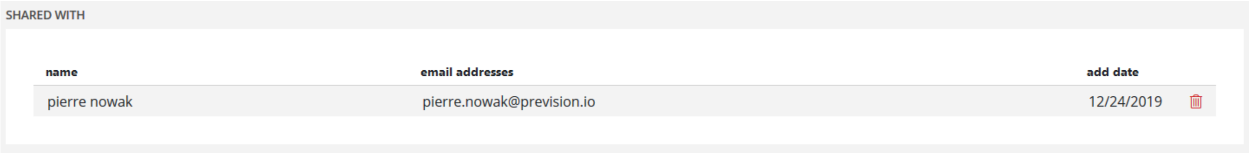
To resume a usecase, just click on the « resume » action link.

Sharing a usecase

You can share a usecase by clicking on the « share » action link.

By default, a usecase isn't shared. If you want to share it, just add an email of someone that has an account on your instance. He'll see the use case in his shared dashboard.

Once shared, you can see who has access :



SHARED WITH

name	email addresses	add date	
pierre nowak	pierre.nowak@prevision.io	12/24/2019	🗑

On the main dashboard, there will be a « 1 » displayed in the « shared » column, telling you that the usecase is currently shared with 1 user.

You can also stop sharing a usecase. To do so, remove the specific by clicking on the « trash » icon next to the desired user.

Making predictions

You can access the prediction tab by clicking on the « prediction » action link.

Stop the calculation

In the case where a usecase is running it is possible to stop the calculation by clicking on the « stop » action link.

Contrary to the pause, this action is final, and the calculation cannot be restarted.

5.1.3 Delete a usecase

In case a usecase is finished, it is possible to delete it by clicking on « remove » action link.

This action is irreversible.

CHAPITRE 6

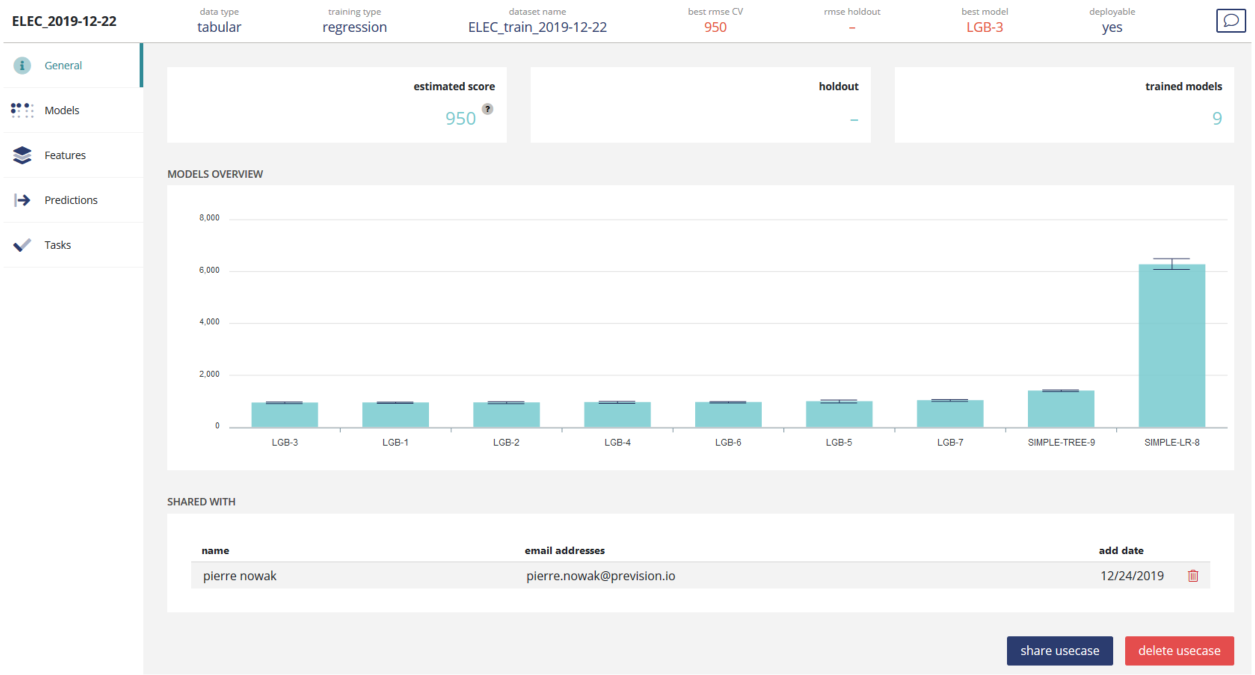
Supervised use cases

The « supervised » usecases include tabular or images use cases :

- Regression
- Classification
- Multi classification

Once a use case of this type is created, whether it is completed or still in process, you can view the performance of the model and data statistics by clicking on the use case name in the dashboard.

You will then be directed to the following screen :



ELEC_2019-12-22

data type	training type	dataset name	best rmse CV	rmse holdout	best model	deployable
tabular	regression	ELEC_train_2019-12-22	950	–	LGB-3	yes

estimated score	holdout	trained models
950	–	9

SHARED WITH

name	email addresses	add date
pierre nowak	pierre.nowak@prevision.io	12/24/2019

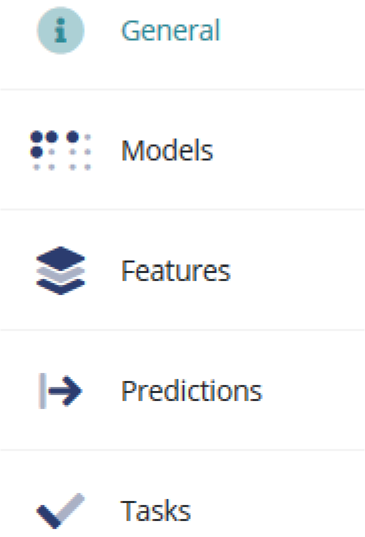
You can also stop or delete the current use cased, depending on his state by clicking on :

⏻

or

For each use case, you can navigate through different screens, namely :

- General
- Models
- Features
- Predictions
- Tasks



Note that it is possible to export most of the charts generated by Prevision.io in different formats (PNG, SVG). To do this, simply click on the icon below, located next to the chart in question :

≡

6.1 General

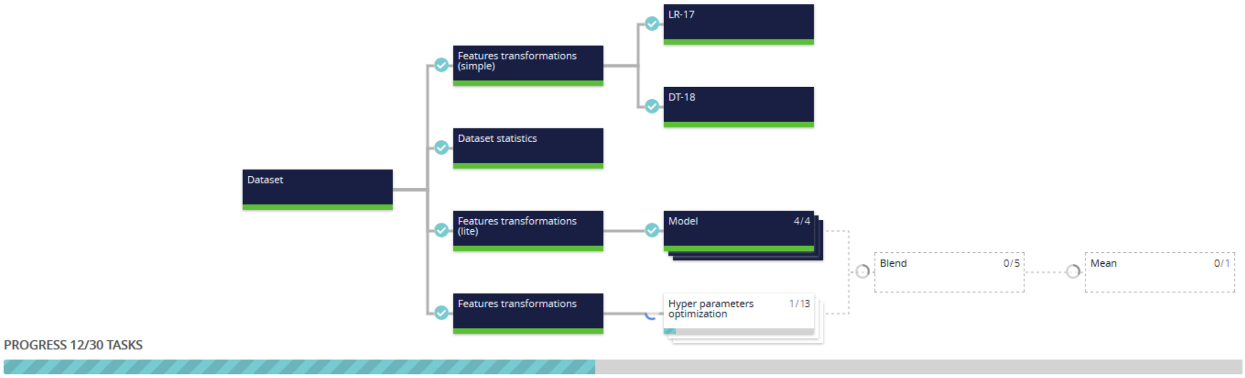
This screen allows you to view general information about a usecase.

On top of the screen, you will find a sum up of the use case :

- Data type
- Training type
- Name
- Best CV score
- Hold out score (if hold out is provided at start)
- Best model technology
- If the model can be deployed (ready to be put in the Store)
- Number of trained models

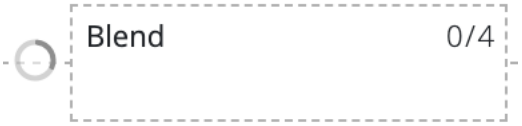
_static/images/ucsumup.png

While the usecase is running, you can monitor finished, current and upcoming tasks in the execution graph :

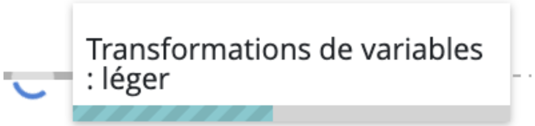


Tasks can be in any one of five states :

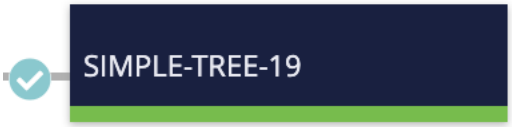
- Pending



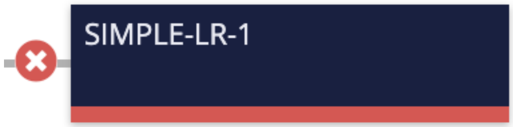
- Running



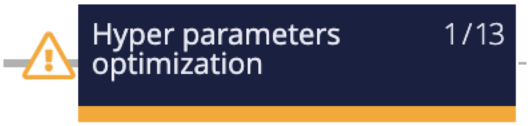
- Done



- Failed

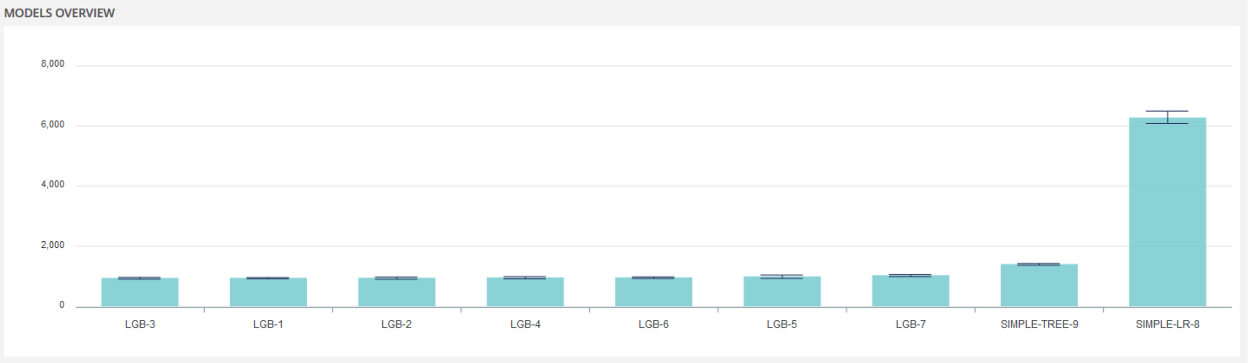


- Partially failed



If you want to remind you what type of experiment you have done, you can add a description to your use case by clicking the following button

In any case, as soon as a model is available, a learning plot will be displayed. It represents the evolution of the metric according to the number of models. This plot is either sorted decreasing or increasing depending of the metric. However, best model will be always first (ie on top left) Marks around bars represents a confidence interval of the performance estimator.

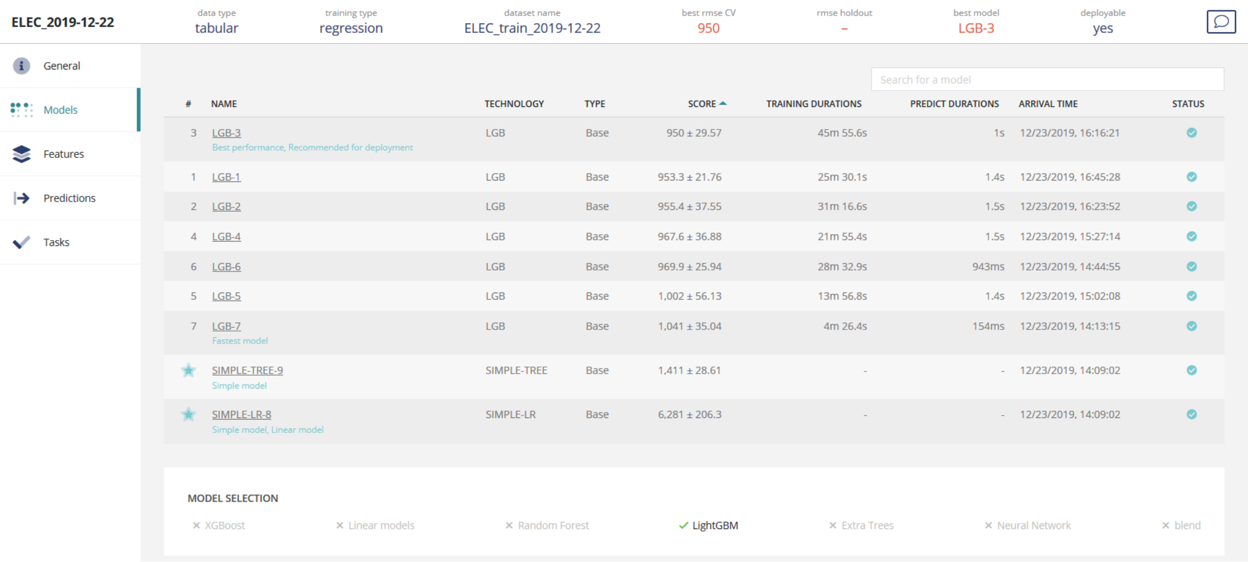


Each bar correspond to one model. Its technology will be written followed by an integer. Example : LGB-3 is a LightGBM type model, that was queued in the 3rd position.

Clicking on a given model will redirect you to the model analysis of the selected one.

6.2 Models

The model tab will display more information about models linked to the current usecase :



ELEC_2019-12-22	data type	training type	dataset name	best rmse CV	rmse holdout	best model	deployable
	tabular	regression	ELEC_train_2019-12-22	950	–	LGB-3	yes

#	NAME	TECHNOLOGY	TYPE	SCORE ▲	TRAINING DURATIONS	PREDICT DURATIONS	ARRIVAL TIME	STATUS
3	LGB-3 Best performance, Recommended for deployment	LGB	Base	950 ± 29.57	45m 55.6s	1s	12/23/2019, 16:16:21	✓
1	LGB-1	LGB	Base	953.3 ± 21.76	25m 30.1s	1.4s	12/23/2019, 16:45:28	✓
2	LGB-2	LGB	Base	955.4 ± 37.55	31m 16.6s	1.5s	12/23/2019, 16:23:52	✓
4	LGB-4	LGB	Base	967.6 ± 36.88	21m 55.4s	1.5s	12/23/2019, 15:27:14	✓
6	LGB-6	LGB	Base	969.9 ± 25.94	28m 32.9s	943ms	12/23/2019, 14:44:55	✓
5	LGB-5	LGB	Base	1,002 ± 56.13	13m 56.8s	1.4s	12/23/2019, 15:02:08	✓
7	LGB-7 Fastest model	LGB	Base	1,041 ± 35.04	4m 26.4s	154ms	12/23/2019, 14:13:15	✓
★	SIMPLE-TREE-9 Simple model	SIMPLE-TREE	Base	1,411 ± 28.61	-	-	12/23/2019, 14:09:02	✓
★	SIMPLE-LR-8 Simple model, Linear model	SIMPLE-LR	Base	6,281 ± 206.3	-	-	12/23/2019, 14:09:02	✓

MODEL SELECTION

✕ XGBoost ✕ Linear models ✕ Random Forest ✓ LightGBM ✕ Extra Trees ✕ Neural Network ✕ blend

You'll find a table of every planned and/or trained model for a given usecase and information related to them :

- Model number
- Model name
- Model technology

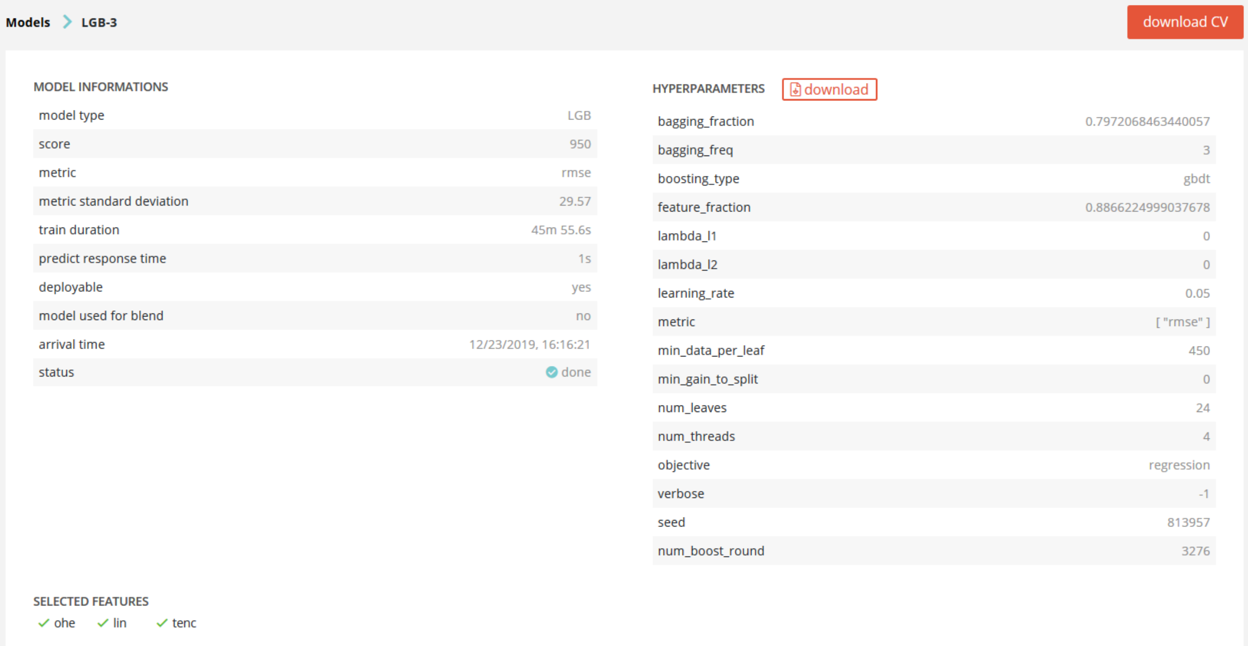
- Model type (among « base » for standard models, « blend » for a mix of standard models, « mean » for a mean of blends)
- Score (CV estimation +- standard deviation of CV score)
- Training durations (training time of the model)
- Predict durations (estimation of a unit prediction. / !time isn't linear with the number of predictions)
- Arrival time (time where the model has finished training)
- Status

It should be noted that some of the algorithms (sometimes the same) have badges, namely :

- `Best performance` : The one with the best predictive performance
- `Recommended for deployment` : The model with the best predictive performance that is not a blend
- `Fastest model` : The one that predicts with the lowest response time
- `Simple model` : A simple model that is visuaisable and can be exported in `SQL`, `Python` or `R`. Only Linear and Decition trees models can be tagged

Underneath, a table will list what kind of models you have selected during the usecase definition. Here, only LightGBM models have been selected by the user.

Clicking on a model will give you technical information about him and also performance analysis



MODEL INFORMATIONS

model type	LGB
score	950
metric	rmse
metric standard deviation	29.57
train duration	45m 55.6s
predict response time	1s
deployable	yes
model used for blend	no
arrival time	12/23/2019, 16:16:21
status	✓ done

HYPERPARAMETERS

bagging_fraction	0.7972068463440057
bagging_freq	3
boosting_type	gbdt
feature_fraction	0.8866224999037678
lambda_l1	0
lambda_l2	0
learning_rate	0.05
metric	["rmse"]
min_data_per_leaf	450
min_gain_to_split	0
num_leaves	24
num_threads	4
objective	regression
verbose	-1
seed	813957
num_boost_round	3276

SELECTED FEATURES

✓ ohe ✓ lin ✓ tenc

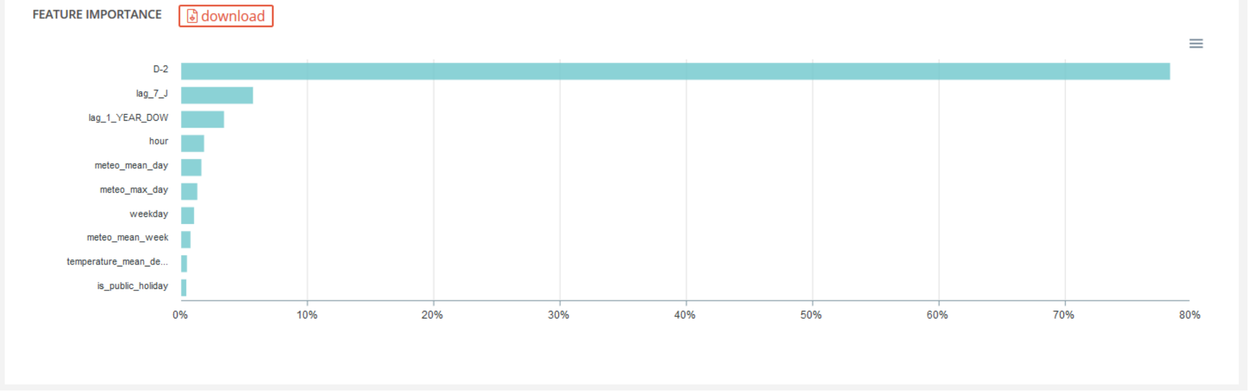
Here we can see more model information, hyper-parameters of the given model and features type retained for the training.

Features types are defined as follow :

- lin : linear features
- ohe : categorical features processed by « one hot encoding »
- freq : categorical features processed by « frequency encoding »
- tenc : categorical features processed by « target encoding »
- ee : categorical features processed by « entity embedding »
- text : text features
- poly : polynomial features
- pca : pca
- kmeans : kmeans clustering done on linear features

It is possible to download hyper-parameters as a JSON and also to download the cross-validation file of the full training set by clicking the top right button.

Feature importance of the selected model will also be displayed :

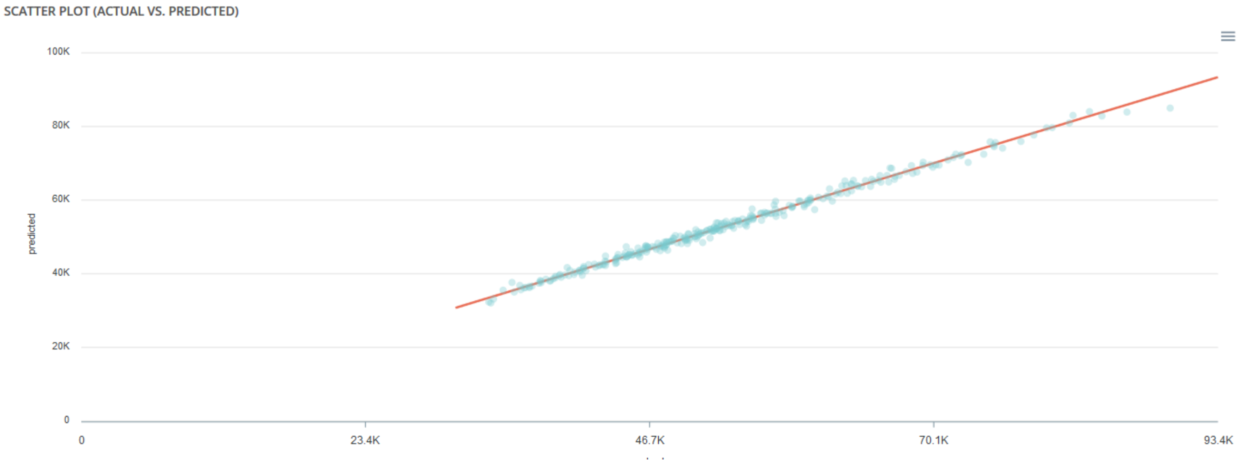


Finally, some performance analysis of models are available. They will differ depending of the usecase type

6.2.1 Regression models

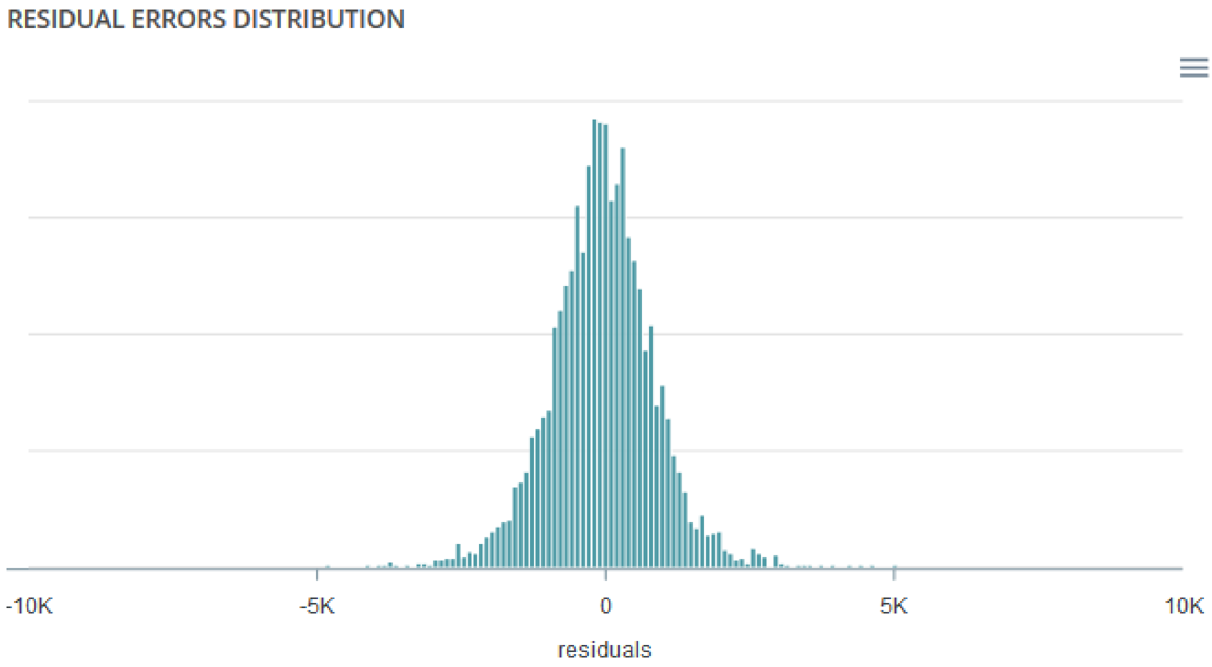
Scatter plot

This graph illustrates the actual values versus the values predicted by the model. A powerful model gathers the point cloud around the orange line.



Residuals

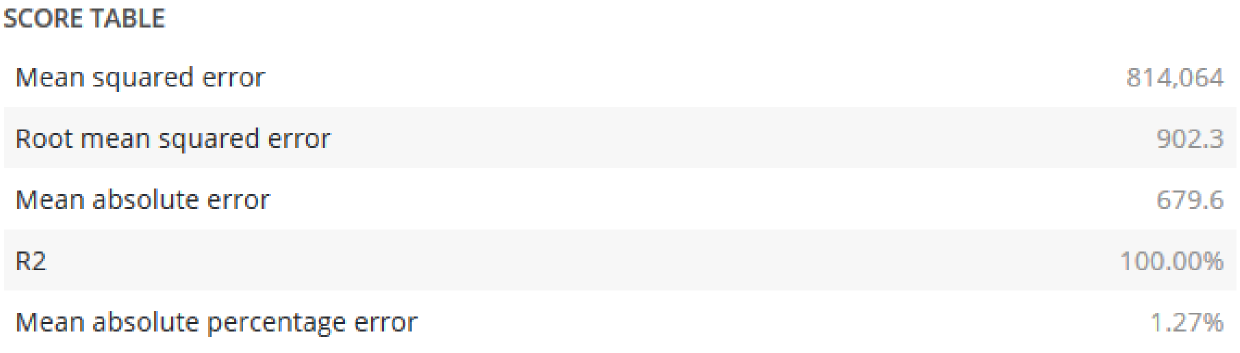
This graph illustrates the dispersion of errors, i.e. residuals. A successful model displays centered and symmetric residues around 0.



Score table

Among the displayed metrics, we have :

- The mean square error (MSE)
- The root of the mean square error (RMSE)
- The mean absolute error (MAE)
- The coefficient of determination (R2)
- The mean absolute percentage error (MAPE)



SCORE TABLE

Mean squared error	814,064
Root mean squared error	902.3
Mean absolute error	679.6
R2	100.00%
Mean absolute percentage error	1.27%

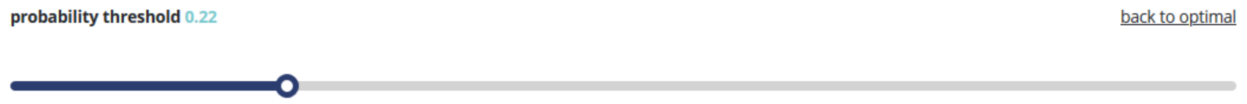
6.2.2 Classification models

Slider

For a binary classification, some graphs and scores may vary according to a probability threshold in relation to which the upper values are considered positive and the lower values negative. This is the case for :

- The scores
- The confusion matrix
- The cost matrix

Thus, you can define the optimal threshold according to your preferences. By default, the threshold corresponds to the one that minimizes the F1-Score. Should you change the position of the threshold, you can click on the « back to optimal » link to position the cursor back to the probability that maximizes the F1-Score.

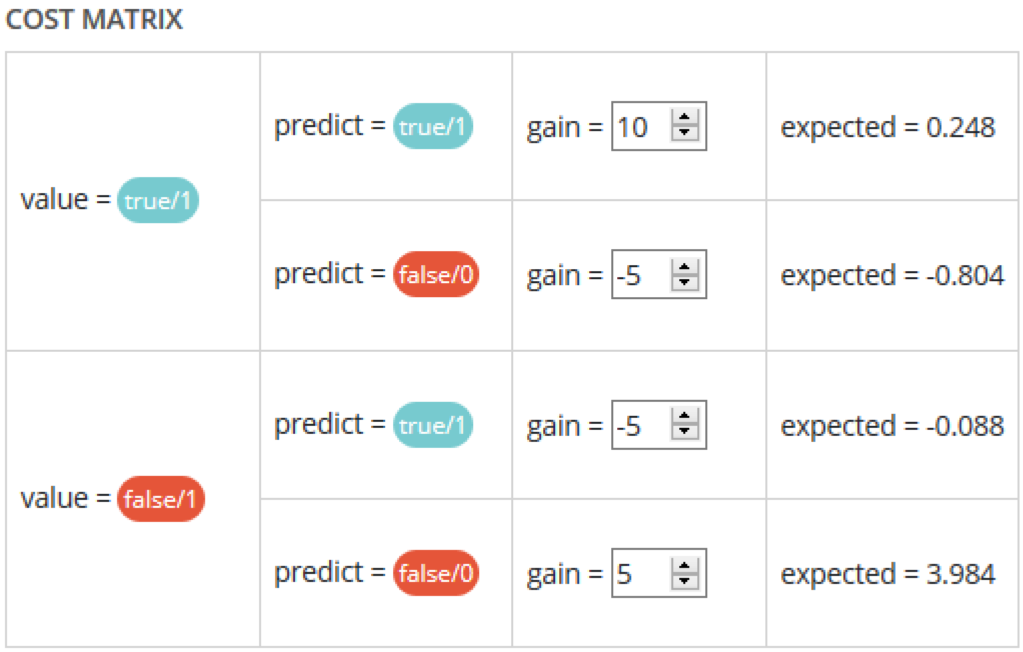


probability threshold 0.22

Cost matrix

Provided that you can quantify the gains or losses associated with true positives, false positives, false negatives, and true negatives, the cost matrix works as an estimator of the average gain for a prediction made by your classifier. In the case explained below, each prediction yields an average of €2.83.

The matrix is initiated with default values that can be freely modified.



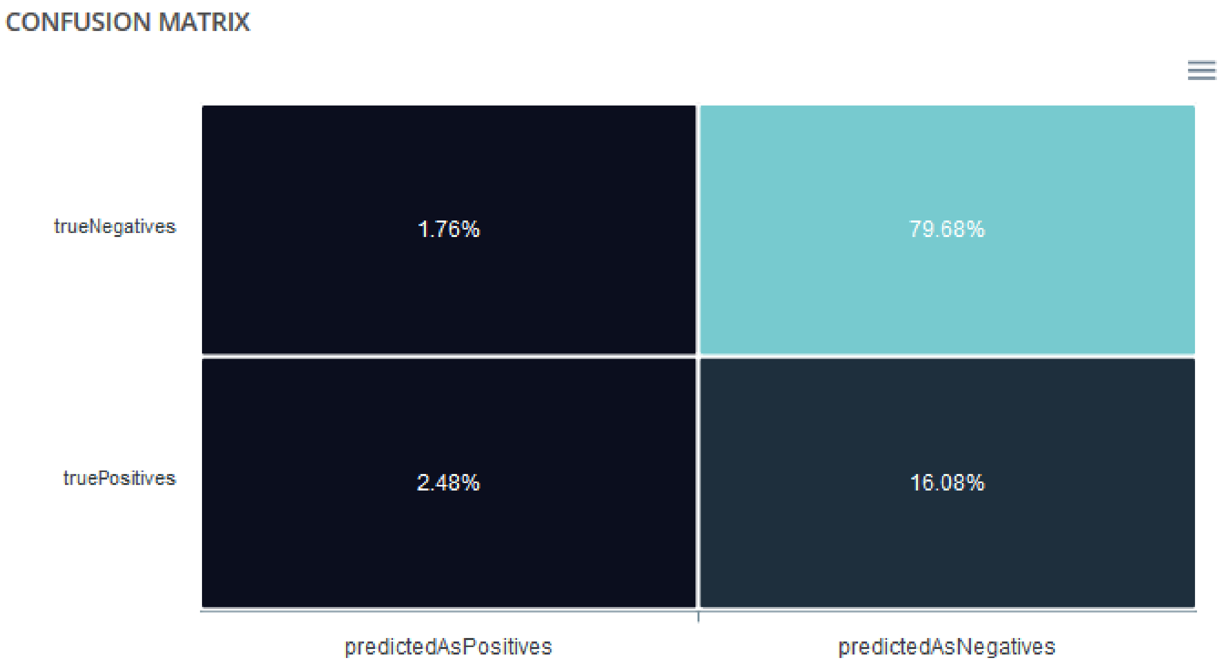
COST MATRIX

value = true/1	predict = true/1	gain = 10	expected = 0.248
	predict = false/0	gain = -5	expected = -0.804
value = false/1	predict = true/1	gain = -5	expected = -0.088
	predict = false/0	gain = 5	expected = 3.984

Confusion matrix

The confusion matrix helps to understand the distribution of true positives, false positives, true negatives and false negatives according to the probability threshold. The boxes in the matrix are darker for large quantities and lighter for small quantities.

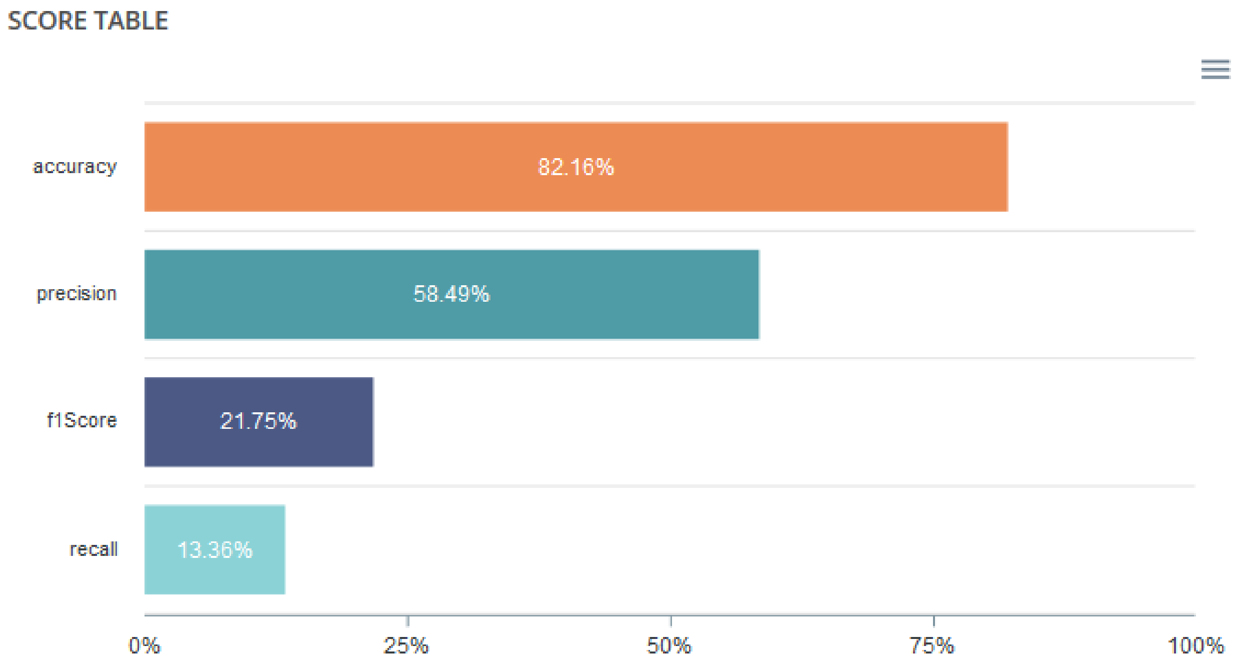
Ideally, most classified individuals should be located on the diagonal of your matrix.



Score table

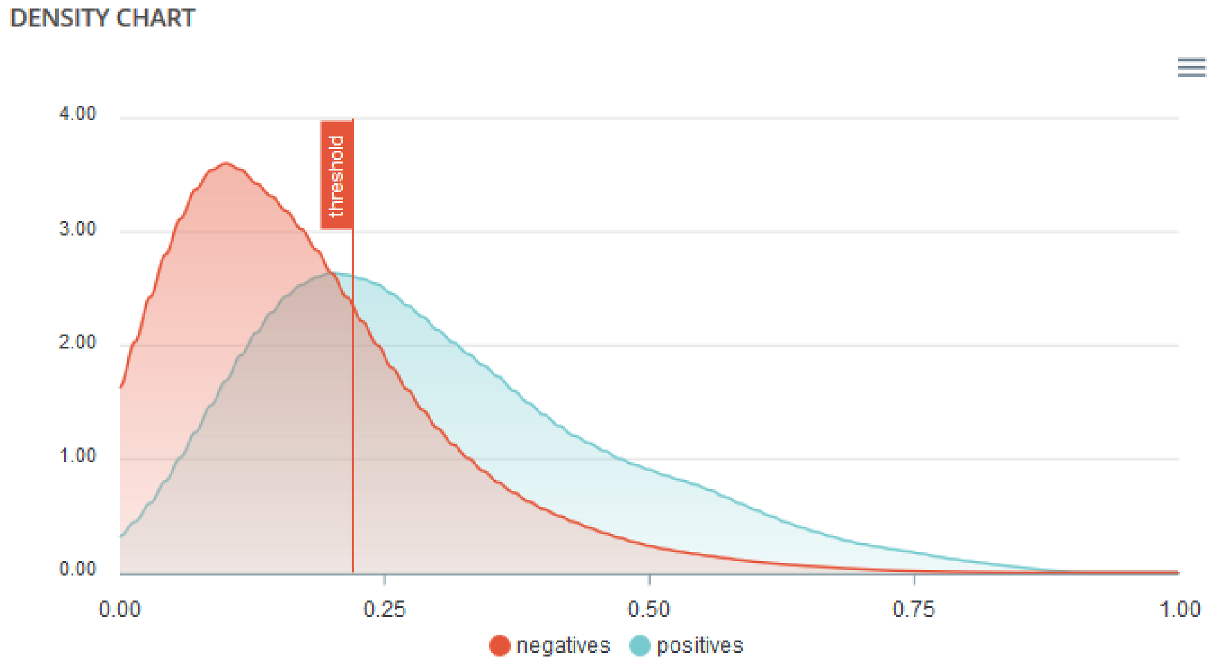
Among the displayed metrics, we have :

- Accuracy : The sum of true positives and true negatives divided by the number of individuals
- F1-Score : Harmonic mean of the precision and the recall
- Precision : True positives divided by the sum of positives
- Recall : True positives divided by the sum of true positives and false negatives



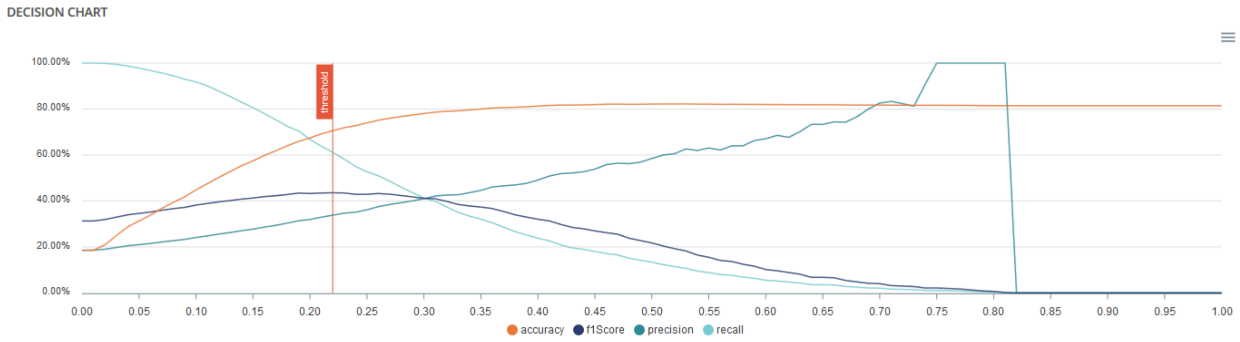
Density chart

The density graph allows you to understand the density of positives and negatives among the predictions. The more efficient your classifier is, the more the 2 density curves are disjointed and centered around 0 and 1.



Gain chart

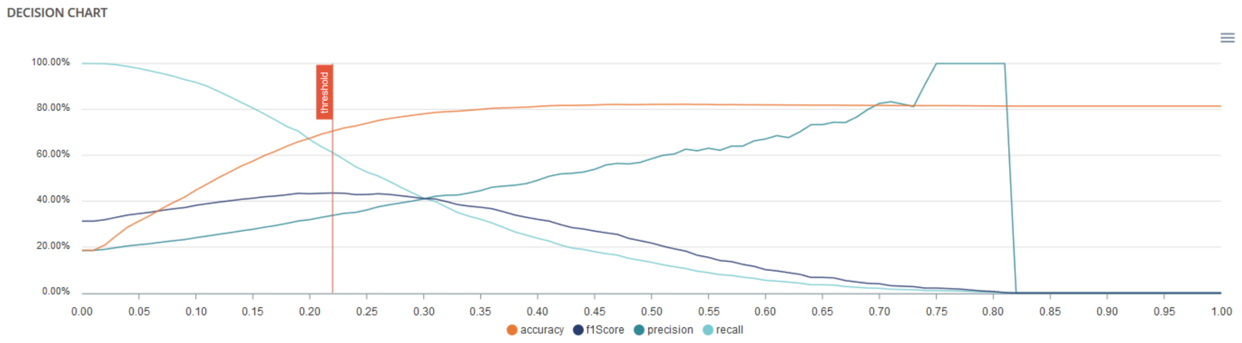
The gain graph allows you to quickly visualize the optimal threshold to select in order to maximise the gain as defined in the cost matrix.



Decision chart

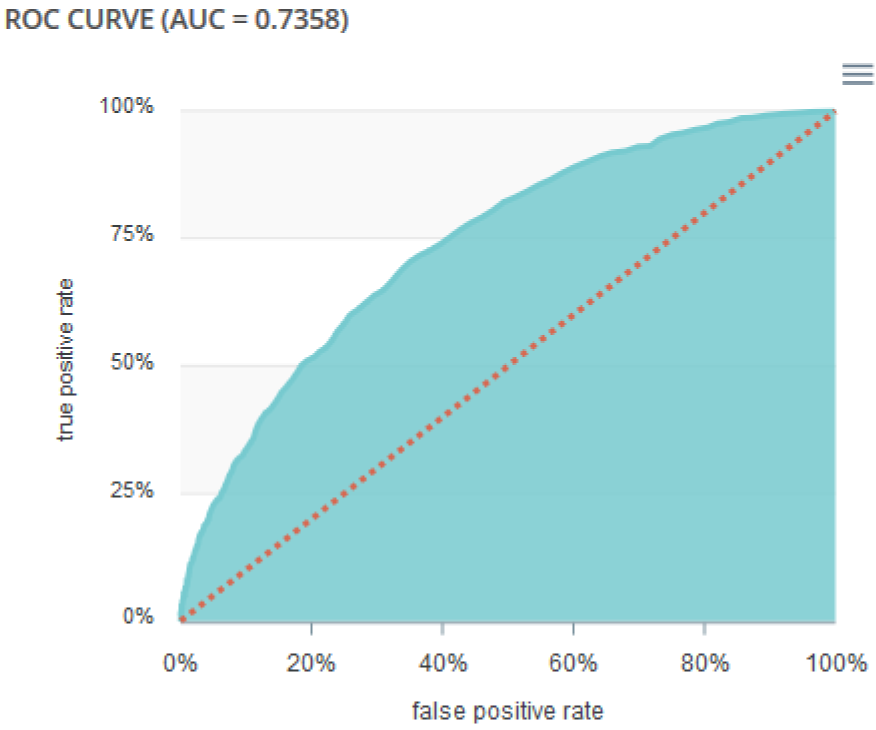
The decision graph allows you to quickly visualize all the proposed metrics, regardless of the probability threshold. Thus, one can visualize at what point the maximum of each metric is reached, making it possible for one to choose its selection threshold.

It should be noted that the discontinuous line curve illustrates the expected gain by prediction. It is therefore totally linked to the cost matrix and will be updated if you change the gain of one of the 4 possible cases in the matrix.



ROC curve

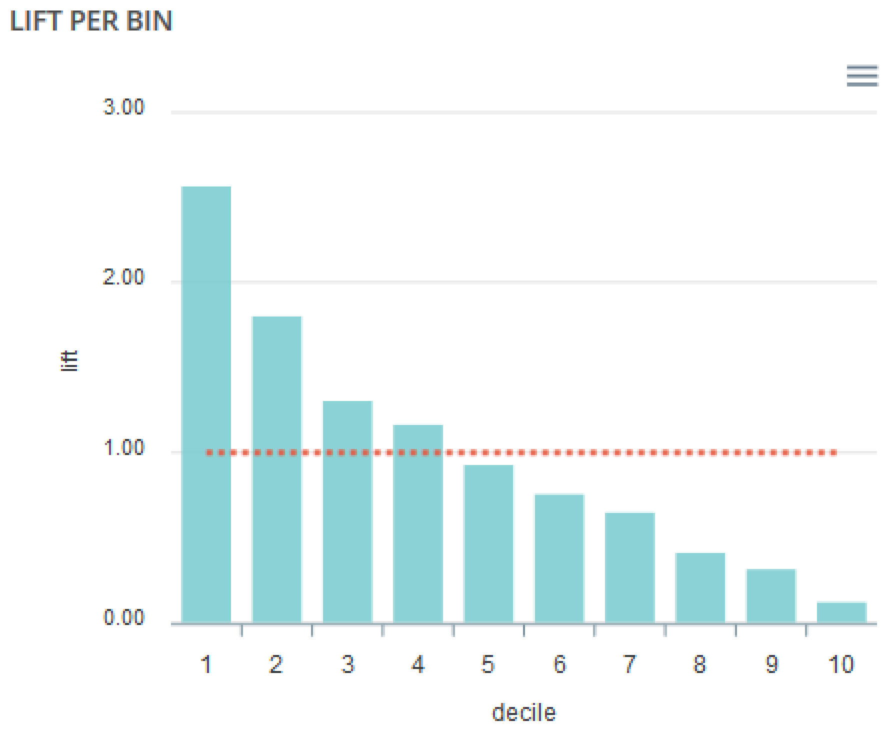
The ROC curve illustrates the overall performance of the classifier (more info : https://en.wikipedia.org/wiki/Receiver_operating_characteristic). The more the curve appears linear, the closer the quality of the classifier is to a random process. The more the curve tends towards the upper left side, the closer the quality of your classifier is to perfection.



Lift per bin

The predictions are sorted in descending order and the lift of each decile (bin) is indicated in the graph. Example : A lift of 4 means that there are 4 times more positives in the considered decile than on average in the population.

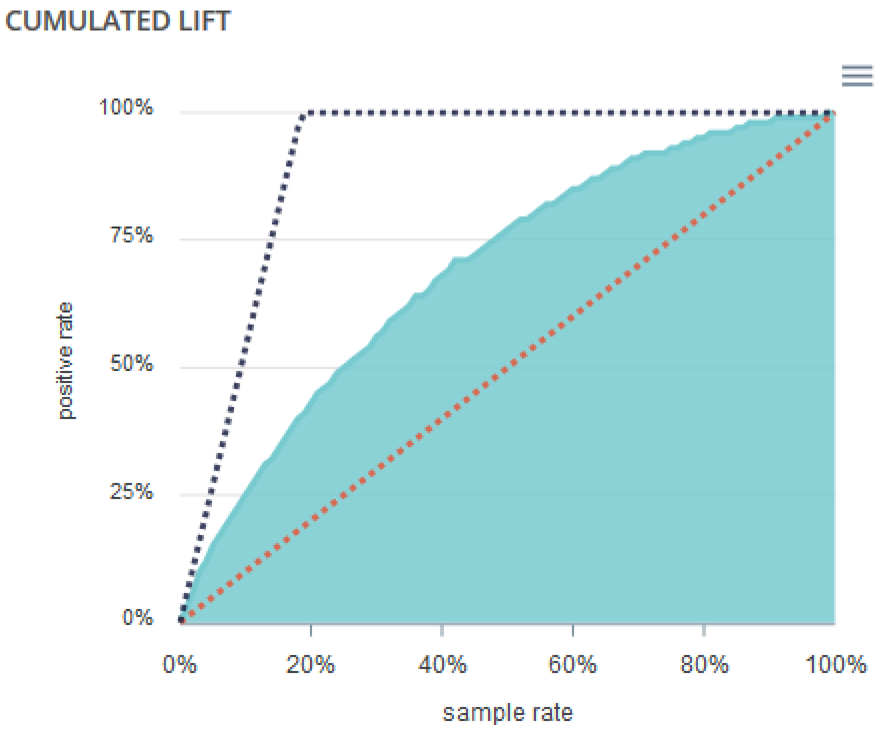
The orange horizontal line shows a lift at 1.



Cumulated lift

The objective of this curve is to measure what proportion of the positives can be achieved by targeting only a subsample of the population. It therefore illustrates the proportion of positives according to the proportion of the selected sub-population.

A diagonal line (orange) illustrates a random pattern (= x % of the positives are obtained by randomly drawing x % of the population). A segmented line (blue) illustrates a perfect model (= 100% of positives are obtained by targeting only the population's positive rate).

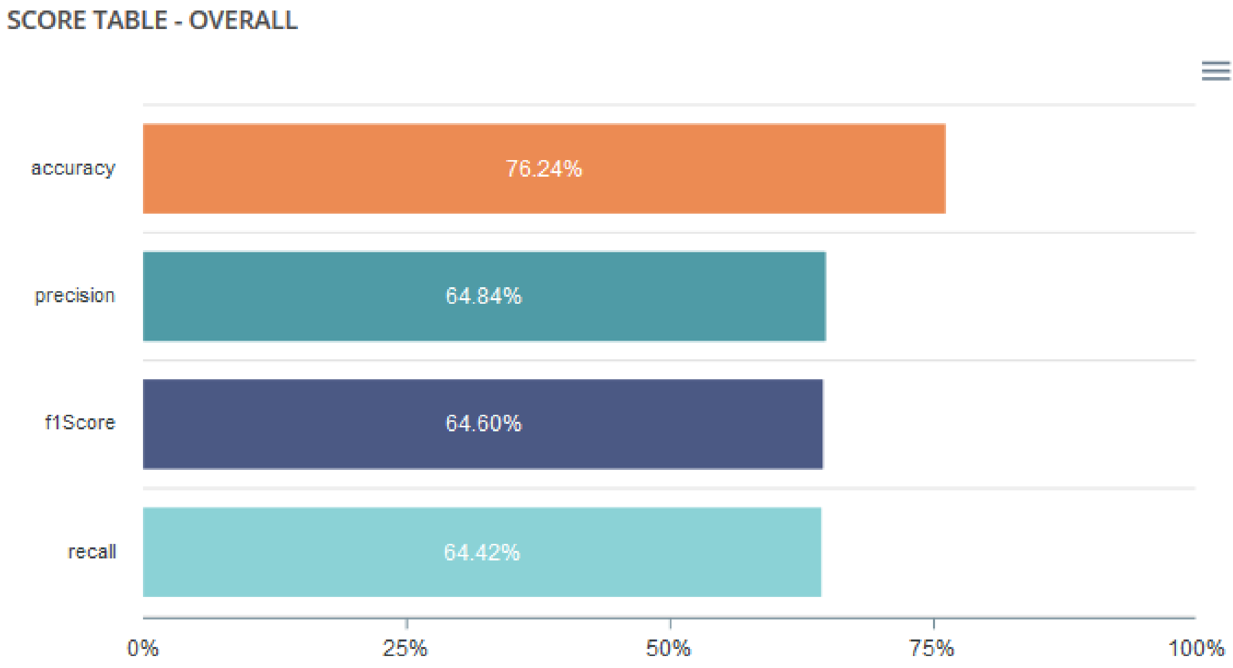


6.2.3 Multiclassification models

Score table

Among the displayed metrics, we have displayed the macro averaged :

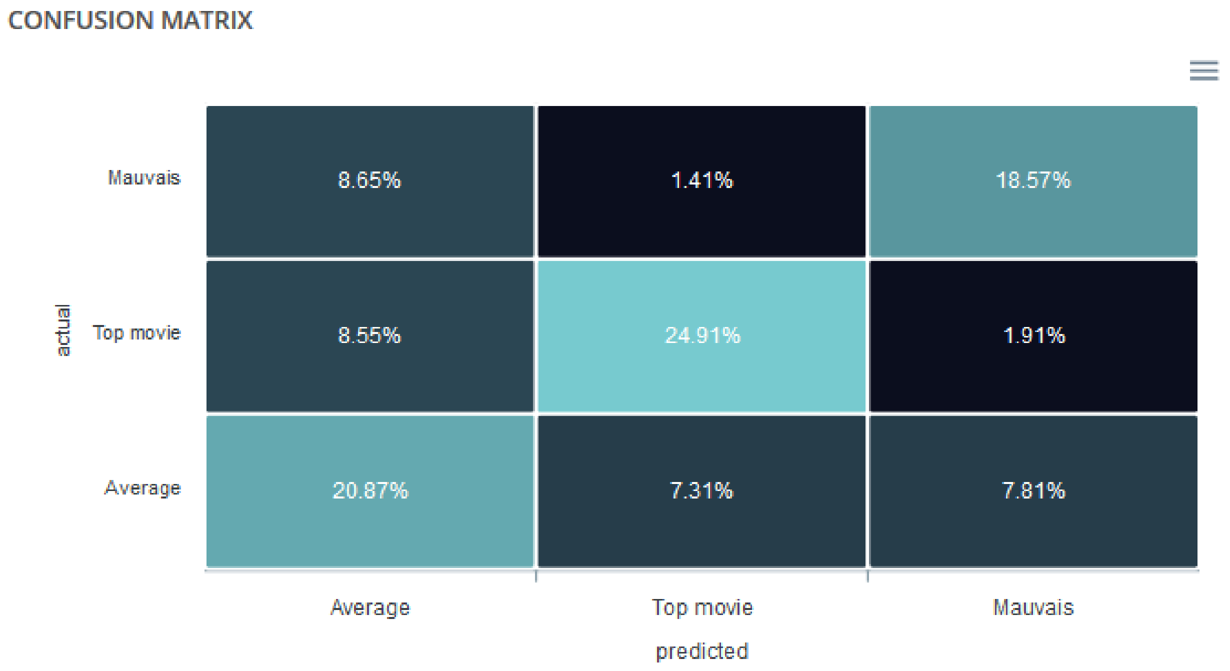
- — Accuracy
- — Precision
- — Recall
- — F1-Score



Confusion matrix

The confusion matrix makes it possible to understand the distribution of predicted values compared to actual values between classes. The boxes in the matrix are darker for large quantities and lighter for small quantities.

Ideally, most classified individuals should be located on the diagonal of your matrix.



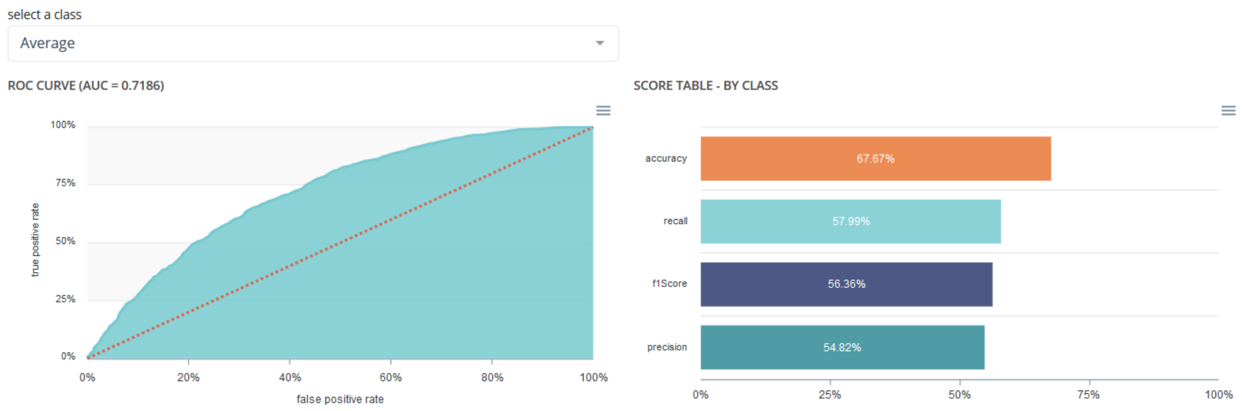
	Average	Top movie	Mauvais
Mauvais	8.65%	1.41%	18.57%
Top movie	8.55%	24.91%	1.91%
Average	20.87%	7.31%	7.81%

ROC curve & per class analysis

The ROC curve illustrates the overall performance of the classifier (more info : https://en.wikipedia.org/wiki/Receiver_operating_characteristic). The more the curve appears linear, the closer the quality of the classifier is to a random process. The more the curve tends towards the upper left side, the closer the quality of your classifier is to perfection.

Within the framework of multiclass classification, there are as many curves as there are classes. They are calculated in « one- versus-all ».

Also, for each class, you'll have detailed metric of it on the right part of the screen :

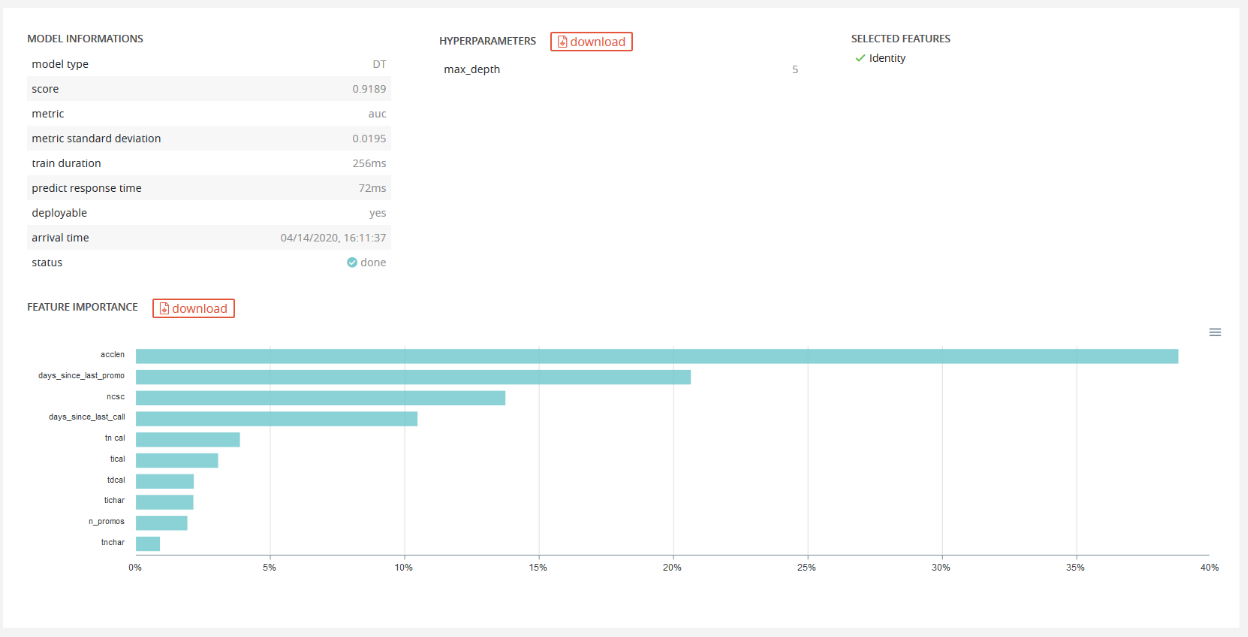


6.2.4 Simple models

For every use case, 2 `simples models` will be trained. One will be tree based the other one will be linear (or logistic, depending of the corresponding `TRAINING TYPE`) Please keep in mind that, because of the nature of simplified models, their predictive power might be lower than more complex one but are easier to understand and to communication to business users.

Decision tree

A simplified decision tree is available. It will have the same level of information of other models, can be fully displayed and is also exportable as SQL, PYTHON and R directly with code generated



MODEL INFORMATIONS

model type	DT
score	0.9189
metric	auc
metric standard deviation	0.0195
train duration	256ms
predict response time	72ms
deployable	yes
arrival time	04/14/2020, 16:11:37
status	✓ done

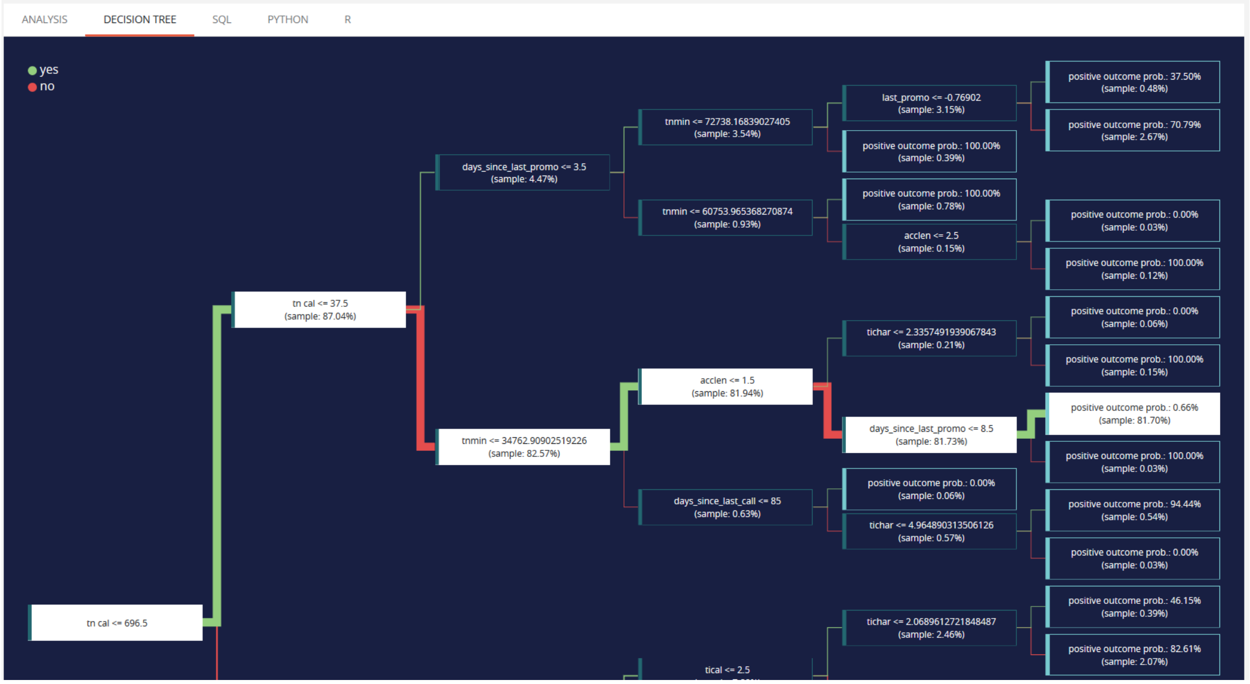
HYPERPARAMETERS download

max_depth	5

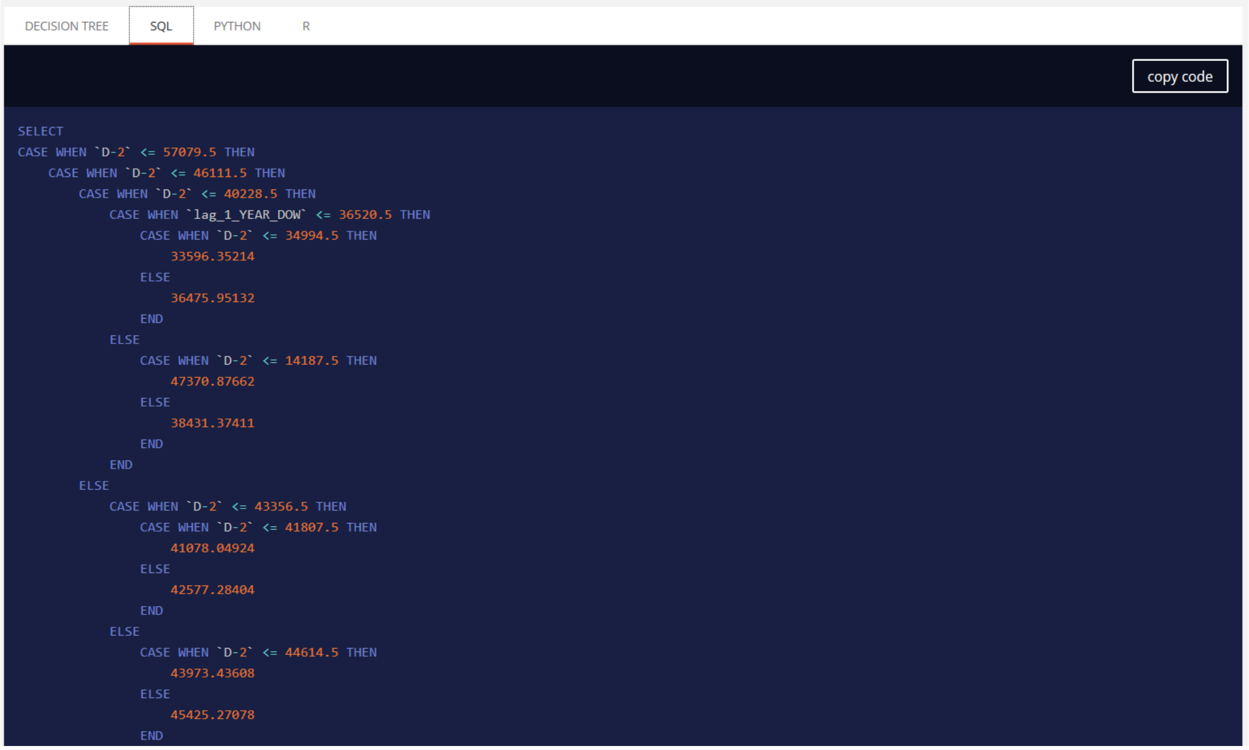
SELECTED FEATURES

✓ Identity

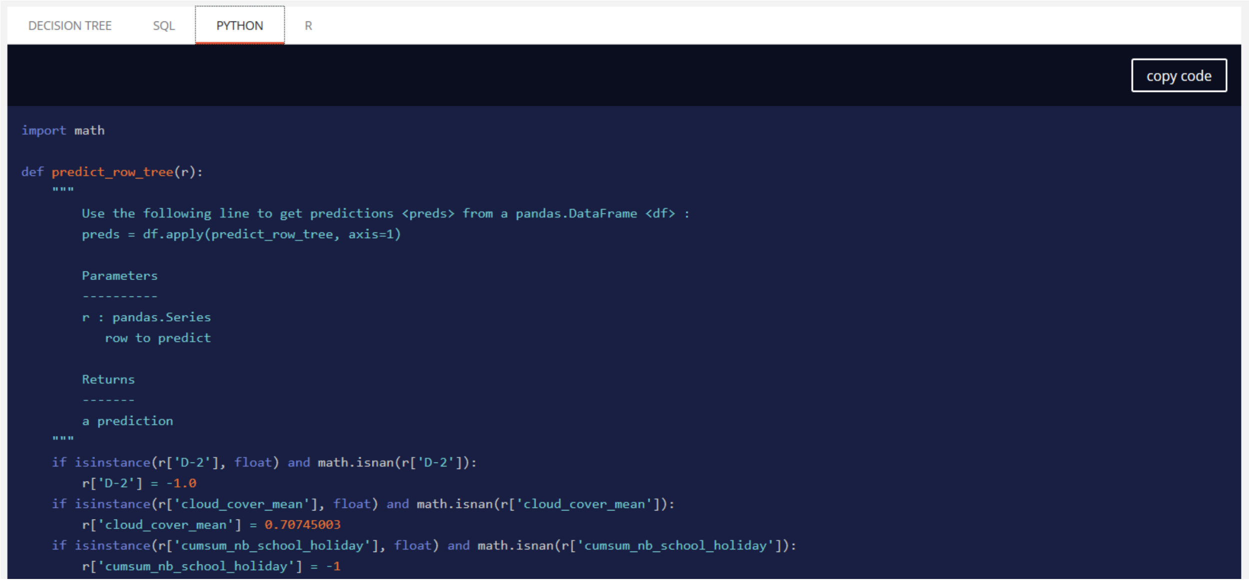
FEATURE IMPORTANCE download



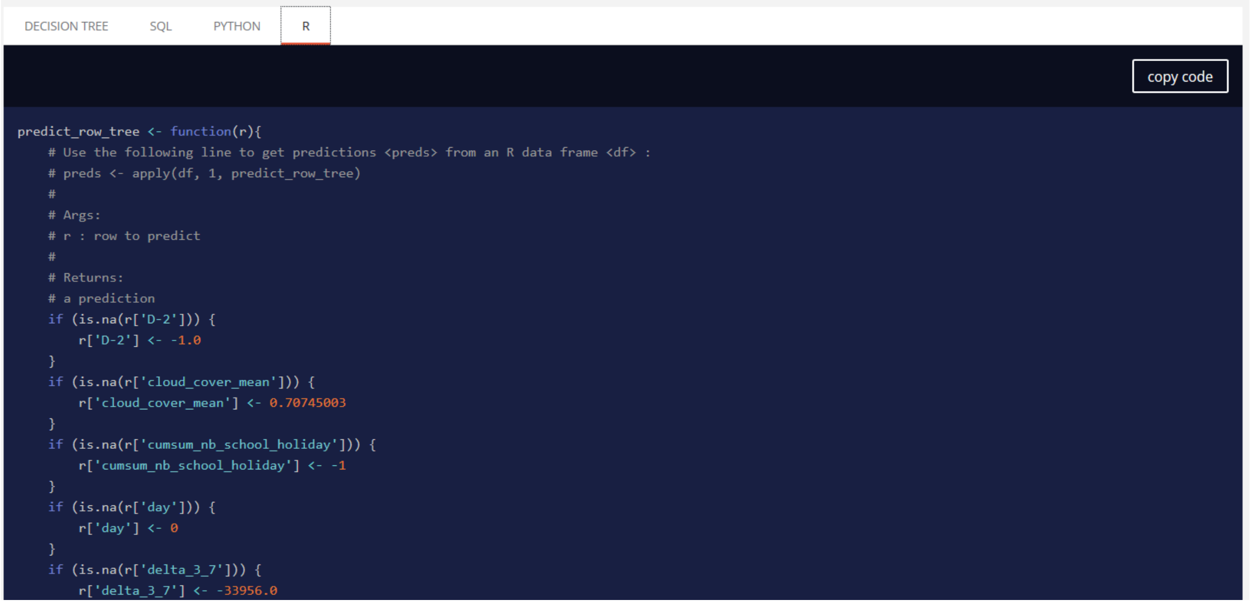
ANALYSIS
DECISION TREE
SQL
PYTHON
R
● yes
● no
tn cal <= 696.5
tn cal <= 37.5
(sample: 87.04%)
days_since_last_promo <= 3.5
(sample: 4.47%)
tnmin <= 34762.90902519226
(sample: 82.57%)
tnmin <= 72738.16839027405
(sample: 3.54%)
tnmin <= 60753.965368270874
(sample: 0.93%)
acclen <= 1.5
(sample: 81.94%)
days_since_last_call <= 85
(sample: 0.63%)
tical <= 2.5
last_promo <= -0.76902
(sample: 3.15%)
positive outcome prob.: 100.00%
(sample: 0.39%)
positive outcome prob.: 100.00%
(sample: 0.78%)
acclen <= 2.5
(sample: 0.15%)
tichar <= 2.3357491939067843
(sample: 0.21%)
days_since_last_promo <= 8.5
(sample: 81.73%)
positive outcome prob.: 0.00%
(sample: 0.06%)
tichar <= 4.964890313506126
(sample: 0.57%)
tichar <= 2.0689612721848487
(sample: 2.46%)
positive outcome prob.: 37.50%
(sample: 0.48%)
positive outcome prob.: 70.79%
(sample: 2.67%)
positive outcome prob.: 0.00%
(sample: 0.03%)
positive outcome prob.: 100.00%
(sample: 0.12%)
positive outcome prob.: 0.00%
(sample: 0.06%)
positive outcome prob.: 100.00%
(sample: 0.15%)
positive outcome prob.: 0.66%
(sample: 81.70%)
positive outcome prob.: 100.00%
(sample: 0.03%)
positive outcome prob.: 94.44%
(sample: 0.54%)
positive outcome prob.: 0.00%
(sample: 0.03%)
positive outcome prob.: 46.15%
(sample: 0.39%)
positive outcome prob.: 82.61%
(sample: 2.07%)



```
SELECT
CASE WHEN `D-2` <= 57079.5 THEN
    CASE WHEN `D-2` <= 46111.5 THEN
        CASE WHEN `D-2` <= 40228.5 THEN
            CASE WHEN `lag_1_YEAR_DOW` <= 36520.5 THEN
                CASE WHEN `D-2` <= 34994.5 THEN
                    33596.35214
                ELSE
                    36475.95132
                END
            ELSE
                CASE WHEN `D-2` <= 14187.5 THEN
                    47370.87662
                ELSE
                    38431.37411
                END
            END
        ELSE
            CASE WHEN `D-2` <= 43356.5 THEN
                CASE WHEN `D-2` <= 41807.5 THEN
                    41078.04924
                ELSE
                    42577.28404
                END
            ELSE
                CASE WHEN `D-2` <= 44614.5 THEN
                    43973.43608
                ELSE
                    45425.27078
                END
```



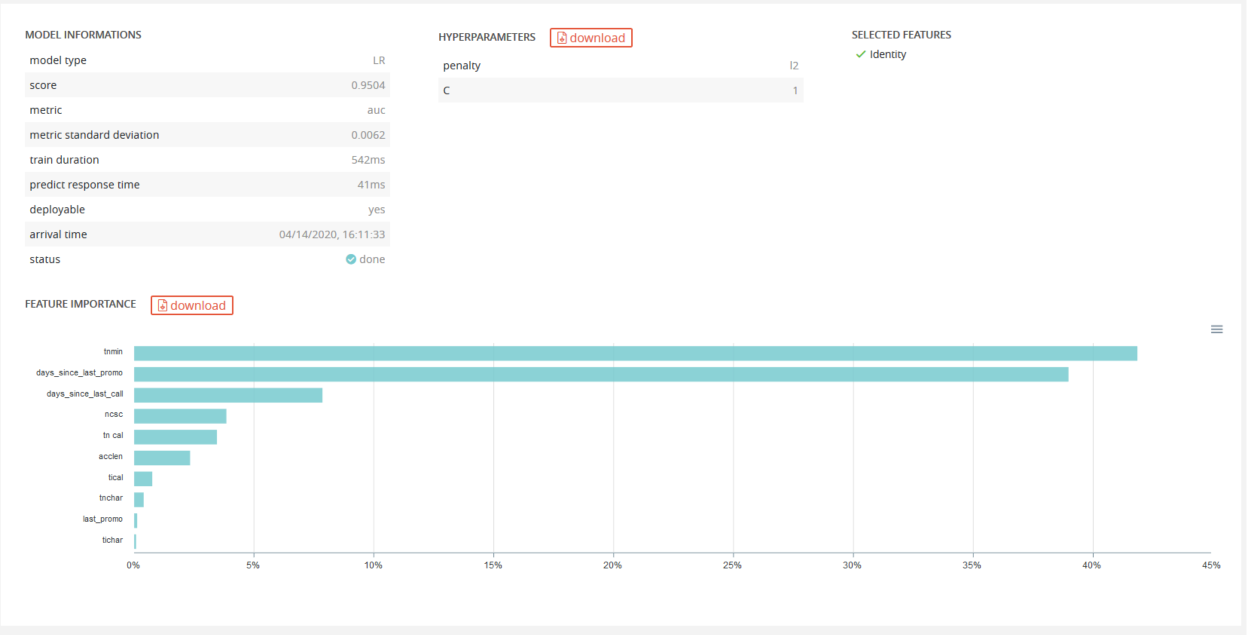
```
import math

def predict_row_tree(r):
    """
        Use the following line to get predictions <preds> from a pandas.DataFrame <df> :
        preds = df.apply(predict_row_tree, axis=1)

        Parameters
        ----------
        r : pandas.Series
           row to predict

        Returns
        -------
        a prediction
    """
    if isinstance(r['D-2'], float) and math.isnan(r['D-2']):
        r['D-2'] = -1.0
    if isinstance(r['cloud_cover_mean'], float) and math.isnan(r['cloud_cover_mean']):
        r['cloud_cover_mean'] = 0.70745003
    if isinstance(r['cumsum_nb_school_holiday'], float) and math.isnan(r['cumsum_nb_school_holiday']):
        r['cumsum_nb_school_holiday'] = -1
```



```
predict_row_tree <- function(r){
    # Use the following line to get predictions <preds> from an R data frame <df> :
    # preds <- apply(df, 1, predict_row_tree)
    #
    # Args:
    # r : row to predict
    #
    # Returns:
    # a prediction
    if (is.na(r['D-2'])) {
        r['D-2'] <- -1.0
    }
    if (is.na(r['cloud_cover_mean'])) {
        r['cloud_cover_mean'] <- 0.70745003
    }
    if (is.na(r['cumsum_nb_school_holiday'])) {
        r['cumsum_nb_school_holiday'] <- -1
    }
    if (is.na(r['day'])) {
        r['day'] <- 0
    }
    if (is.na(r['delta_3_7'])) {
        r['delta_3_7'] <- -33956.0
```

Linear model

A simplified linear (or logistic when doing a classification) regression is available. It will have the same level of information of other models, can be fully displayed and is also exportable as SQL, PYTHON and R directly with code generated



MODEL INFORMATIONS

model type	LR
score	0.9504
metric	auc
metric standard deviation	0.0062
train duration	542ms
predict response time	41ms
deployable	yes
arrival time	04/14/2020, 16:11:33
status	✓ done

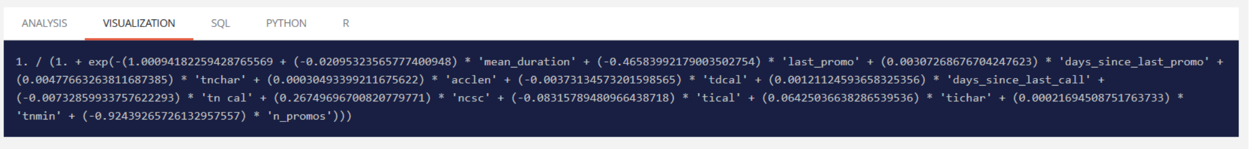
HYPERPARAMETERS download

penalty	l2
C	1

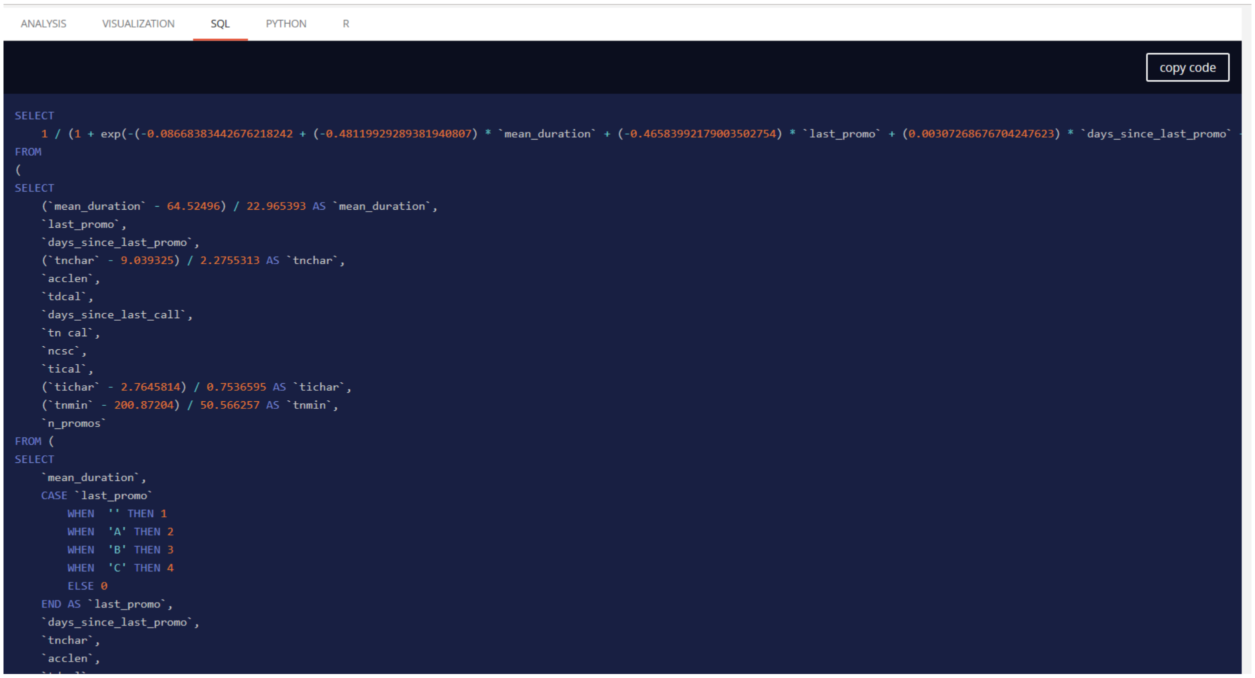
SELECTED FEATURES

✓ Identity

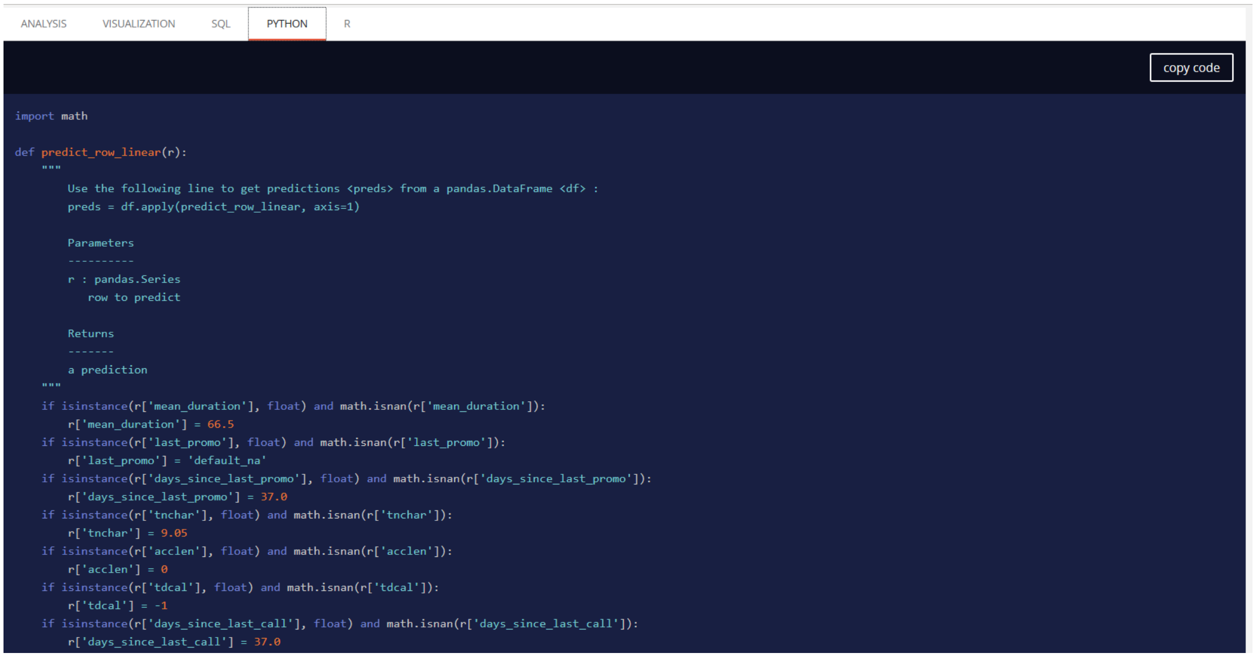
FEATURE IMPORTANCE download



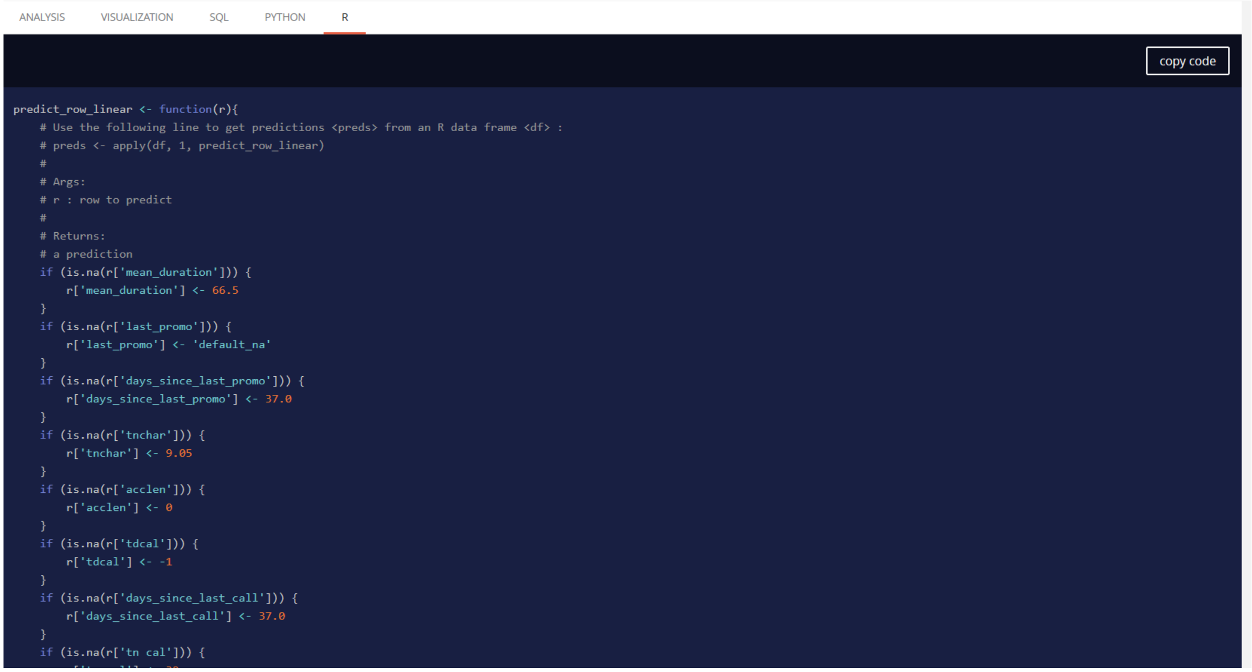
```
1. / (1. + exp(-(1.00094182259428765569 + (-0.02095323565777400948) * 'mean_duration' + (-0.46583992179003502754) * 'last_promo' + (0.00307268676704247623) * 'days_since_last_promo' +
(0.00477663263811687385) * 'tnchar' + (0.00030493399211675622) * 'acclen' + (-0.00373134573201598565) * 'tdcal' + (0.00121124593658325356) * 'days_since_last_call' +
(-0.00732859933757622293) * 'tn cal' + (0.26749696700820779771) * 'ncsc' + (-0.08315789480966438718) * 'tical' + (0.06425036638286539536) * 'tichar' + (0.00021694508751763733) *
'tnmin' + (-0.92439265726132957557) * 'n_promos')))
```



```sql
SELECT
    1 / (1 + exp(-(-0.08668383442676218242 + (-0.48119929289381940807) * `mean_duration` + (-0.46583992179003502754) * `last_promo` + (0.00307268676704247623) * `days_since_last_promo`
FROM
(
SELECT
    (`mean_duration` - 64.52496) / 22.965393 AS `mean_duration`,
    `last_promo`,
    `days_since_last_promo`,
    (`tnchar` - 9.039325) / 2.2755313 AS `tnchar`,
    `acclen`,
    `tdcal`,
    `days_since_last_call`,
    `tn cal`,
    `ncsc`,
    `tical`,
    (`tichar` - 2.7645814) / 0.7536595 AS `tichar`,
    (`tnmin` - 200.87204) / 50.566257 AS `tnmin`,
    `n_promos`
FROM (
SELECT
    `mean_duration`,
    CASE `last_promo`
        WHEN  '' THEN 1
        WHEN  'A' THEN 2
        WHEN  'B' THEN 3
        WHEN  'C' THEN 4
        ELSE 0
    END AS `last_promo`,
    `days_since_last_promo`,
    `tnchar`,
    `acclen`,
```



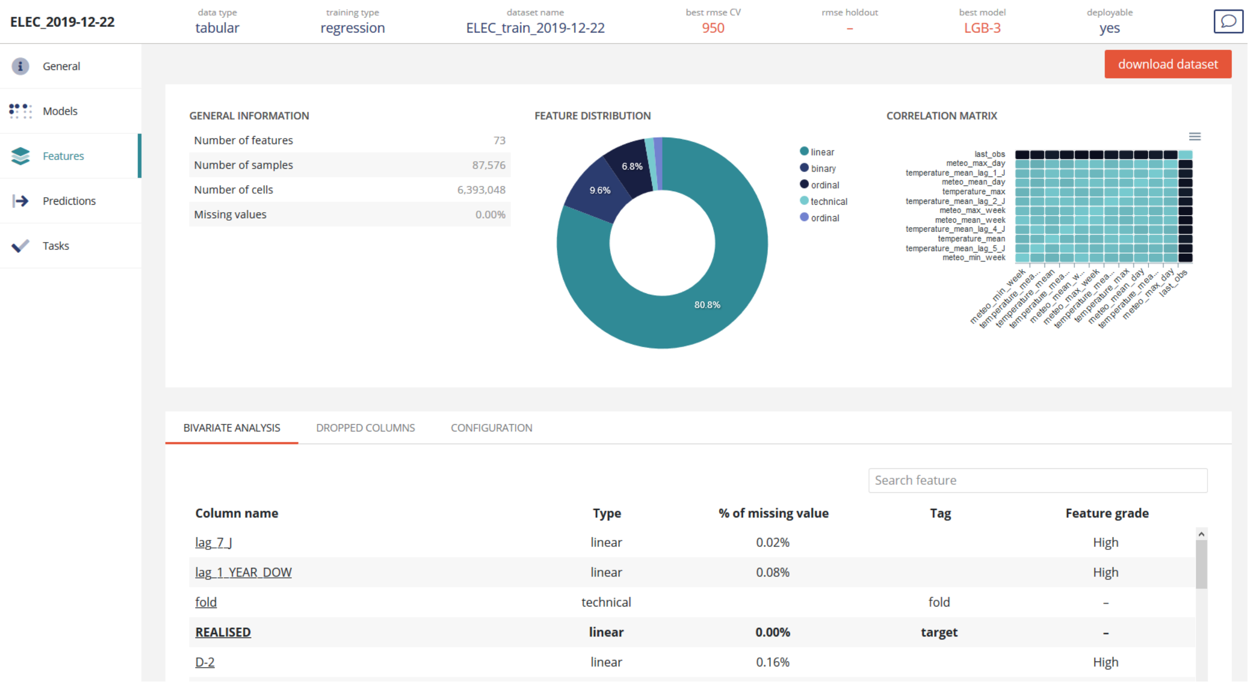
```python
import math

def predict_row_linear(r):
    """
        Use the following line to get predictions <preds> from a pandas.DataFrame <df> :
        preds = df.apply(predict_row_linear, axis=1)

        Parameters
        ----------
        r : pandas.Series
           row to predict

        Returns
        -------
        a prediction
    """
    if isinstance(r['mean_duration'], float) and math.isnan(r['mean_duration']):
        r['mean_duration'] = 66.5
    if isinstance(r['last_promo'], float) and math.isnan(r['last_promo']):
        r['last_promo'] = 'default_na'
    if isinstance(r['days_since_last_promo'], float) and math.isnan(r['days_since_last_promo']):
        r['days_since_last_promo'] = 37.0
    if isinstance(r['tnchar'], float) and math.isnan(r['tnchar']):
        r['tnchar'] = 9.05
    if isinstance(r['acclen'], float) and math.isnan(r['acclen']):
        r['acclen'] = 0
    if isinstance(r['tdcal'], float) and math.isnan(r['tdcal']):
        r['tdcal'] = -1
    if isinstance(r['days_since_last_call'], float) and math.isnan(r['days_since_last_call']):
        r['days_since_last_call'] = 37.0
```



```
predict_row_linear <- function(r){
    # Use the following line to get predictions <preds> from an R data frame <df> :
    # preds <- apply(df, 1, predict_row_linear)
    #
    # Args:
    # r : row to predict
    #
    # Returns:
    # a prediction
    if (is.na(r['mean_duration'])) {
        r['mean_duration'] <- 66.5
    }
    if (is.na(r['last_promo'])) {
        r['last_promo'] <- 'default_na'
    }
    if (is.na(r['days_since_last_promo'])) {
        r['days_since_last_promo'] <- 37.0
    }
    if (is.na(r['tnchar'])) {
        r['tnchar'] <- 9.05
    }
    if (is.na(r['acclen'])) {
        r['acclen'] <- 0
    }
    if (is.na(r['tdcal'])) {
        r['tdcal'] <- -1
    }
    if (is.na(r['days_since_last_call'])) {
        r['days_since_last_call'] <- 37.0
    }
    if (is.na(r['tn cal'])) {
```

6.3 Features

Detailed information about features linked to a usecase are available in the screen :



ELEC_2019-12-22

data type	training type	dataset name	best rmse CV	rmse holdout	best model	deployable
tabular	regression	ELEC_train_2019-12-22	950	–	LGB-3	yes

GENERAL INFORMATION

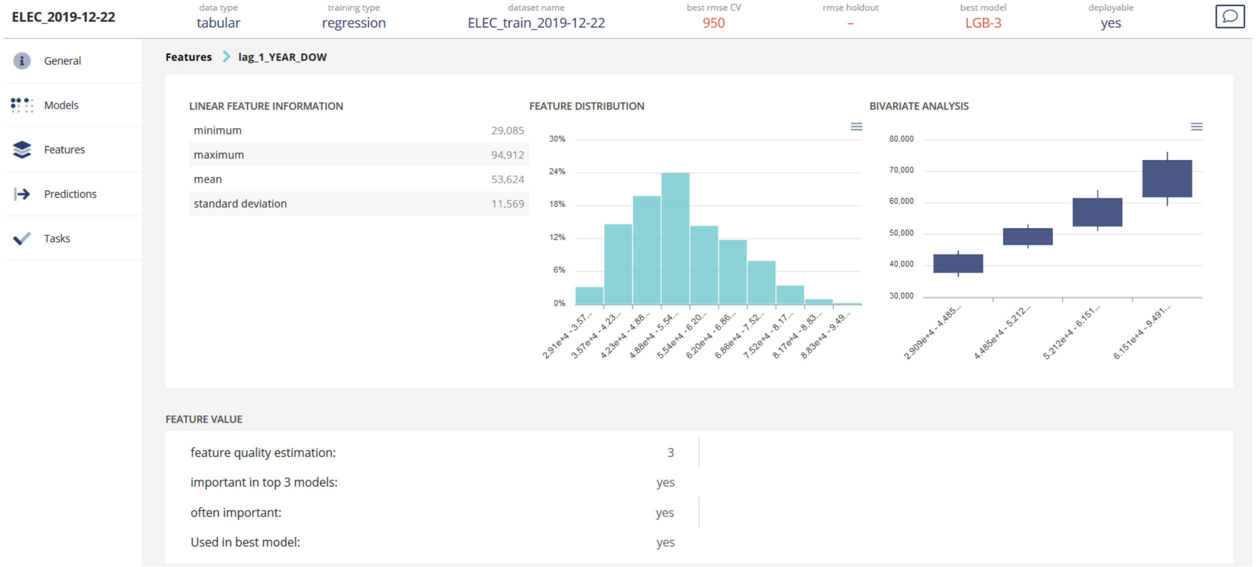
Number of features	73
Number of samples	87,576
Number of cells	6,393,048
Missing values	0.00%

Column name	Type	% of missing value	Tag	Feature grade
lag_7_J	linear	0.02%		High
lag_1_YEAR_DOW	linear	0.08%		High
fold	technical		fold	–
REALISED	**linear**	**0.00%**	**target**	**–**
D-2	linear	0.16%		High

You'll find the same information that in the dataset page, with some additional statistics calculated on this specific usecase :

- — Feature grade : in one word if the feature is globally important for the use case
- — Dropped column for the usecase
- — Configuration of feature engineering to be done on the usecase

Clicking on a given feature will give you some detailed analytics on it :



ELEC_2019-12-22

data type: tabular | training type: regression | dataset name: ELEC_train_2019-12-22 | best rmse CV: 950 | rmse holdout: – | best model: LGB-3 | deployable: yes

LINEAR FEATURE INFORMATION

minimum	29,085
maximum	94,912
mean	53,624
standard deviation	11,569

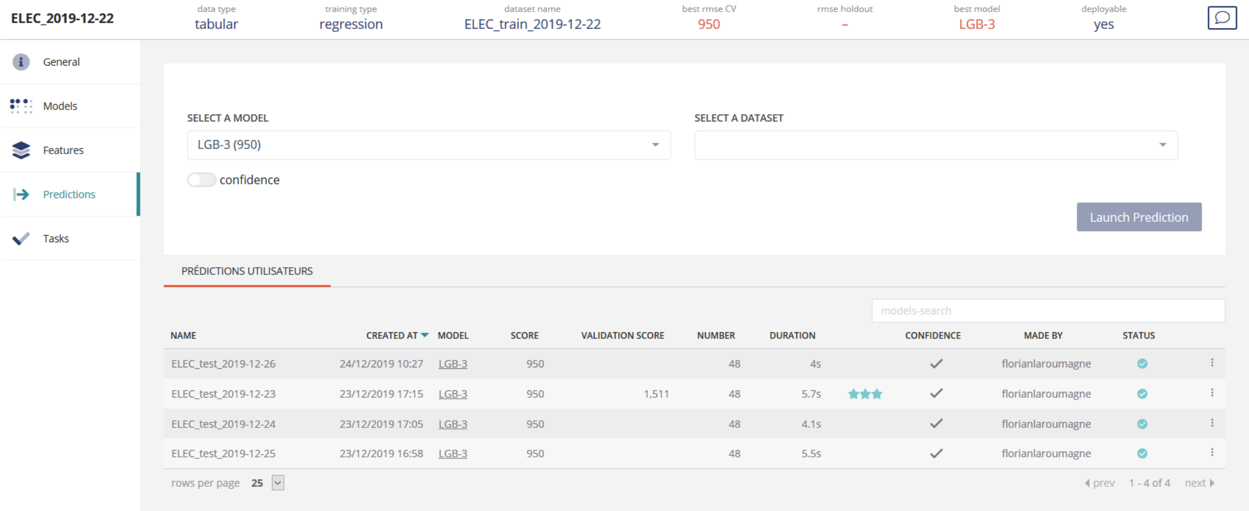
FEATURE VALUE

feature quality estimation:	3
important in top 3 models:	yes
often important:	yes
Used in best model:	yes

You will find a summary of the feature, its distribution and its bivariate distribution conditioned to the TARGET

6.4 Predictions

All predictions linked to a usecase are available in the screen :



ELEC_2019-12-22

data type: tabular | training type: regression | dataset name: ELEC_train_2019-12-22 | best rmse CV: 950 | rmse holdout: – | best model: LGB-3 | deployable: yes

SELECT A MODEL: LGB-3 (950)

SELECT A DATASET:

○ confidence

NAME	CREATED AT	MODEL	SCORE	VALIDATION SCORE	NUMBER	DURATION		CONFIDENCE	MADE BY	STATUS
ELEC_test_2019-12-26	24/12/2019 10:27	LGB-3	950		48	4s		✓	florianlaroumagne	✓
ELEC_test_2019-12-23	23/12/2019 17:15	LGB-3	950	1,511	48	5.7s	★★★	✓	florianlaroumagne	✓
ELEC_test_2019-12-24	23/12/2019 17:05	LGB-3	950		48	4.1s		✓	florianlaroumagne	✓
ELEC_test_2019-12-25	23/12/2019 16:58	LGB-3	950		48	5.5s		✓	florianlaroumagne	✓

In order to create a new prediction, you need to select a model (the best one is selected by default) and to select a dataset you want to predict on and clicking on the « Launch Prediction » button. It is imperative that it has the same structure as the learning set, with the exception of the following technical columns : [`TARGET`, `WEIGHT`, `FOLD`], which are generaly not present in the data set to be predicted.

If the file sent contained a *TARGET*, which may be the case when benchmarking, the platform will automatically calculate the score on the test set and display it next to the estimated cross-validation score.

Note that when columns are missing in the test set, the platform will ask you if you really want to perform the prediction. If you confirm your choice, the relevant columns will be imputed with missing data over the entire dataset. Note that this can influence the performance of the prediction.

All predictions requested by the user will be listed in a table containing :

- Name of the dataset to predict on
- Creation date of the prediction
- Selected model
- Score estimated by the CV process
- Validation score : only available if the TARGET is given in the dataset to be predicted on
- Number of individuals (rows) predicted
- Duration of the prediction
- Has confidence been enabled ?
- Who requester the prediction
- Status of the prediction
- Action link :
  - Explanation
  - Prediction file download
  - Remove the prediction

Please note that if a hold out dataset has been provided, a new tab called « hold out prediction » will contains directly prediction of the hold out for each trained model with the same information of user's predictions.

In addition to providing prediction, Prevision.io offers the ability to quantify the level of certainty around this prediction. Thus, we can know if the algorithms are confident or if they have doubts about the predicted value.

To access this additional information, you must tick the « confidence » slider when sending a dataset for prediction.

Note that this feature slightly increases the prediction time. We therefore recommend that you use it only when you feel it is necessary.

In the case of a regression, you will have 10 new columns in the dataset. Indeed, in addition to the traditional couple (ID, TARGET) you will also have the variables :

- TARGET_quantile=1
- TARGET_quantile=5
- TARGET_quantile=10
- TARGET_quantile=25
- TARGET_quantile=50
- TARGET_quantile=75
- TARGET_quantile=90
- TARGET_quantile=95
- TARGET_quantile=99

These features correspond to the quantiles at 1%, 5%, 10% … That is to say, we estimate that there is an X% chance that the prediction is below the quantile X.

In the case of a classification or a multiclass classification you will also obtain 2 new variables : « confidence » and « credibility ».

Confidence is an indicator of certainty that corresponds to a possible conflict between the majority class and the other classes. Thus, the higher the confidence, the more confident we are that the class with the highest probability is the right one. This makes sense in multiclass classifications. For example :

Class_1	Class_2	Class_3
0.9	0.05	0.05
0.5	0.45	0.05

In a case like this, « Class_1 » always holds the majority. In the first example, the algorithm clearly favors the majority class when distributing the probabilities : it will therefore have high confidence. Conversely, in the second example there is a likely conflict between « Class_1 » and « Class_2 ». Confidence will therefore be reduced.

Credibility is another indicator of certainty that specifies whether the prediction is based on a similar decision – and therefore close to an example of the training dataset – or whether it does not resemble any known case.

Let us imagine that we have a use case that consists in predicting house prices. Let's say we want to predict the price of an 80m2 house : in this case, there is nothing out of the ordinary, there are probably houses of 80m2 in the training dataset so the credibility will be high. Now, suppose we want to predict the price of a 700m2 loft.

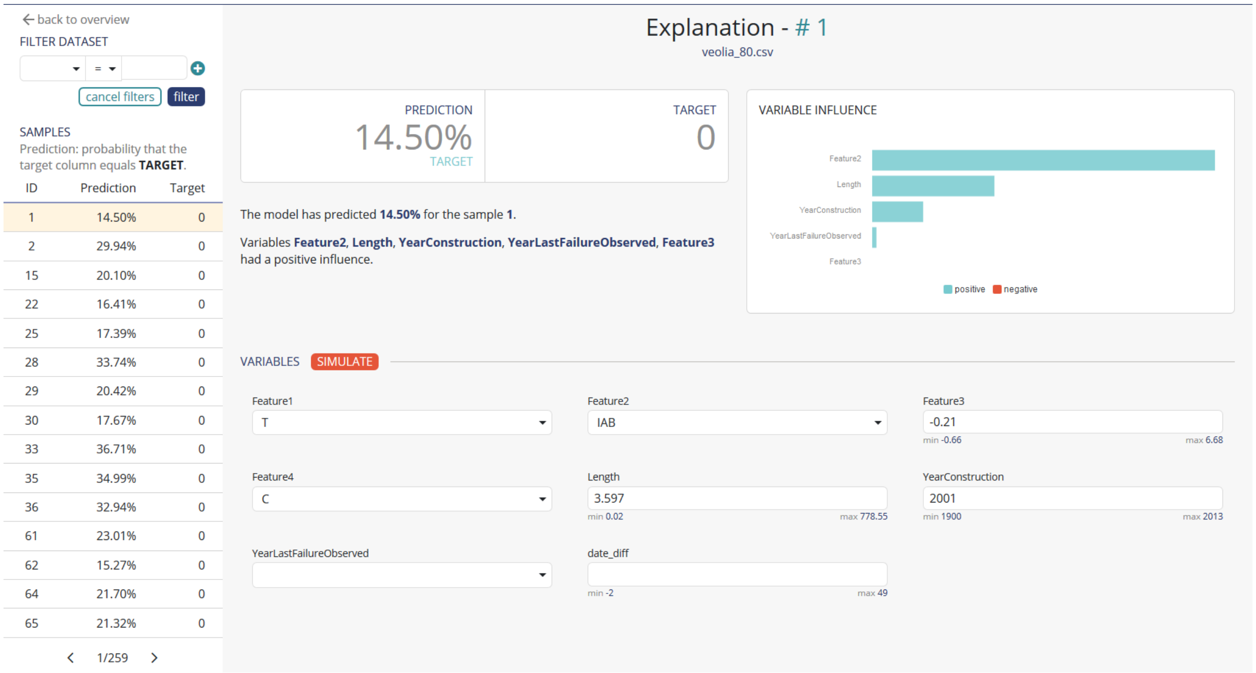
This type of housing being little or not represented, the algorithm will predict a price (probably very high) but its credibility will be low because this type of housing does not really resemble a typology present in the training dataset.

In addition to certainty indicators, Prevision.io allows users to understand the decision made by algorithms at the level of the statistical individual.

To obtain an explicability report, you must click the « explain » action link on the prediction you want to explain.

Warning : It may take some time to load this screen, especially if the number of individuals to explain is large.

You will then reach the next page :



SAMPLES
Prediction: probability that the target column equals **TARGET**.

ID	Prediction	Target
1	14.50%	0
2	29.94%	0
15	20.10%	0
22	16.41%	0
25	17.39%	0
28	33.74%	0
29	20.42%	0
30	17.67%	0
33	36.71%	0
35	34.99%	0
36	32.94%	0
61	23.01%	0
62	15.27%	0
64	21.70%	0
65	21.32%	0

‹ 1/259 ›

Explanation - # 1
veolia_80.csv

PREDICTION 14.50% TARGET

TARGET 0

The model has predicted **14.50%** for the sample **1**.

Variables **Feature2**, **Length**, **YearConstruction**, **YearLastFailureObserved**, **Feature3** had a positive influence.

VARIABLES SIMULATE

Feature1: T

Feature2: IAB

Feature3: -0.21 (min -0.66, max 6.68)

Feature4: C

Length: 3.597 (min 0.02, max 778.55)

YearConstruction: 2001 (min 1900, max 2013)

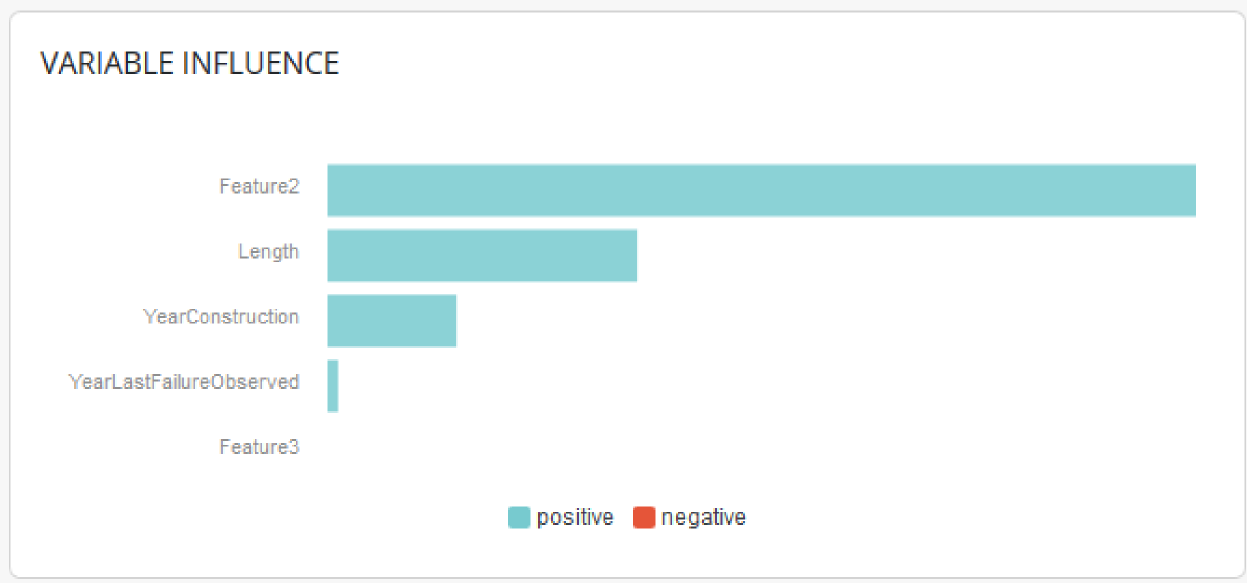
YearLastFailureObserved:

date_diff: (min -2, max 49)

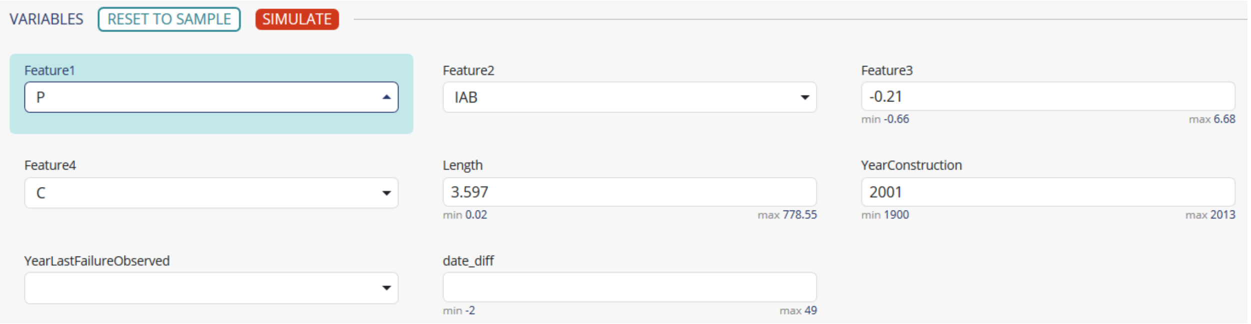
At the top left, you will find an interface that allows you to filter predictions according to the values of their variables. Up to 2 filters (and/or) can be used simultaneously.

Below, you will find a list of the predictions corresponding to the considered perimeter (with all default predictions). Each line of this table is clickable, the information will then be updated directly on the right side of the screen.

Each prediction is explained, the most important features for the prediction considered are explained in a sentence and also in the graph at the top right :



The lower part of the table features all the values of the features of the selected prediction. It is possible to modify them on the go and clicking on the « simulate » button to see a new prediction and explanation :



Feature1: P

Feature2: IAB

Feature3: -0.21 (min -0.66, max 6.68)

Feature4: C

Length: 3.597 (min 0.02, max 778.55)

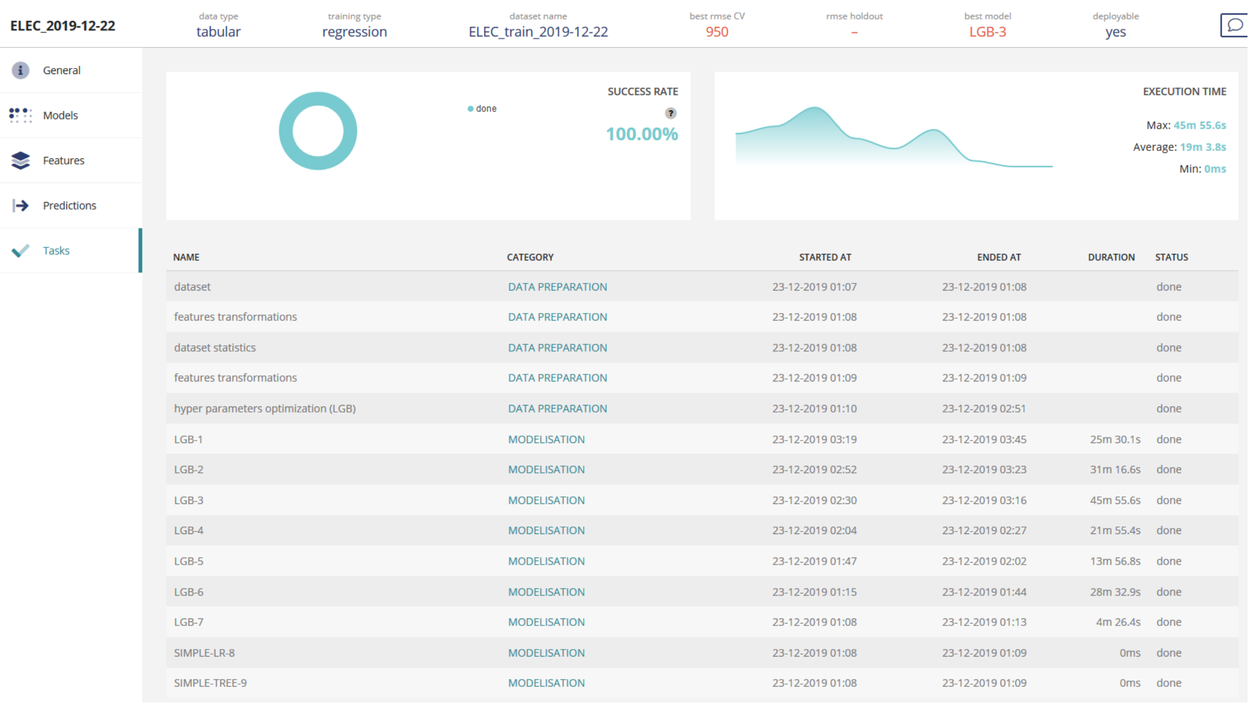
YearConstruction: 2001 (min 1900, max 2013)

YearLastFailureObserved:

date_diff: (min -2, max 49)

6.5 Tasks

Tasks linked to the usecase will be displayed here :



ELEC_2019-12-22

data type: tabular
training type: regression
dataset name: ELEC_train_2019-12-22
best rmse CV: 950
rmse holdout: -
best model: LGB-3
deployable: yes

SUCCESS RATE: 100.00%

EXECUTION TIME
Max: 45m 55.6s
Average: 19m 3.8s
Min: 0ms

NAME	CATEGORY	STARTED AT	ENDED AT	DURATION	STATUS
dataset	DATA PREPARATION	23-12-2019 01:07	23-12-2019 01:08		done
features transformations	DATA PREPARATION	23-12-2019 01:08	23-12-2019 01:08		done
dataset statistics	DATA PREPARATION	23-12-2019 01:08	23-12-2019 01:08		done
features transformations	DATA PREPARATION	23-12-2019 01:09	23-12-2019 01:09		done
hyper parameters optimization (LGB)	DATA PREPARATION	23-12-2019 01:10	23-12-2019 02:51		done
LGB-1	MODELISATION	23-12-2019 03:19	23-12-2019 03:45	25m 30.1s	done
LGB-2	MODELISATION	23-12-2019 02:52	23-12-2019 03:23	31m 16.6s	done
LGB-3	MODELISATION	23-12-2019 02:30	23-12-2019 03:16	45m 55.6s	done
LGB-4	MODELISATION	23-12-2019 02:04	23-12-2019 02:27	21m 55.4s	done
LGB-5	MODELISATION	23-12-2019 01:47	23-12-2019 02:02	13m 56.8s	done
LGB-6	MODELISATION	23-12-2019 01:15	23-12-2019 01:44	28m 32.9s	done
LGB-7	MODELISATION	23-12-2019 01:08	23-12-2019 01:13	4m 26.4s	done
SIMPLE-LR-8	MODELISATION	23-12-2019 01:08	23-12-2019 01:09	0ms	done
SIMPLE-TREE-9	MODELISATION	23-12-2019 01:08	23-12-2019 01:09	0ms	done

You'll find :

- Success rate of tasks (100% means all finished successfully)
- Execution time of tasks, ordered by tasks arrival
- A table with more detailed information :
  - Task name
  - Task category
  - Task start date
  - Task end date
  - Task duration
  - Task status

CHAPITRE 7

NLP

NLP treatments can be applied to textual features. To do that, your dataset has to contain some textual features. Then, during advanced configuration of your use case, you can apply some feature engineering using the "textual features" options.

Textual features : textual features are detected and automatically converted into numbers using 3 techniques :

Word embedding approach using Word2Vec/Glove. Words are projected to a dense vector space, where semantic distance between words are preserved : Prevision trains a word2vec algorithm on the actual input corpus, to generate their corresponding vectors. More information about Word embedding on https://en.wikipedia.org/wiki/Word_embedding

Sentence Embedding using Transformers approach. Prevision has integrated BERT-based transformers, as a pre-trained contextual model, that captures words relationships in a bidirectional way. BERT transformer makes it possible to generate more efficient vectors than word Embedding algorithms, it has a linguistic "representation" of its own. To make a text classification, we can use these vector representations as input to basic classifiers to make text classification. Bert (base/uncased) is used on english text and Multi Lingual (base/cased) is used on french text. More information about Transformers on https://en.wikipedia.org/wiki/Transformer_(machine_learning_model). The Python Package used is Sentence Transformers (https://www.sbert.net/docs/pretrained_models.html)

By default, only TF-IDF approach is used.

Advices :
— For better performance, it is advisable to check the word embedding and sentence embedding options.
— Checking its additional options will increase the time required for feature engineering, modeling, and prediction
You will find more information about NLP features and applications with prevision.io plateform with the following links :

https://www.youtube.com/watch?v=8Zu7mpdk528

https://prevision.io/fr/nouvelle-version-v10-13-performances-sur-le-textes-ameliorees-et-reconnaissance-dimage-en-temps-reel/

https://medium.com/prevision-io/automated-nlp-with-prevision-io-part1-naive-bayes-classifier-475fa8bd73de

CHAPITRE 8

Time series use cases

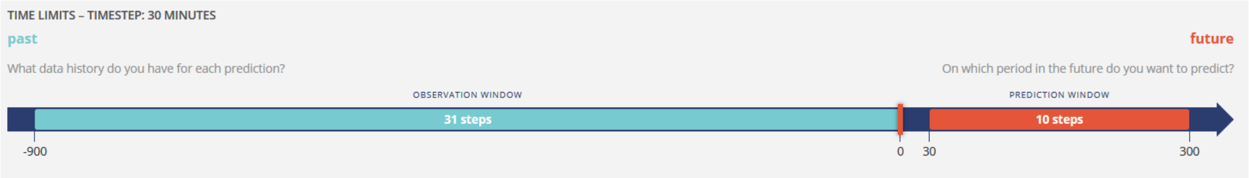
8.1 Analysis

8.1.1 General analysis

Since timeseries usecase are regressions, you'll find the same level of analytics than for its tabular counterpart.

8.1.2 Time gauge

We recall the selection criteria entered by the user on the time gauge :



TIME LIMITS – TIMESTEP: 30 MINUTES

past

What data history do you have for each prediction?

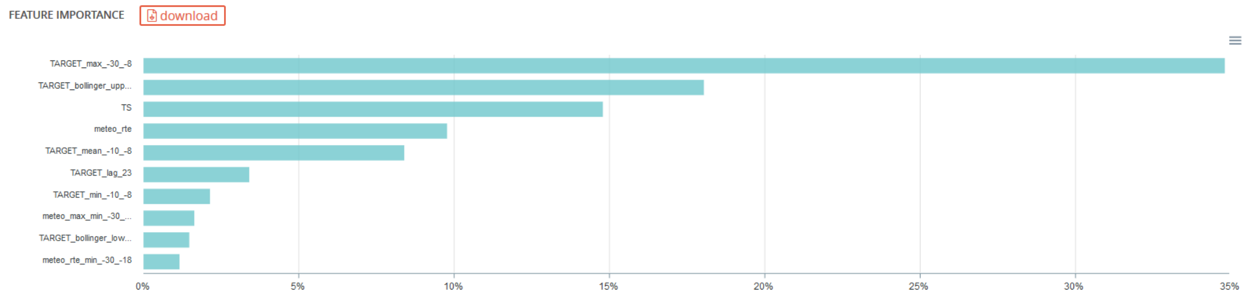
future

On which period in the future do you want to predict?

8.1.3 Feature importance

The goal of the timeserie modelisation is to find automatically new temporal features that will increase the predictive power of the model. Temporel features will be created based on statistical signifiance such as autocorrelation function (ACF), partial autocorrelation function (PACF), correlation with the TARGET, . . .

Created features can be found in the feature importance :



They are constructed with the name of the original feature, followed by some moving agregate functions :

- featurename_lag_X = `lag` (offset) of X timestep of `featurename`
- featurename_min_a_b = `minimum` of `featurename` between a and b timestep
- featurename_max_a_b = `maximum` of `featurename` between a and b timestep
- featurename_mean_a_b = `mean` (moving average) of `featurename` between a and b timestep
- featurename_bollinger_upper_a_b = `upper bound of bolliger` (~ moving average + sd) of `featurename` between a and b timestep
- featurename_bollinger_lower_a_b = `lower bound of bollinger` (~ moving average - sd) of `featurename` between a and b timestep

Please keep in mind that `featurename` can be the `TARGET` or any feature present in the dataset.

8.2 Predictions

When forecasting, it is necessary to send a historical dataset of at least the same length as the interval between the 2 boundaries of the historical window. This set will be completely filled with the actual data (including the target) and will be completed with the data to be forecasted :

- The target that will be absent -> Prevision.io will detect the period to be predicted from the moment the target ceases to be known
- The data will be filled in a priori
- Non a priori data will be missing

The output of this step will be a file (time, value) filled over the forecasted period. In addition, if the historical period is longer than the length of the window, forecasts will be made using this data and will allow a test score to be calculated directly in the application.

8.3 In case of problems

8.3.1 During training

Given the complexity of time series modeling, it is essential that the data set respects the following constraints during the learning phase :

- Check that the target is numeric
- Check the constraints on the temporal window and the history window
- Check that a time column is filled in in ISO 8601 format (or in classic formats, such as DD/MM/YYYYY or DD-MM-YYYY hh :mm for example)
- Check that the time spacing is consistent for at least 80% of the data (e.g. : You send a series of one day at the hourly step. If more than 5 data are missing, the calculation will not be successful)
- Check, when there is a group, that the columns designated as such identify a unique time series (i.e. a maximum value on a timestamp)
- Check, when there is a group, that the time step is consistent between the groups

— Check that the time steps and the number of missing data respect the rules mentioned above, including all intersections induced by the possible presence of groups

Remarks :

— Evaluation is performed on a time split cross validation
— In case of multiple lines on the same timestamp, only the first event is kept
— In case of missing timestamps, the last known value is propagated to the next known timestamp
— Each group must contain at least 3 observations. If this is not the case, the group will be deleted from the dataset

8.3.2 During forecast

I have a file containing 0 forecasts

Make sure you have provided a dataset with a missing target starting from a given timestamp. If the target column is still filled, we cannot extend the forecast, especially if your use case contains a priori groups and features.

The prediction returns inconsistent results

Check that the a priori features in particular are correctly filled in for the values to be forecasted.

Check that all the labelled data corresponding to the history window is filled in. Missing data will be imputed as equal to the mean of the target, which can screw results.

Check that the difference between the time of training and the prediction is not too high. Time series may require more frequent re-training than other use cases because of natural target drift.

The prediction returns an error

In general, check that you provide a sufficient history consistent with the definition of your use case.

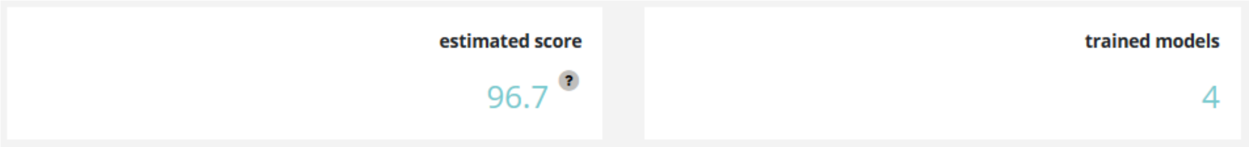
If your dataset contains groups :

— Check that the groups are temporally consistent, i.e. for each group there are as many time steps as the others
— Check that no new groups appear at the time of the forecast

CHAPITRE 9

Object detector

9.1 Performance



estimated score

96.7 ?

trained models

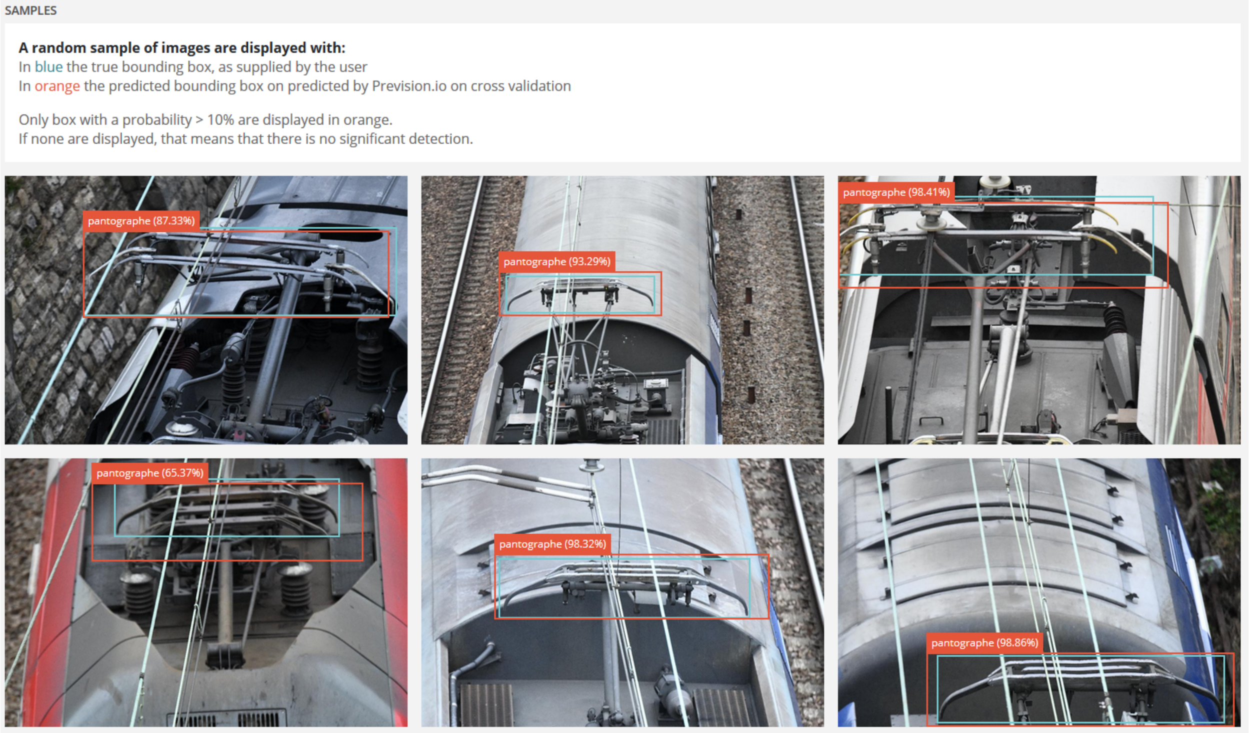
4

Performance of object detector use case is computed using mAP (mean Average Precision) : https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

This metric is computed on up to 5 differents models trained on the Data Set.

Please note that the performance spread can be important between first and last models.

9.2 Model analysis



SAMPLES

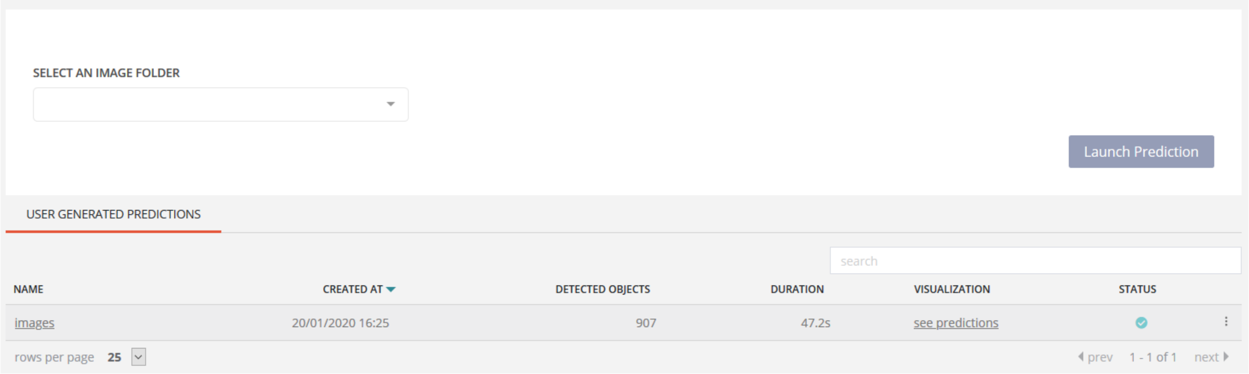
A random sample of images are displayed with:
In blue the true bounding box, as supplied by the user
In orange the predicted bounding box on predicted by Prevision.io on cross validation

Only box with a probability > 10% are displayed in orange.
If none are displayed, that means that there is no significant detection.

A random sample of images are displayed with :
— In blue the true bounding box, as supplied by the user
— In orange the predicted bounding box on predicted by Prevision.io on cross validation
Only box with a probability > 10% are displayed in orange.

If none are display, that means that there is no significant detection.

9.3 Prediction

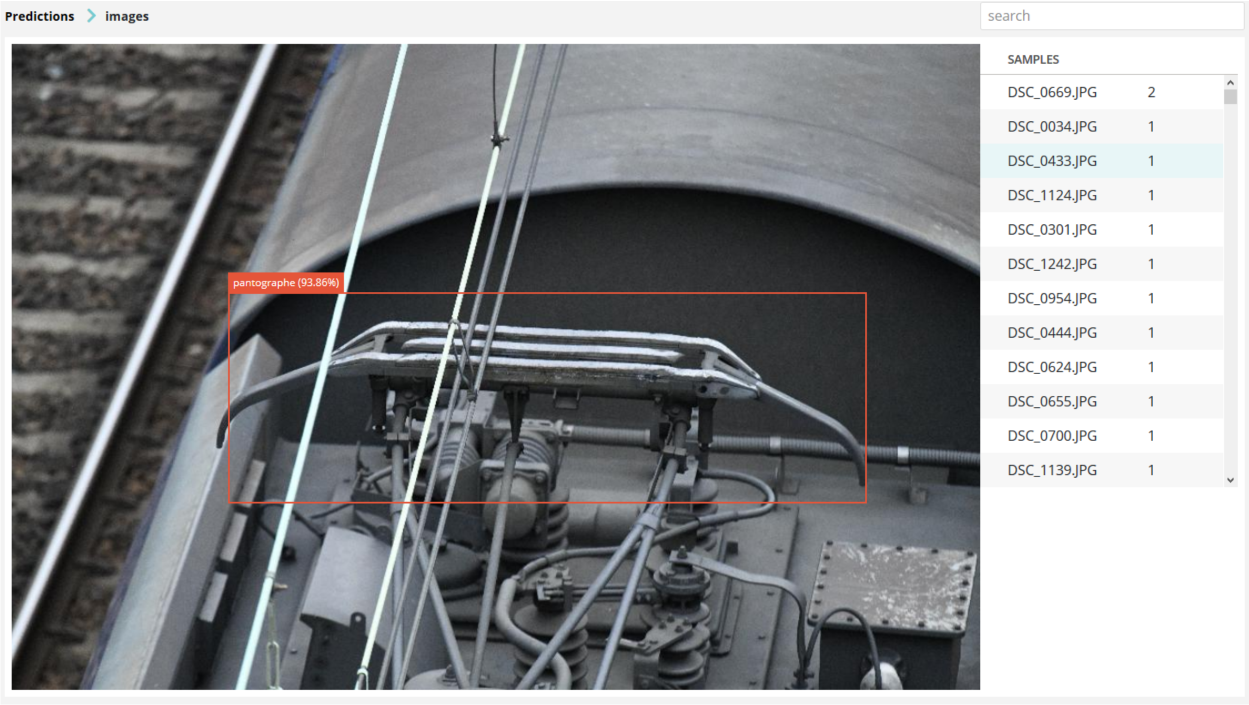


SELECT AN IMAGE FOLDER

NAME	CREATED AT	DETECTED OBJECTS	DURATION	VISUALIZATION	STATUS
images	20/01/2020 16:25	907	47.2s	see predictions	✓

Prediction works like other image Data Set.

Once done, you can directly check results with probability associated with each bounding box by clicking the see predictions link :



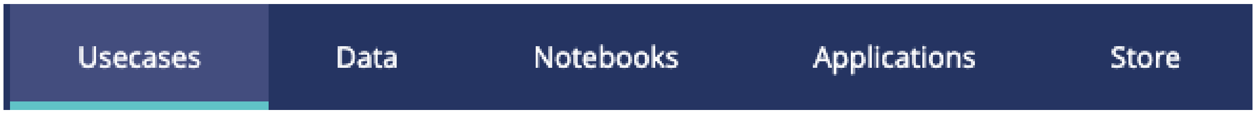
SAMPLES	
DSC_0669.JPG	2
DSC_0034.JPG	1
DSC_0433.JPG	1
DSC_1124.JPG	1
DSC_0301.JPG	1
DSC_1242.JPG	1
DSC_0954.JPG	1
DSC_0444.JPG	1
DSC_0624.JPG	1
DSC_0655.JPG	1
DSC_0700.JPG	1
DSC_1139.JPG	1

CHAPITRE 10

Notebooks

Prevision.io offers various tools to enable data science use cases to be carried out. Among these tools, there are notebooks and production tools.

These are accessible by clicking on the tab below :



You will then be directed to the following screen :

10.1 Development environment

10.1.1 JUPYTERLAB

For Python users, a JUPYTERLAB environment (https://github.com/jupyterlab/jupyterlab) is available in Prevision.io

Note that a set of packages is preinstalled on your instance (list : https://previsionio.readthedocs.io/fr/latest/_static/ressources/packages_python.txt), particularly the previsionio package that encapsulates the functions using the tool's native APIs. Package documentation link : https://prevision-python.readthedocs.io/en/latest/

10.1.2 R STUDIO

For R users, a R STUDIO environment (https://www.rstudio.com) is available in Prevision.io

Note that a set of packages is preinstalled on your instance (list : https://previsionio.readthedocs.io/fr/latest/_static/ressources/packages_R.txt), particularly the previsionio package that encapsulates the functions that use the tool's native APIs. Package documentation link : https://previsionio.readthedocs.io/fr/latest/_static/ressources/previsionio.pdf

CHAPITRE 11

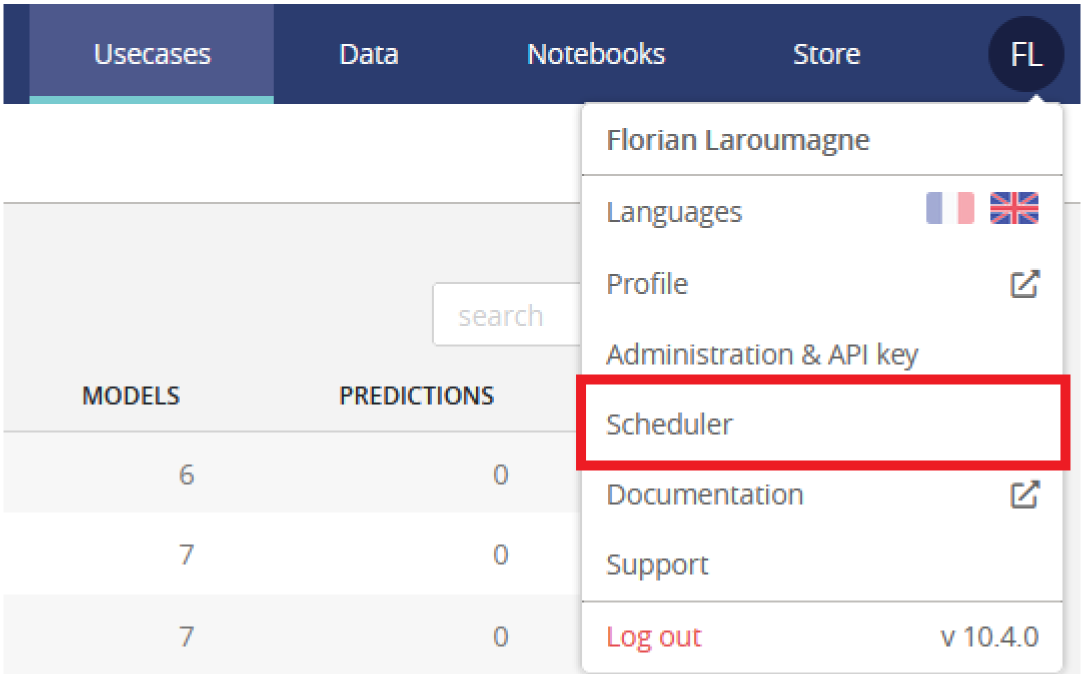
Scheduler

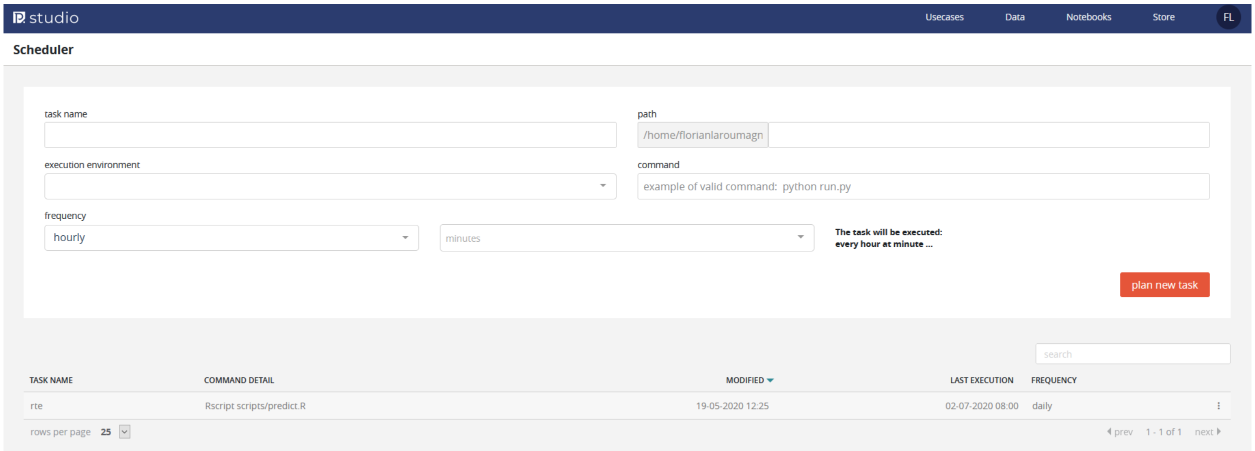
Prevision.io allows you to schedule tasks, based on user-developed R or Python scripts. These scripts can be used, for example, to create models and make predictions.

The scheduler's role is to automate the recurrence of these actions (e. g. making predictions every day, training every month, etc.)

11.1 Table of scheduled tasks

The scheduled tasks table allows you to view all the created tasks. You can access the table by clicking on the most up-right button then scheduler :





Scheduler

task name

path: /home/florianlaroumagn

execution environment

command: example of valid command: python run.py

frequency: hourly

minutes

The task will be executed:
every hour at minute ...

TASK NAME	COMMAND DETAIL	MODIFIED	LAST EXECUTION	FREQUENCY
rte	Rscript scripts/predict.R	19-05-2020 12:25	02-07-2020 08:00	daily

rows per page 25

11.1.1 Creating a task

To create a new task, you must fill information at the top part of the screen, such as :

- The name of the task
- The path to the related script
- The execution environnement among
  - Python 3.6.5
  - R 3.6.1
- The execution command, typically
  - For python *python script.py*
  - For R *Rscript script.R*
- The frequency of execution

11.1.2 Task table

All tasks created will be displayed in the bottom table with the according informations. You'll see :

- The task name
- The execution command
- The created / modified date
- The last execution
- The frequency
- A contextual menu allowing you to
    - Force run the task (will trigger a new execution on it)
    - Delete the task

Remarks :

- Creating the task can take a few minutes. Avoid creating a task that must start almost immediately
- The execution time to be filled in is in UTC (therefore, there will be a 1-hour time difference in winter and a 2-hour difference in summer)

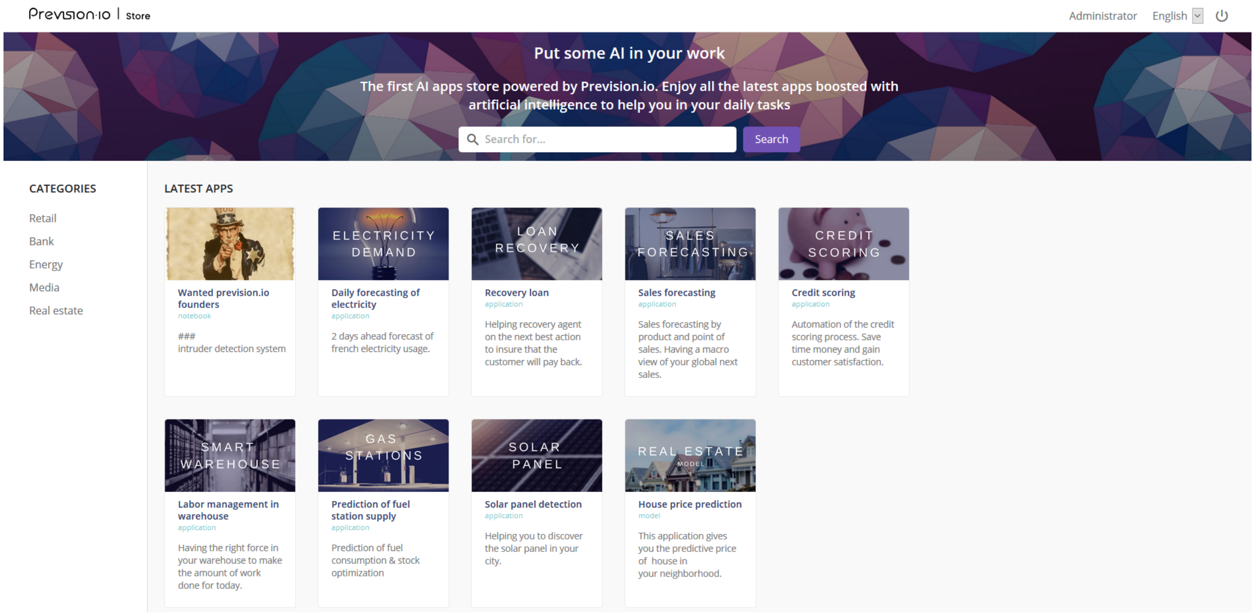
CHAPITRE 12

Store

The store of your instance is reachable at : https://xxx.prevision.io/store (xxx being the name of your instance)

It can host :
- Models trained in the STUDIO
- Python notebooks (More information : https://jupyter.org/)
- Packaged Dash applications (More information : https://plot.ly/products/dash/)
- Packaged Shiny applications (More information : https://shiny.rstudio.com/)

Each of these applications is protected by rights that were specified at the time of deployment. Thus, each end user only sees the applications to which he or she is entitled.



Put some AI in your work

The first AI apps store powered by Prevision.io. Enjoy all the latest apps boosted with artificial intelligence to help you in your daily tasks

Wanted prevision.io founders
notebook
###
intruder detection system

Daily forecasting of electricity
application
2 days ahead forecast of french electricity usage.

Recovery loan
application
Helping recovery agent on the next best action to insure that the customer will pay back.

Sales forecasting
application
Sales forecasting by product and point of sales. Having a macro view of your global next sales.

Credit scoring
application
Automation of the credit scoring process. Save time money and gain customer satisfaction.

Labor management in warehouse
application
Having the right force in your warehouse to make the amount of work done for today.

Prediction of fuel station supply
application
Prediction of fuel consumption & stock optimization

Solar panel detection
application
Helping you to discover the solar panel in your city.

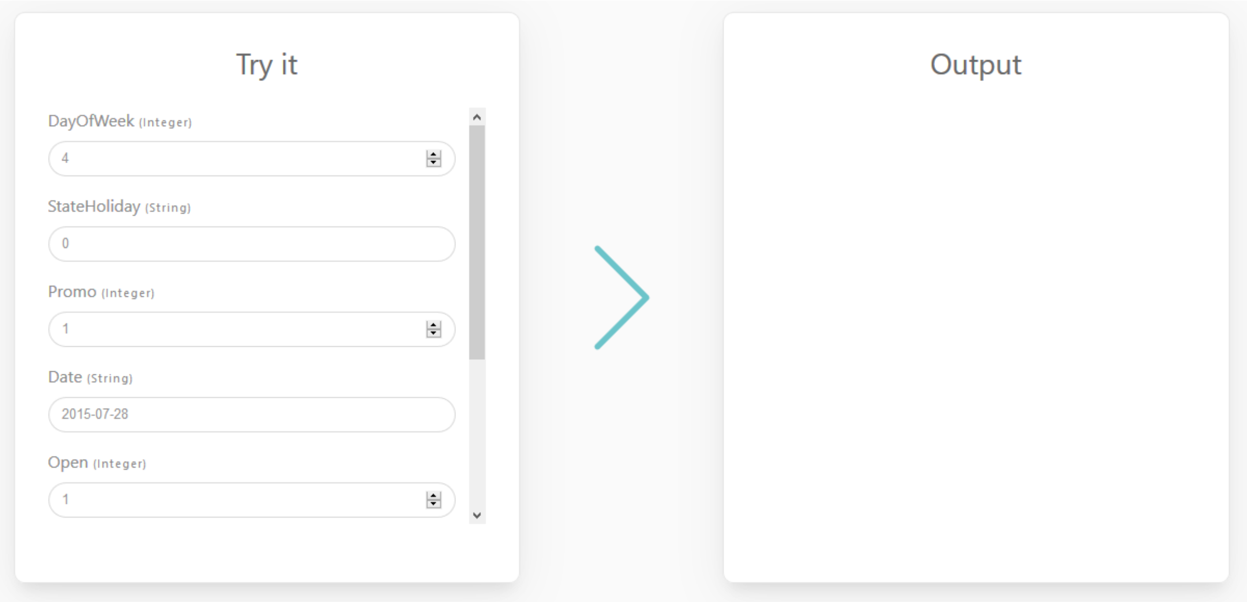
House price prediction
model
This application gives you the predictive price of house in your neighborhood.

Each tile represents an application. The name and description displayed are the ones you entered when you created the application. To access it, simply click on the name of the application.

12.1 Accessing deployed content

12.1.1 Model

If you access a model, you will have the following interface :



DayOfWeek (Integer)
4

StateHoliday (String)
0

Promo (Integer)
1

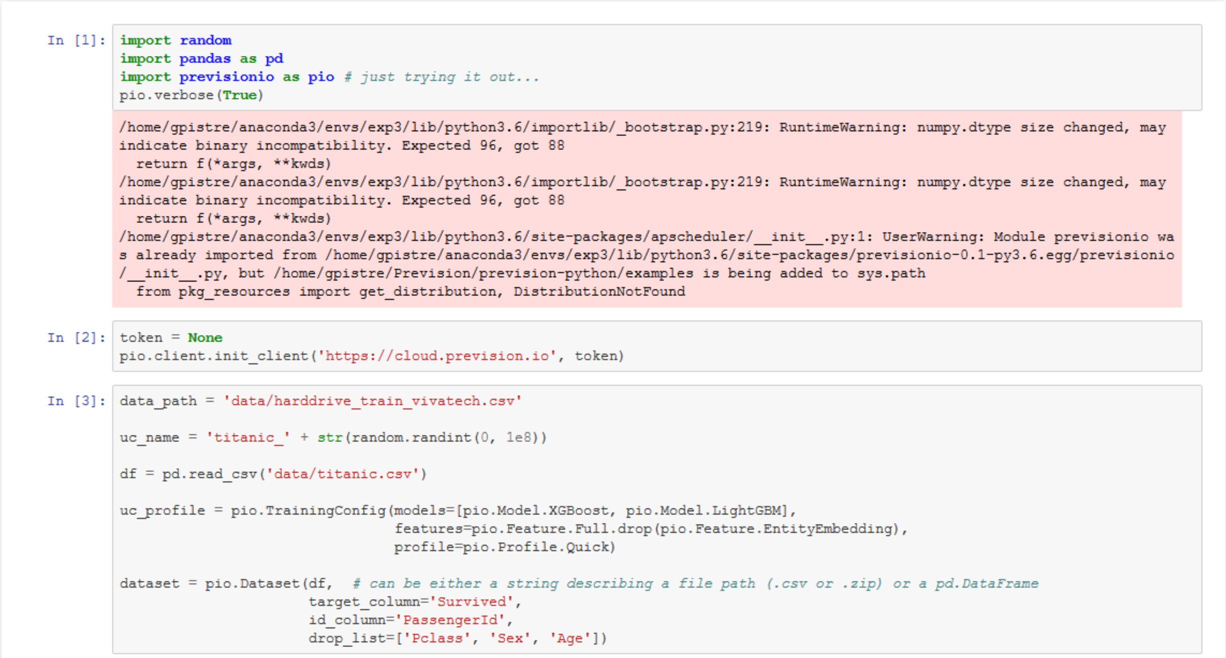
Date (String)
2015-07-28

Open (Integer)
1

In the left part, you can select all the values of the features of your use case. Once you have chosen the values, you can start a prediction by clicking on the « Submit » button. The prediction will then be displayed on the right-hand side of the screen.

12.1.2 Notebook

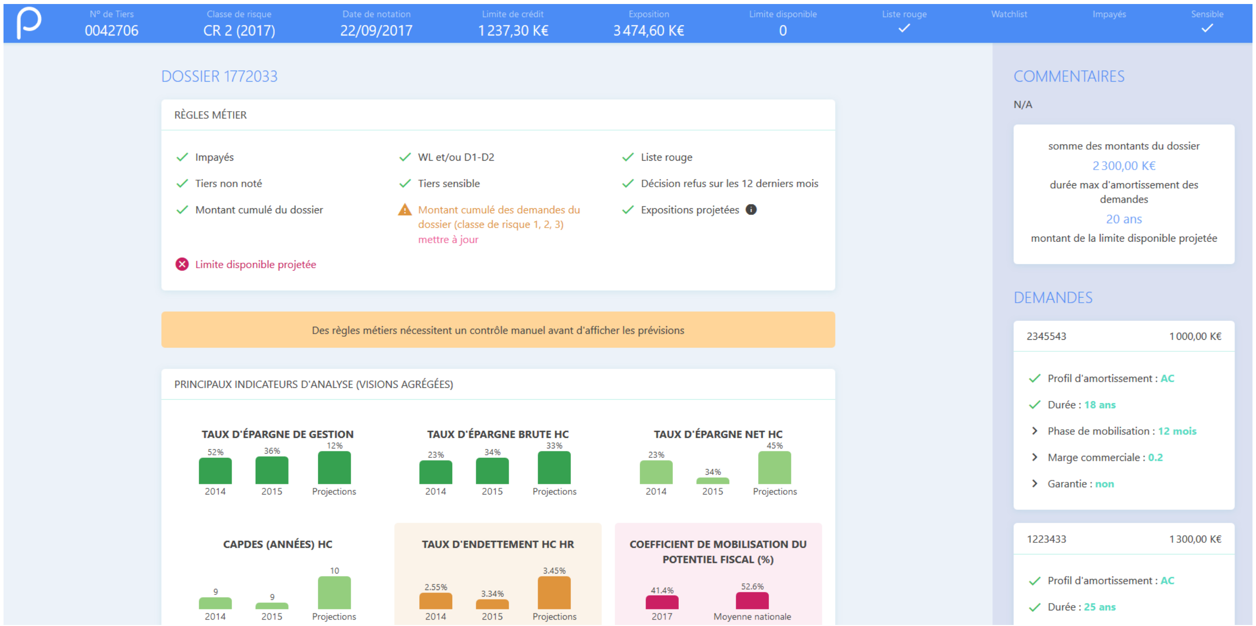
If you access a notebook, you will have the following interface :



```
In [1]: import random
        import pandas as pd
        import previsionio as pio # just trying it out...
        pio.verbose(True)

/home/gpistre/anaconda3/envs/exp3/lib/python3.6/importlib/_bootstrap.py:219: RuntimeWarning: numpy.dtype size changed, may
indicate binary incompatibility. Expected 96, got 88
  return f(*args, **kwds)
/home/gpistre/anaconda3/envs/exp3/lib/python3.6/importlib/_bootstrap.py:219: RuntimeWarning: numpy.dtype size changed, may
indicate binary incompatibility. Expected 96, got 88
  return f(*args, **kwds)
/home/gpistre/anaconda3/envs/exp3/lib/python3.6/site-packages/apscheduler/__init__.py:1: UserWarning: Module previsionio wa
s already imported from /home/gpistre/anaconda3/envs/exp3/lib/python3.6/site-packages/previsionio-0.1-py3.6.egg/previsionio
/__init__.py, but /home/gpistre/Prevision/prevision-python/examples is being added to sys.path
  from pkg_resources import get_distribution, DistributionNotFound

In [2]: token = None
        pio.client.init_client('https://cloud.prevision.io', token)

In [3]: data_path = 'data/harddrive_train_vivatech.csv'

        uc_name = 'titanic_' + str(random.randint(0, 1e8))

        df = pd.read_csv('data/titanic.csv')

        uc_profile = pio.TrainingConfig(models=[pio.Model.XGBoost, pio.Model.LightGBM],
                                        features=pio.Feature.Full.drop(pio.Feature.EntityEmbedding),
                                        profile=pio.Profile.Quick)

        dataset = pio.Dataset(df,  # can be either a string describing a file path (.csv or .zip) or a pd.DataFrame
                              target_column='Survived',
                              id_column='PassengerId',
                              drop_list=['Pclass', 'Sex', 'Age'])
```

12.1.3 Packaged application

If you access a dashboard, you have access to the interface developed by the studio users. It will therefore be totally personalized. Here are 2 examples of a dashboards :



Nº de Tiers	Classe de risque	Date de notation	Limite de crédit	Exposition	Limite disponible	Liste rouge	Watchlist	Impayés	Sensible
0042706	CR 2 (2017)	22/09/2017	1 237,30 K€	3 474,60 K€	0	✓			✓

DOSSIER 1772033

RÈGLES MÉTIER

✓ Impayés
✓ Tiers non noté
✓ Montant cumulé du dossier
✓ WL et/ou D1-D2
✓ Tiers sensible
⚠ Montant cumulé des demandes du dossier (classe de risque 1, 2, 3)
mettre à jour
✓ Liste rouge
✓ Décision refus sur les 12 derniers mois
✓ Expositions projetées

☒ Limite disponible projetée

Des règles métiers nécessitent un contrôle manuel avant d'afficher les prévisions

PRINCIPAUX INDICATEURS D'ANALYSE (VISIONS AGRÉGÉES)

COMMENTAIRES

N/A

somme des montants du dossier
2 300,00 K€
durée max d'amortissement des demandes
20 ans
montant de la limite disponible projetée

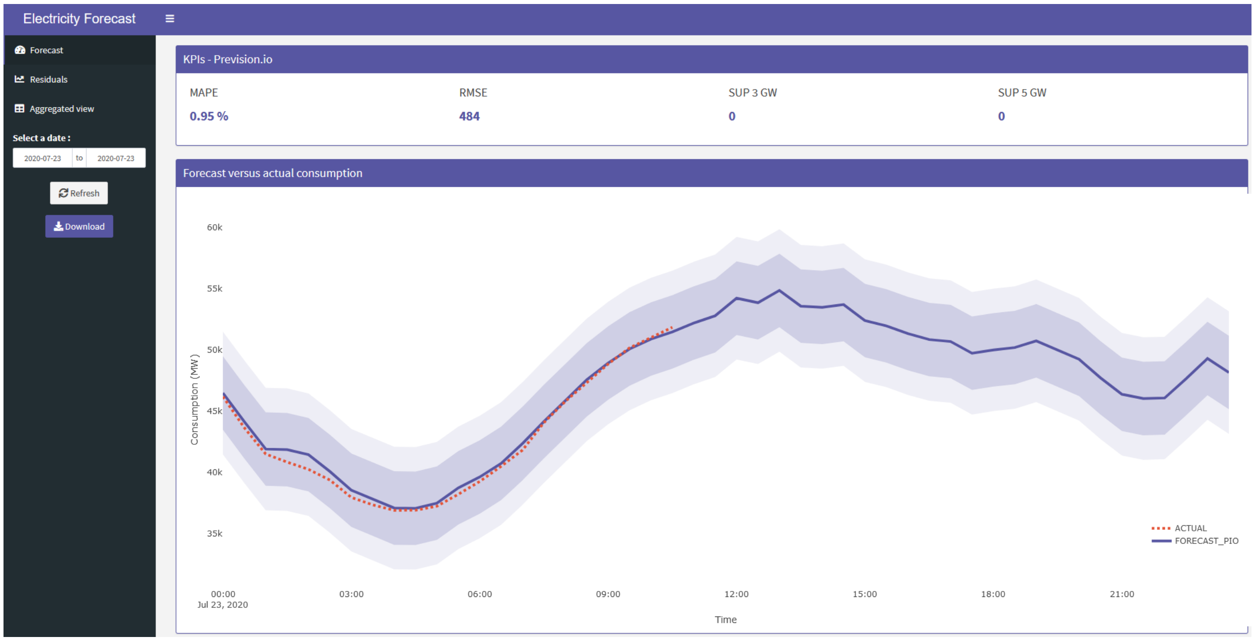
DEMANDES

2345543 1 000,00 K€

✓ Profil d'amortissement : AC
✓ Durée : 18 ans
› Phase de mobilisation : 12 mois
› Marge commerciale : 0.2
› Garantie : non

1223433 1 300,00 K€

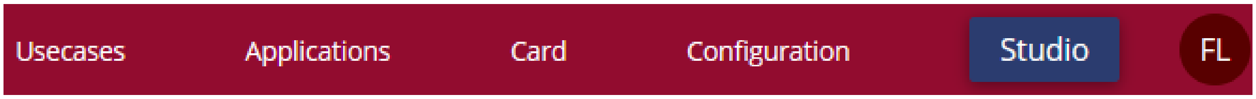
✓ Profil d'amortissement : AC
✓ Durée : 25 ans



MAPE	RMSE	SUP 3 GW	SUP 5 GW
0.95 %	484	0	0

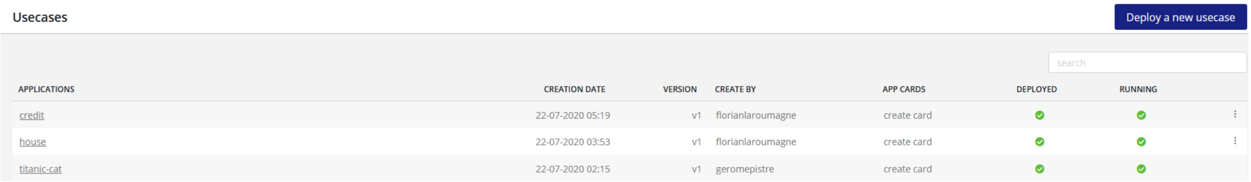
12.2 Deploying new content

To deploy new content in the store, you need to have administrators privileges. If this is the case, the store's header will look like :



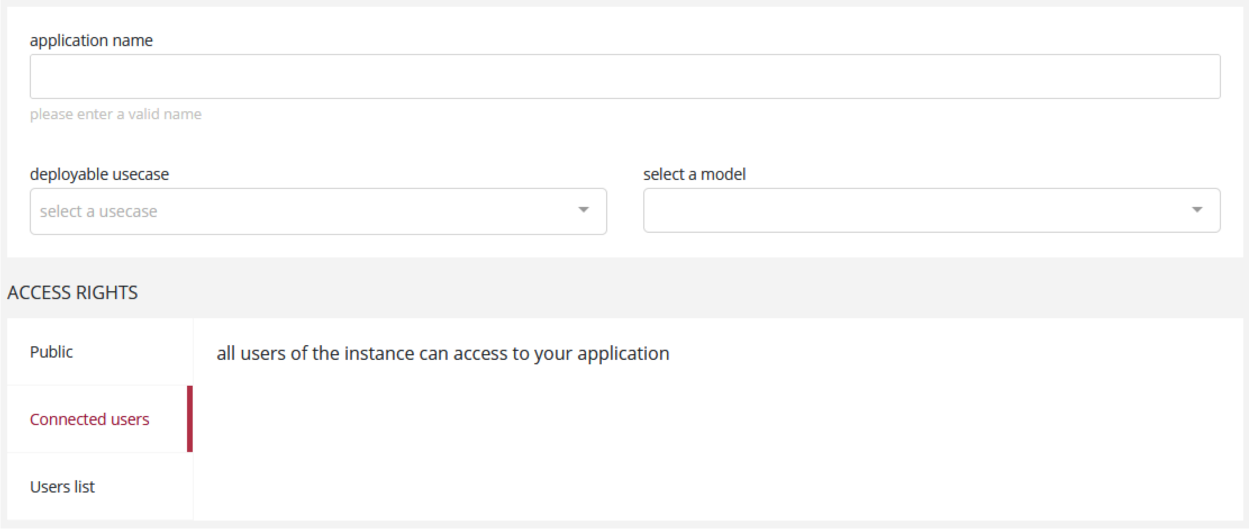
12.2.1 Deploying new usecase

To deploy a new usecase, click on the `Usecases` part in the header. By then, you'll see every usecases already deployed by every store admin :



APPLICATIONS	CREATION DATE	VERSION	CREATE BY	APP CARDS	DEPLOYED	RUNNING	
credit	22-07-2020 05:19	v1	florianlaroumagne	create card	✓	✓	⋮
house	22-07-2020 03:53	v1	florianlaroumagne	create card	✓	✓	⋮
titanic-cat	22-07-2020 02:15	v1	geromepistre	create card	✓	✓	

As you can see, a deployed usecase has :
— A name
— A creation date
— A version number
— A creator
— A card (optionnal)
— A deployed status
— A running status
— A contextual menu (accessible only to your apps), allowing you to remove it
To deploy a new usecase, click on the top right button `Deploy a new usecase`. You'll land on the following form :



application name

please enter a valid name

deployable usecase: select a usecase

select a model

ACCESS RIGHTS

Public

Connected users

Users list

all users of the instance can access to your application

Here, you are asked to provide :

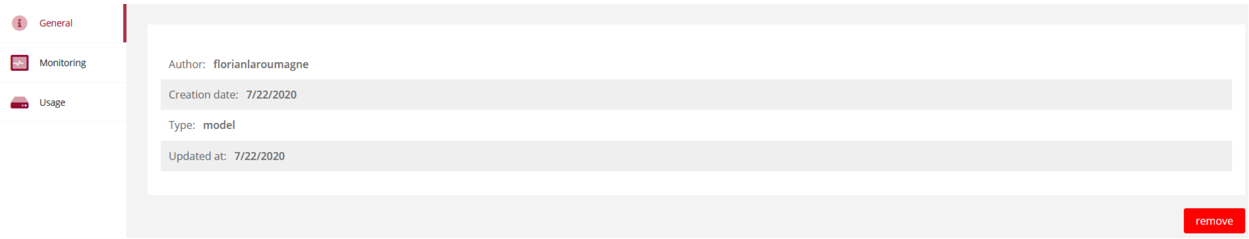
- A name for you application
- A deployable usecase comming from Prevision.io's STUDIO
- A model related to the selected usecase. By default the « recommanded » one from the STUDIO will be displayed in the combobox but you are free to select the one you want
- Rights to give to the application, among :
  - Public : Every people with the link to the application will be able to access it
  - Connected users : Every users of the instance will be able to access it once logged in
  - Users list : Only specified users will be able to access it once logged in

Afterwards, a click on the `Deploy` button located at the top-right of the screen will start the deploying process. It will typically take less than 1 minute to deploy a new usecase.

Monitoring a deployed usecase

To monitor a deployed usecase, just click on its name in the listing.

General informations



Author: florianlaroumagne

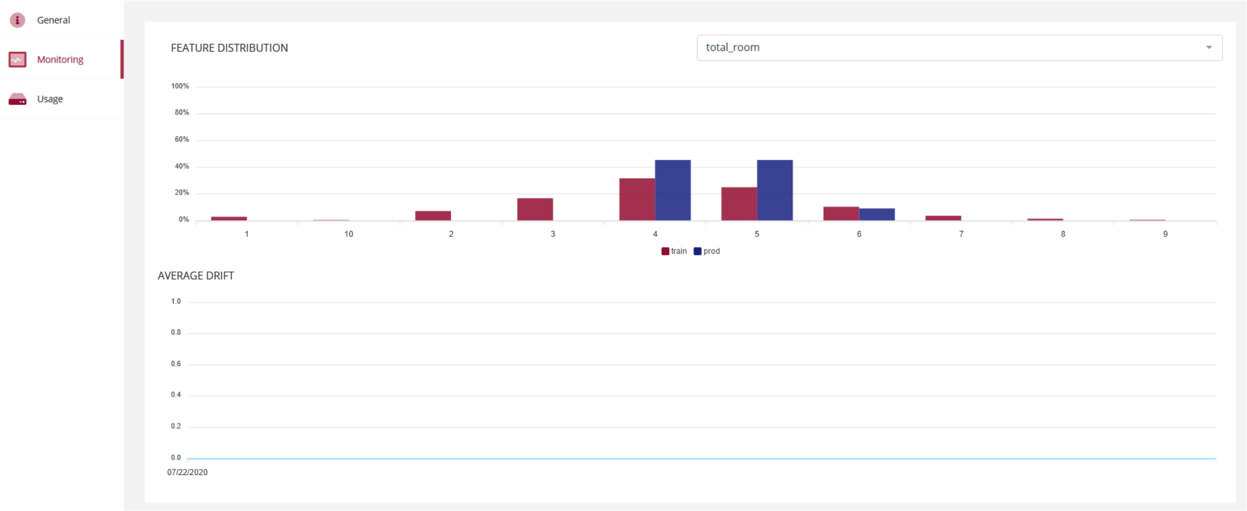
Creation date: 7/22/2020

Type: model

Updated at: 7/22/2020

Here you'll find general information, such as the deployment date or the owner of the usecase.

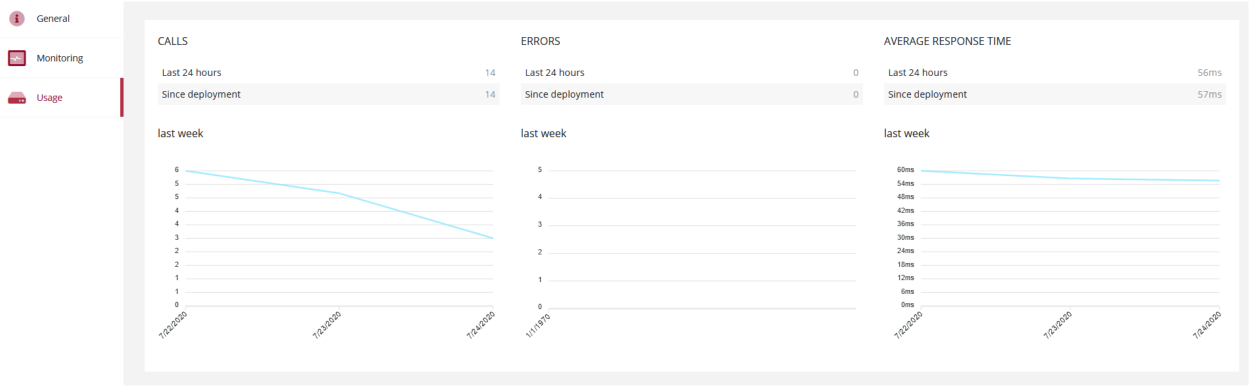
Monitoring (datascience)



In this screen, you can access distribution of every features and predictions related to the usecase. You can compare distribution of training data versus data fed to the model.

On the bottom part of the screen, you'll see information about average drift. It will tell you if globally, distribution of features are still aligned between training data and data fed to the model. Please consider retraining your model when the average drift start to raise.

Usage



CALLS

Last 24 hours	14
Since deployment	14

ERRORS

Last 24 hours	0
Since deployment	0

AVERAGE RESPONSE TIME

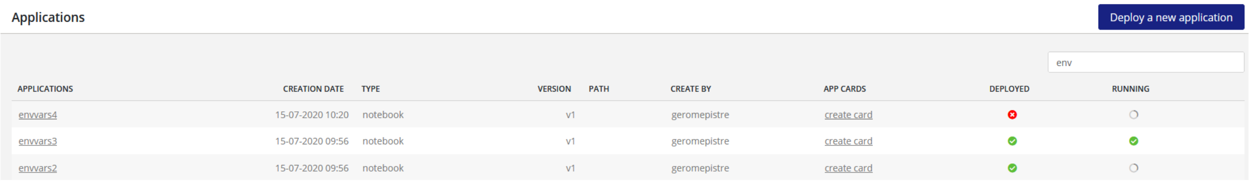
Last 24 hours	56ms
Since deployment	57ms

In this screen, you can access information about the usage of the model such as :

- Number of calls since model deployment
- Number of calls for the last 24 hours
- A plot of number of calls last week, grouped by day
- Number of errors since model deployment
- Number of error for the last 24 hours
- A plot of number of errors last week, grouped by day
- Average response time (ms) since model deployment
- Average response time (ms) for the last 24 hours
- A plot of average response time (ms) last week, grouped by day

12.2.2 Deploying a new application

To deploy a new application, click on the `Applications` part in the header. By then, you'll see every application already deployed by every store admin :



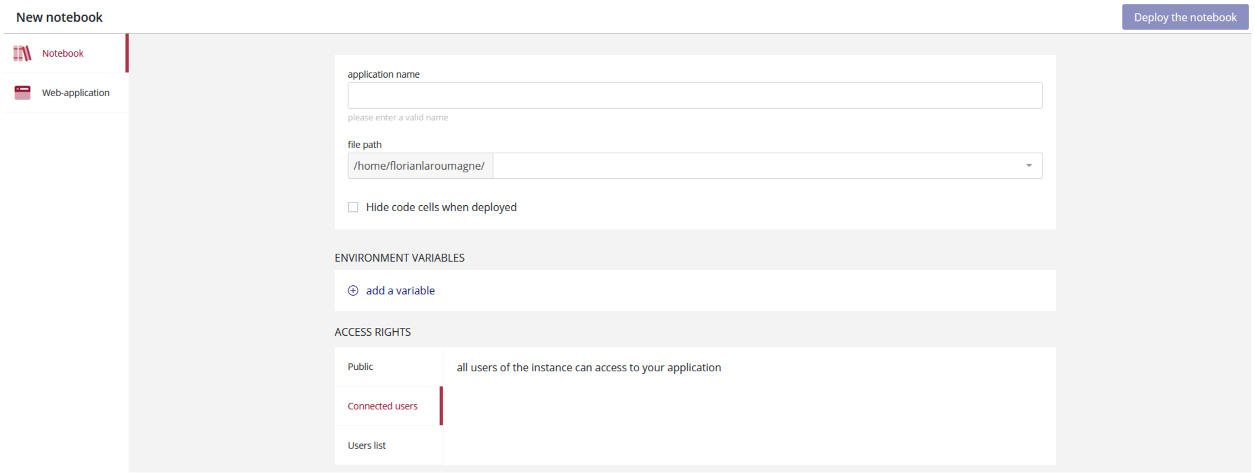
Applications

APPLICATIONS	CREATION DATE	TYPE	VERSION	PATH	CREATE BY	APP CARDS	DEPLOYED	RUNNING
envvars4	15-07-2020 10:20	notebook	v1		geromepistre	create card	✗	○
envvars3	15-07-2020 09:56	notebook	v1		geromepistre	create card	✓	✓
envvars2	15-07-2020 09:56	notebook	v1		geromepistre	create card	✓	○

To deploy a new application, click on the top right button `Deploy a new application`.

Deploying a notebook

You can chose to deploy a notebook on the most left part of the screen. If selected, you'll access the following form :



New notebook

application name

please enter a valid name

file path

/home/florianlaroumagne/

□ Hide code cells when deployed

ENVIRONMENT VARIABLES

ACCESS RIGHTS

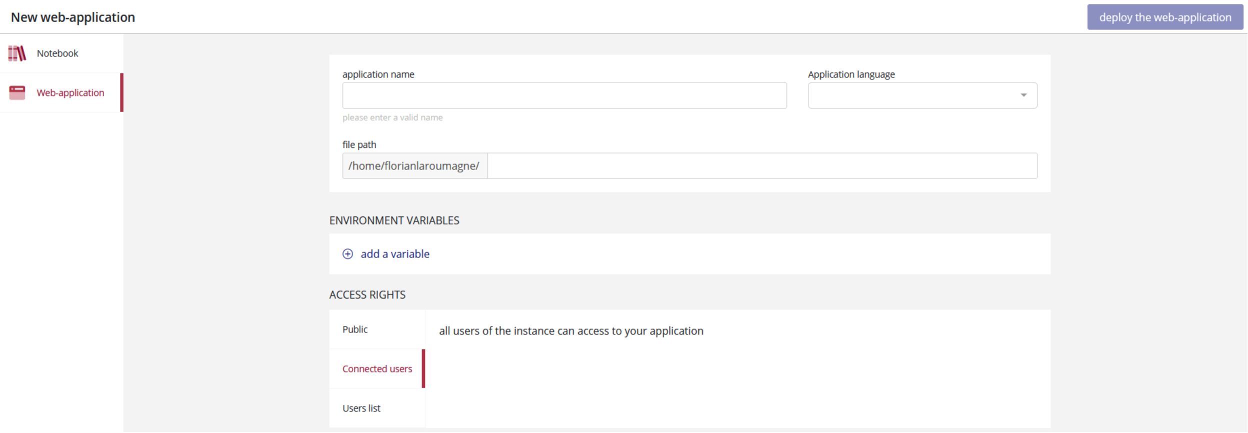
all users of the instance can access to your application

Here, you are asked to provide :
- A name for you application
- The path to the notebook located in your personnal IDE's folder (/ !only .ipynb are supported for now)
- The possibility to hide code cells when deployed (optionnal)
- Environnement variables (optionnal)
- Rights to give to the application, among :
    - Public : Every people with the link to the application will be able to access it
    - Connected users : Every users of the instance will be able to access it once logged in
    - Users list : Only specified users will be able to access it once logged in

Afterwards, a click on the `Deploy the notebook` button located at the top-right of the screen will start the deploying process. It will typically take less than 1 minute to deploy a new notebook.

Deploying a web application

You can chose to deploy a web application on the most left part of the screen. If selected, you'll access the following form :



New web-application

deploy the web-application

application name

please enter a valid name

Application language

file path

/home/florianlaroumagne/

ENVIRONMENT VARIABLES

⊕ add a variable

ACCESS RIGHTS

Public

Connected users

Users list

all users of the instance can access to your application

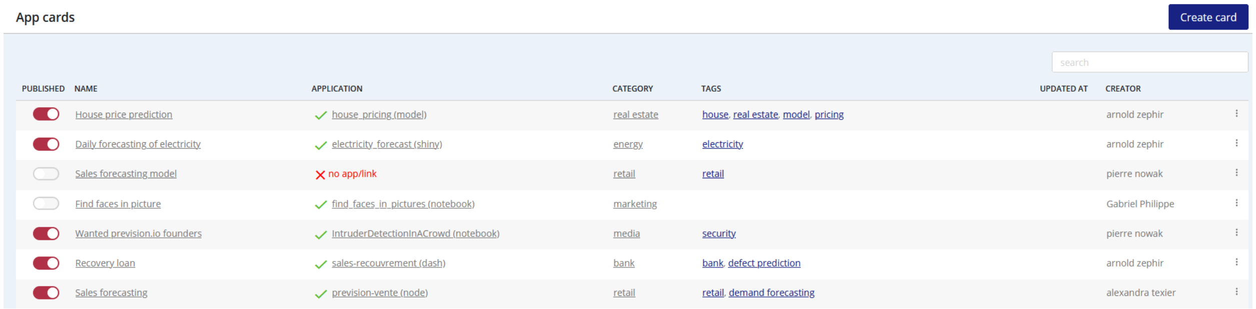
Here, you are asked to provide :

- A name for you application
- An application language (among R or PYTHON)
- The path to the application located in your personnal IDE's folder
- Environnement variables (optionnal)
- Rights to give to the application, among :
  - Public : Every people with the link to the application will be able to access it
  - Connected users : Every users of the instance will be able to access it once logged in
  - Users list : Only specified users will be able to access it once logged in

Afterwards, a click on the `Deploy the web-application` button located at the top-right of the screen will start the deploying process. It will typically take less than 5 minutes to deploy a new web-application.

12.2.3 Adding a card to your application

If you want that your applications appears on the front page of the store, you need to attach a `card` to it. To do so click on the `Card` part in the STORE's header. By then, you'll see every cards already deployed by every store admin :



App cards

Create card

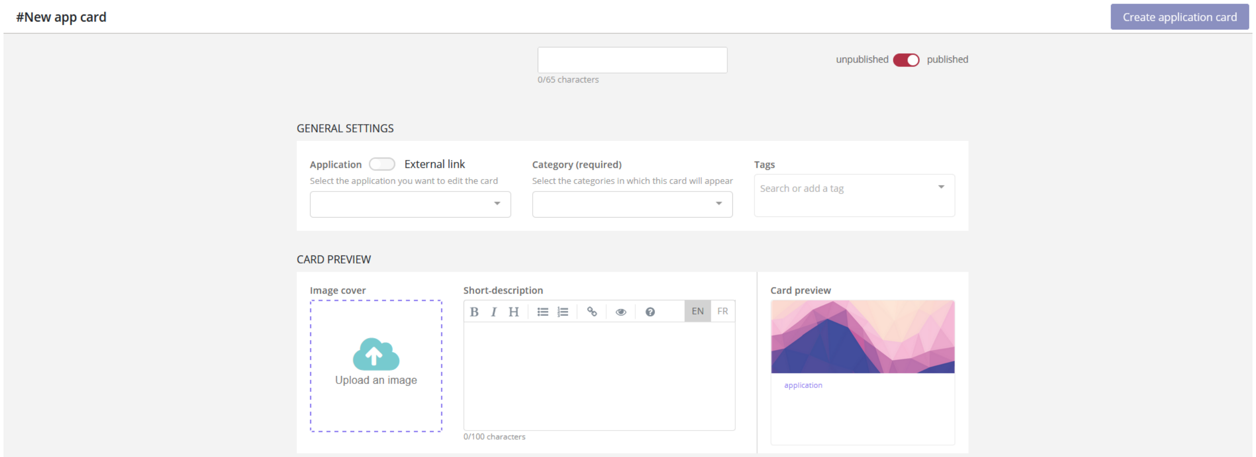
search

PUBLISHED	NAME	APPLICATION	CATEGORY	TAGS	UPDATED AT	CREATOR	
●	House price prediction	✓ house_pricing (model)	real estate	house, real estate, model, pricing		arnold zephir	⋮
●	Daily forecasting of electricity	✓ electricity_forecast (shiny)	energy	electricity		arnold zephir	⋮
○	Sales forecasting model	✗ no app/link	retail	retail		pierre nowak	⋮
○	Find faces in picture	✓ find_faces_in_pictures (notebook)	marketing			Gabriel Philippe	⋮
●	Wanted prevision.io founders	✓ IntruderDetectionInACrowd (notebook)	media	security		pierre nowak	⋮
●	Recovery loan	✓ sales-recouvrement (dash)	bank	bank, defect prediction		arnold zephir	⋮
●	Sales forecasting	✓ prevision-vente (node)	retail	retail, demand forecasting		alexandra texier	⋮

As you can see, a card has :

- A published status (yes / no)
- A name
- An application linked to it (previously deployed)
- A category
- Tags (optionnal)
- An update time
- A creator
- A contextual menu, allowing you to :
  - Edit it
  - See it
  - Remove it

To create a new card, click on the top right button `Create card`. You'll land on the following form :



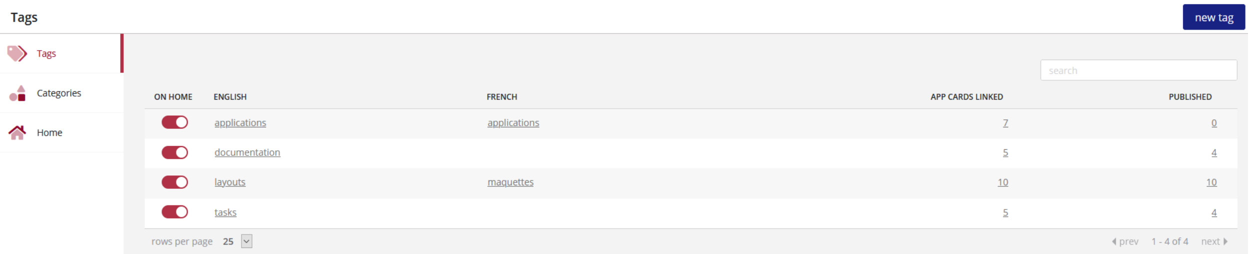
Here, you are asked to provide :
- A name for you card
- A publshed status (yes / no)
- Some settings related to the card, such as :
  - The application linked to it (or an external link)
  - Its category (required)
  - Tags (optionnal)
  - Card picture
  - Card short description (in english and french)
  - Card long description (in english and french)
  - « How to use it » description (in english and french)

Afterwards, a click on the `Create application card` button located at the top-right of the screen will start the deploying process. It will typically take less than 1 minute to deploy a new card.

Remarks : Since cards have to be linked to an existing Category, you may need to create a new category prior to created a new card. To do so, please head to the `Configuration` part of the STORE, located in the header.

12.3 Configuration

To access the configuration part of the STORE, please head to the `Configuration` link located in the header.



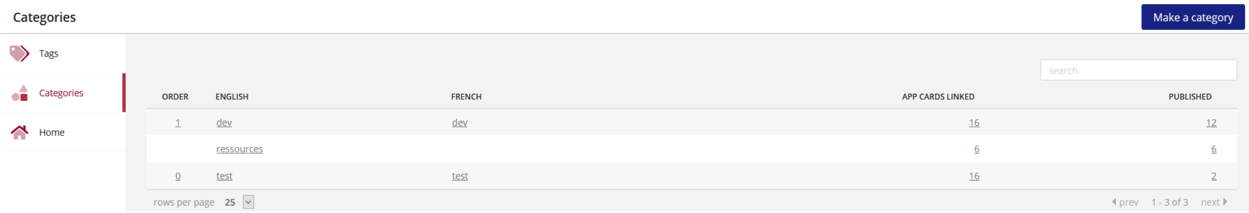
ON HOME	ENGLISH	FRENCH	APP CARDS LINKED	PUBLISHED
●	applications	applications	7	0
●	documentation		5	4
●	layouts	maquettes	10	10
●	tasks		5	4

Here, you can configure :
- Tags
- Cateogires
- Home page of the STORE

12.3.1 Tags

Tags are meta description of a card. They can be created easily by clicking on the `new tag` button in the top right of the screen. You just need to add a tag name in english and in french and publish it.

12.3.2 Categories



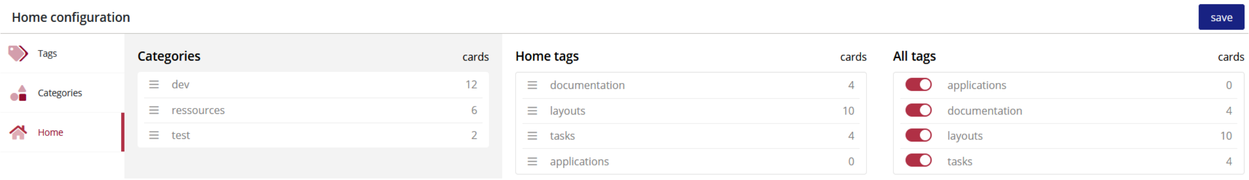
ORDER	ENGLISH	FRENCH	APP CARDS LINKED	PUBLISHED
1	dev	dev	16	12
	ressources		6	6
0	test	test	16	2

rows per page 25 ◂ prev 1 - 3 of 3 next ▸

Categories are a group of topics that cards can be linked to. They are mandatory so please make sur to create a new category before trying to create a new card.

Just like tags, categories can be created only with and english and french name.

12.3.3 Home



Categories	cards
dev	12
ressources	6
test	2

Home tags	cards
documentation	4
layouts	10
tasks	4
applications	0

All tags	cards
applications	0
documentation	4
layouts	10
tasks	4

Here you can configure how cards are beeing displayed in the home page of the STORE. You can chose to display (or not) categories and tags and eventually specify their display order.

CHAPITRE 13

Api

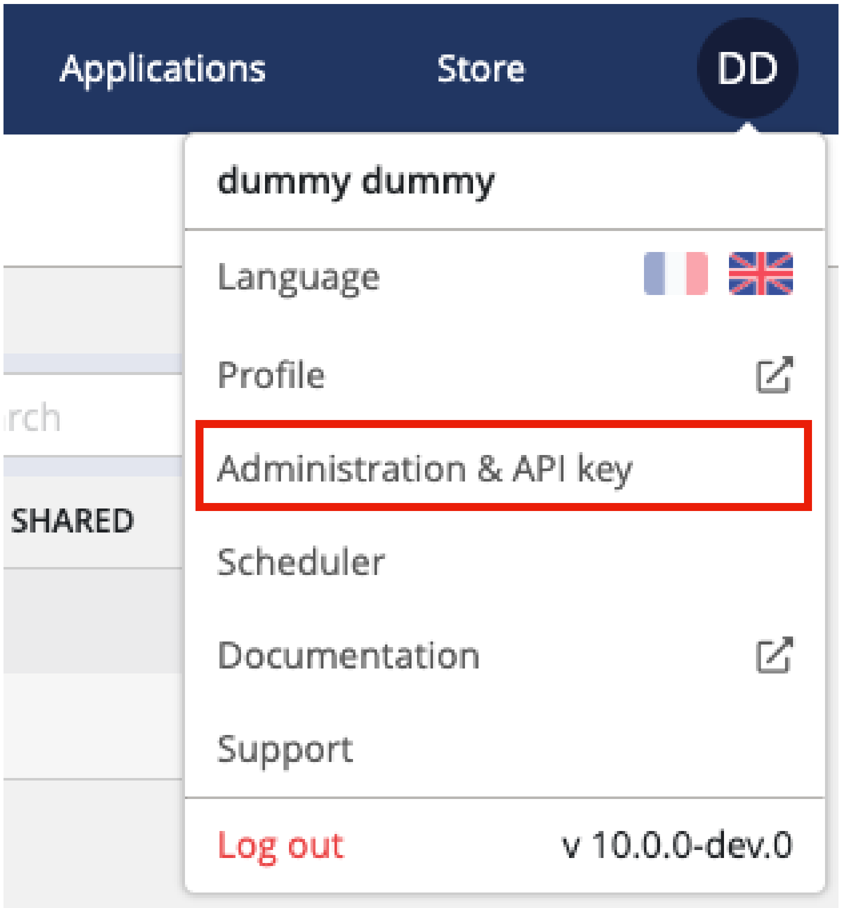
Prevision.io offers the possibility of training, predicting and use SDK by API call.

All the APIs available to interact with Prevision.io, without front-end, are detailed below : https://xxx.prevision.io/api/documentation (xxx is the name of your instance).

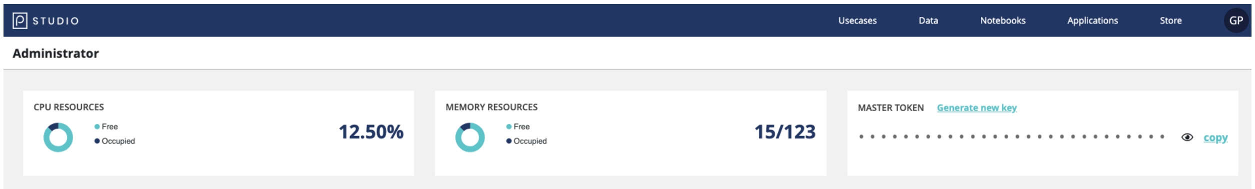
13.1 Master key

Any HTTP request for access to the prevision.io APIs must include a Header Authorization whose value is a valid JSON Web Token. This master token must also be set to use the SDK, either through an environment variable or explicitly when initializing the client.

This can be obtained by going to the user menu and clicking on the *API key* item (*Administration & API key* if you have admin rights) :



The component to copy and (if needed) regenerate the master token is located on the right of the screen.



Warning : This key is strictly private. It must therefore be known only to the account holder.

If the key is ever compromised, it can be regenerated. In this case the old key will become obsolete. It will then be necessary to update your applications with the new key.

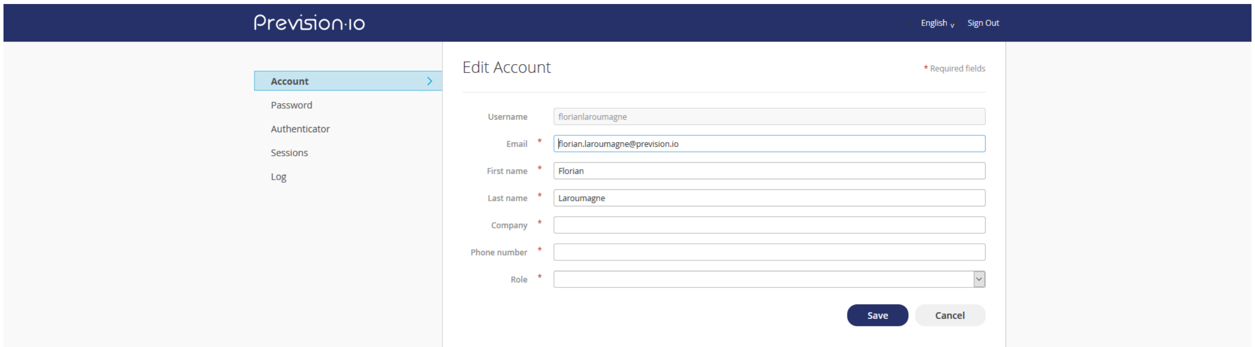
CHAPITRE 14

My account

You can access your account details by clicking on the following icon :

👤

In this screen, you will find the information you filled in when you registered. This information can be modified at any time, except for your email address.



Edit Account

* Required fields

Username	florianlaroumagne
Email *	florian.laroumagne@prevision.io
First name *	Florian
Last name *	Laroumagne
Company *	
Phone number *	
Role *	

You can also manage your password, activate two-factor authentication, or consult the login connections to your account.

In any case, if you encounter any problems, do not hesitate to contact us at the following address : support@prevision.io

CHAPITRE 15

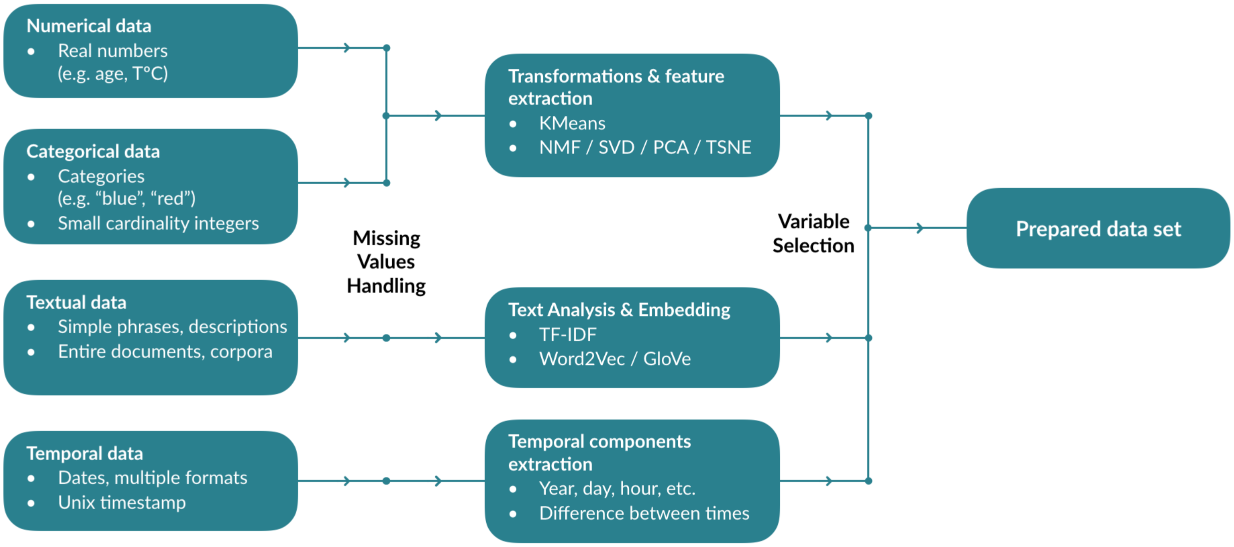
How does it work ?

The first step that the platform will perform is a data transformation. The types (Numeric, Text, Date, etc.) of the different columns will be detected, and specific transformations will be applied to them.

- Numerical & categorical variables :
    - Unsupervised clustering (K-Means)
    - Non-negative matrix factorization
    - Singular Value Decomposition
    - Principal Component Analysis
    - Stochastic Neighbor Embedding (T-SNE)
- Text variables :
    - Statistical analysis (TF-IDF)
    - Embedding (Word2Vec & GloVe)
- Time data :
    - Extraction of components (Year, day, time, etc.)
    - Distances (durations) between data set dates

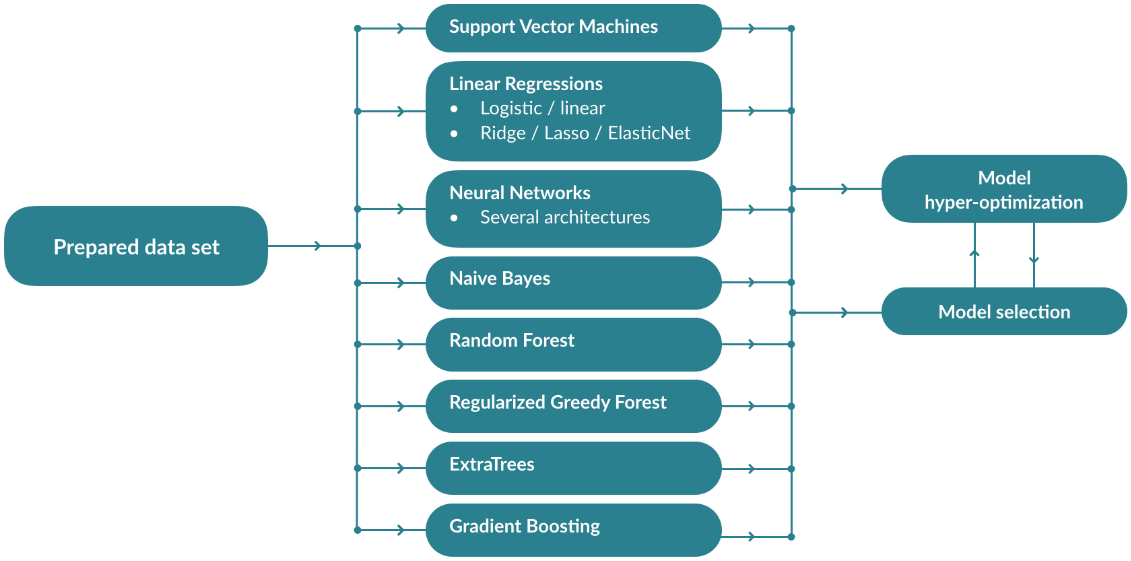
Depending on the problem, missing values will be processed differently, imputed if possible, or encoded in a specific way.

All transformed features will be evaluated and selected to create the data set used for the training.



15.1 Model training

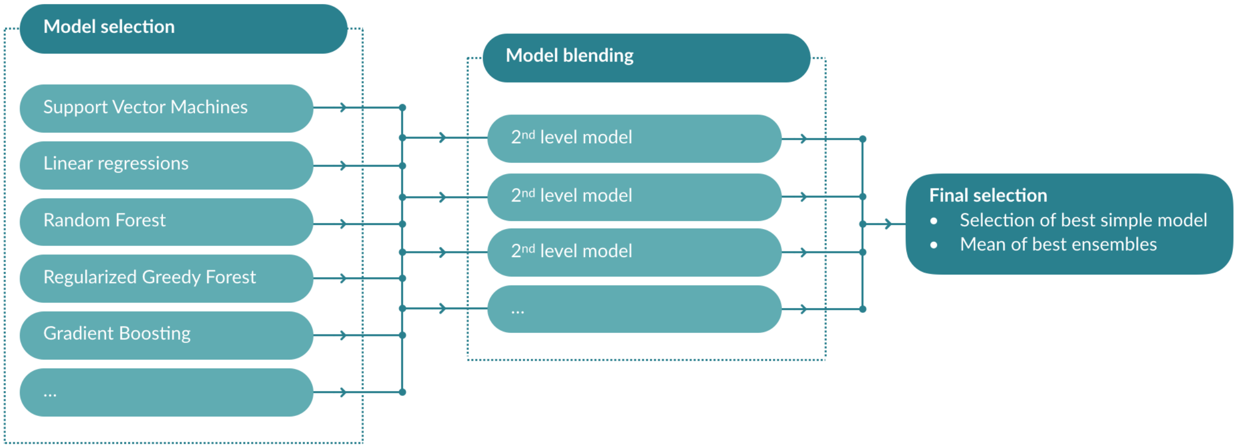
From the transformed dataset, a collection of models will be trained, and their optimal hyper-parameters will be optimized. To avoid the combinatorial explosion that would result from the « brute-force » optimization of all algorithms and their hyper-parameters, the Previsision.io engine learns the right combinations of models and parameters according to the data set and the type of problem in order to rapidly train the best models



15.2 Ensemble learning

Once a series of models have been trained, the platform will train a second level of models using the predictions of the first as input data. This makes it possible to compensate for the errors of the different models and to provide more stable and accurate predictions.

The average of the predictions of the second-level models is taken to maximize the stability of the predictions. The cross-validation system avoids the leakage of information across the different models and ensures a theoretical score as close as possible to reality.



CHAPITRE 16

Release notes

16.1 Version 10.16

16.1.1 Studio

Features

- Increase time of validation email : validation email for a cloud.prevision.io subscription is now increase from 5 minutes to 30 minutes
- NLP : Use case creation is blocked if user unchecked all textual features from feature engineering with a 100% textual features dataset
- Some liste sort were not functional and have been deactivated in order to prevent deviant behavior. These sorts will be re-activated as soon as the identified technical issue will be corrected

Bug Fixes

- Unification of date format between store and studio
- Changement of data type and training during the creation of a use case is no longer impossible
- All feature engineering used during the creation of a use case are now displayed in the features configuration menu of a use case
- Numerical features using thousands separators are no longer considered as categorial features
- Token validity time as been increased
- Object detector :
  - Labels and percentages of an object detector use case are now visible in the image sample of the use case
  - Prediction images from a shared use case are now visible
- Dashboards lists are now refreshed when user delete a use case or a dataset from corresponding menus
- Decision tree's numerical values are now given to two decimal points
- Bivariate analysis graph is now sort by lexicographic order
- Conflict between null/negative values and metric calculation

16.1.2 Store

Features

- Drift visualisation between main and challenger models is now available in order to choose better the production model

Bug fixes

- Feature distribution graph are now fixed
- Main & challenger models :
    - Improvements of prediction and feature distribution graphs
    - Improvement of drift graph
    - General improvement of graphs design
- No results from a prediction
- Unification of date format
- Card images of applications are now displayed
- Most probable predictions from a multi classification application is now displayed first

16.2 Version 10.15

16.2.1 Studio

Features

NLP :
- New rules regarding NLP use case have been defined regarding the dataset :
    - A 100% textual feature dataset :
        - Simple and normal models have been deactivated in the use case creation interface. Only the Naive Bayes (NBC) model and advanced models can now be applied. This is due to the fact that simple and normal models are not working with dataset containing only textual features dataset and produce errors in the DAG.
        - Messages in the interface are now visible in order to explain to the user why these models are not available in those cases
    - A 0% textual feature dataset :
        - Feature Engineering « textual features » are deactivated as they cannot be used in those cases
        - A message in the interface explains to the users the new interface behavior
    - Mix between textual and other features :
        - New messages have been displayed in the interface explaining that FE textual features will be only applied to advanced models and Naive Bayes model
- The default configuration of a use case including textual features is now as the image bellow. Users will now used a more powerful textual feature by default

Experimental time series :
- This option is no longer available in the interface due to frequent problems. Will be reintegrated after mre research.
- The SDK no longer allows you to use this option.

Bug fixes :
- Shared memory limit increased from 8Gb to 64Gb
- Duplication of dataset features

16.2.2 Store

Features

- Model challenger :
    - Tag main & challenger : you can define for each deployed use case which model is your main and, optionally, which one is the challenger. You will then access to comparison graphs between these two models and change at any time the main model in production.
    - Versioning : each new configuration of main/challenger model creates a new version of your application. For each version, you now have access to metrics allowing you to determine which model is the most suitable to be deployed between your main and your challenger models.
    - Rollback : you can rollback to an old main/challenger model configuration by clicking on the rollback button in the version menu of a deployed application
- New graphs for a deployed use case :
    - In the usage menu, you will now find graphs showing you the numbers of predictions, the response time, all the errors in the last 24 hours and all the errors since deployment.
    - Prediction distribution :
        - Binary classification : positive and negative prediction distribution. Raw for the train model, validation threshold for the production model
        - Multi-classification : repartition of predictions by modality. Raw for the train model and best prediction for production model. The number of displayed top modalities can be configured.
        - Regression : Distribution of values for train and production predictions models split by intervals equal of 1/10 of train results intervals

Activity logs of deployed application are now available in the application.

16.3 Version 10.14

16.3.1 Studio

Features

- Object detector :
    - Migration from YoloV3 to YoloV5 and improvements of object detector uses cases
    - Target file can now be load for object detector prediction in order to have a prediction score
- Time series :
    - Rules of time series are now displayed in the interface of use case creation
    - Derivation and forecast windows fields are now easier to fulfill thanks to auto completion and restriction according to time series rules
- Administration :
    - CPU, RAM and notebook lifetime can now be defined up by admins
- Versioning :
    - Models can now be identified as deployable. In the store, the list of deployable models is now the one identified in the studio as deployable
- Reporting :
    - Report of use case models can be now generated trough the use case page
- Feature importance :
    - Feature importance of a model are now paged 10 by page
    - You can also search a feature and sort the features by importance
- DAG :
    - The DAG of a use case is now more detailed

- Other features :
  - When new version of a use case is created, the type of training is no longer editable
  - Feature grades have now an explanation in the interface
  - In the model page, all feature engineering used by this models are now displayed, even the internal ones
- Bug fixes :
  - Difference between top bar and use case page about best performance calculation
  - Bug regarding object detector use cases and the « filename » name column
  - Train a classification using error_rate_binary metric created an error

16.3.2 Store

Features

- Object detector :
  - Object detector use cases can now be deploy from the studio to the store
- Other features :
  - List of deployable models is now limited to identified models in the studio
  - Private GIT repositories are now identified with a lock icon
- Bug fixes :
  - Status display not « running » when a use case is deployed
  - Fields of a new app versioning where not pre filled when using GitHub

16.4 Version 10.13

Release date : 2020-10-22

16.4.1 Studio

Features

- User interface of models selection : Users choices regarding model selection in advanced options during the creation of a use case are now extended.
- FastText : First NLP treatments are coming in Prevision.io platform. Text roles features of a use case can now be optimized using NLP algorithm
- Image Detector use cases : migration to YoloV5 treatment allowing a better treatment for image detector use cases
- UX/UI improvements

Bug fixes

- Fix of a display issue regarding number format depending on user interface language
- Lack of some calculated feature grades
- F1 score calculation of a model is now based on the optimal threshold value
- Fix of graphical display regarding bivariate analysis

16.4.2 Store

Features

- Webapp and notebook versioning : creation of new version of an already existing web app and notebook

- Access to private GitLab repo : SSO connexion to GitLab
- Build and deploy logs are now available for webapps
- application list of allowed user is now editable

Bug fixes

- Some field were not typed producing errors when fulfill with wrong type of data
- fix regarding an issue affecting all user of an instance instead of the user selected

16.5 Version 10.12

Release date : 2020-10-08

- Feature engineering importance
- Deployed Model SDK

16.6 Version 10.10

Release date : 2020-09-11

- Free trial version
- New interface for deployed usecases

16.7 Version 10.9

Release date : 2020-08-20

- New Prevision.io Studio homepage
- New Prevision.io Studio help page

16.8 Version 10.8

Release date : 2020-08-06

- SDK : Add versioning & sharing methods
- Add embedded support in store and studio through Zendesk

16.9 Version 10.7

Release date : 2020-07-23

- Connectors for Google Cloud Platform Buckets and BigQuery
- Advanced analytics for time series datasets
- Public mode for app deployment in store
- Subdomain URL mode for app deployment in store
- Add the capability to define environment variables when deploying apps in stores

16.10 Version 10.6

Release date : 2020-07-09
- Various improvements for apps deployment in store
- Better handling of very large datasets (> 10k columns)

16.11 Version 10.5

Release date : 2020-06-25
- Optimizations & bug fixes
- Model and app deployment is now entirely located in the Prevision.io store

16.12 Version 10.4

Release date : 2020-06-11
- Detailed statistics and analyses for datasets accessible from the data page

16.13 Version 10.3

Release date : 2020-05-28
- Usecase versioning

16.14 Version 10.1

Release date : 2020-04-10
- Create a new usecase from an existing one
- Simple models updated in order to match classical model analytics
- R & Python packages updated + new packages availlable for development environment

16.15 Version 10.0

Release date : 2020-03-05
- New graph-based usecase training monitoring
- Update scheduler page

16.16 Version 9.7

Release date : 2020-02-20
- Update notebook page to include current CPU & RAM usage
- Update and relocate administration page (now in top-right menu)
- Access data explorer from data screen

16.17 Version 9.6

Release date : 2020-02-06
- Change and relocate main menu to top bar
- Start usecase from data screen
- Update contact page

16.18 Version 9.5

Release date : 2019-12-19
- Refactoring of the main dashboard screen
- Refactoring of the usecase screen, including new analytics
- R & Python packages updated to matchs usecase APIs
- Improved explain screen stability when simulating predictions
- Added support of object detection usecase with CPU only (might take some computing time)
- Feature quality estimation

16.19 Version 9.4

Release date : 2019-10-04
- Refactoring of the data screen
- R & Python packages updated to matchs data screen APIs
- Improved rules of detection of typical columns (ID, TARGET, FOLD, WEIGHT)
- Improved explain screen stability when values are missing
- Improved date columns parsing in a dataset that handles multiple time zones
- Faster prediction time retrival when listing a high number of predictions
- Creation of an open data base with accessible data for special days (holidays, public days, sales, …) and for weather data

16.20 Version 9.3

Release date : 2019-08-14
- Refactoring of the new use case screen
- R & Python packages updated to matchs new use case APIs
- Refactoring of Prevision.io APIs. Documentation available @ https://xxx.prevision.io/api/documentation (xxx = instance name)
- Creation of instance specific Prevision.io's store, visible @ https://xxx.prevision.io/store (xxx = instance name)
- Optimisation of training time for gradient boosting trees models

CHAPITRE 17

Known issues

This page lists known issues with Prevision.io, along with ways you can avoid or recover from these issues. Those issues are given a high priority to the Prevision.io development team and will be solved in an upcoming release.

- There can be small differences between the unit and batch predict for the same data and model.
- The decision tree simple model will not accurately display the modalities for categorical features, instead only showing the encoded modalities. (red, green, blue will show up as 1, 2, 3)
- Downloading large files (> 1GB) from the notebooks can sometimes fail silently (the downloaded file might be incomplete).
- For multi-classification usecases, the bivariate analysis shows the top 3 classes *per bin* instead of the top 3 classes overall.
- The .svg format plot download will sometimes fail.
- Dataset upload will fail with empty datasets (only header). In that case the « Parsed » status will never complete or fail.
- Some advanced metrics (F2, F3, F4 Score, Lift, . . .) don't have a star system related to them

17.1 Experimental features

- ALN timeseries usecases might crash when given an ID column