

---

# Prevision.io Documentation

**Gerome Pistre**

**oct. 22, 2021**



---

# Prevision.IO Concepts and Documentation

---

<b>1</b>	<b>Value proposition</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>3</b>
<b>3</b>	<b>Conditions</b>	<b>5</b>
<b>4</b>	<b>contacts</b>	<b>7</b>
<b>5</b>	<b>Getting started</b>	<b>9</b>
5.1	Account creation . . . . .	9
5.2	Connection . . . . .	9
5.3	Cloud & freetrial limitations . . . . .	9
5.3.1	Studio . . . . .	11
5.3.1.1	Projects . . . . .	11
5.3.1.2	Data . . . . .	16
5.3.1.3	Experiments . . . . .	26
5.3.1.4	Pipelines . . . . .	67
5.3.1.5	Deployments . . . . .	78
5.3.1.6	Contributors . . . . .	82
5.3.1.7	Notebooks . . . . .	84
5.3.2	API . . . . .	85
5.3.2.1	Using The API . . . . .	85
5.3.3	SDK . . . . .	86
5.3.3.1	Using the Python SDK . . . . .	86
5.3.3.2	Using the R SDK . . . . .	88
5.3.3.3	Using the Prevision Quantum NN SDK . . . . .	88
5.3.4	Guides and Howto . . . . .	88
5.3.4.1	Datascience Guide . . . . .	88





# CHAPITRE 1

---

## Value proposition

---

AI in the enterprise promises leaps in efficiency, business innovation and customer-facing performance. However, to enable greater adoption, democratization and acceptance of AI, organizations must overcome not only talent gaps, but deployment, usability and governance gaps too. The real enabler of enterprise AI is the removal of friction between Business Departments, Data Science, and IT users. Data science is following the path of software development : Integrated environments, agile, iterative methods, and move to modularity and no-code approaches that empower both expert and citizen developers and data scientists alike. The winners in the Enterprise AI revolution are those who can achieve faster, more agile, more integrated, more collaborative production cycles across DataOps, ML Ops and DevOps areas. Prevision is an end-to-end enterprise AI platform, specifically designed to enable business users, data scientists, and developers to deliver AI projects that deliver ROI, faster. It streamlines the creation, deployment and management of AI-powered business applications across their full lifecycle.



## CHAPITRE 2

---

### Requirements

---

Prevision.IA is a SAAS platform optimized for Firefox and Chrome navigators. The cloud version can be accessed online at <https://cloud.prevision.io>, or it can be deployed on-premise or in your private cloud. Please visit us at <https://prevision.io> if you have any questions regarding our deployment capabilities.



## CHAPITRE 3

---

### Conditions

---

Please read the general terms and conditions available on following link : <https://cloud.prevision.io/terms>



## CHAPITRE 4

---

### contacts

---

If you have any questions about using Prevision.IO platform please contact us using the chat button on the Prevision.IO store interface or by email at the following contact address :

[support@prevision.io](mailto:support@prevision.io)





### 5.1 Account creation

By clicking to the following address, you will land on the connection page which allows you to create an account or sign in if you already have a Prevision.IO account.

<https://cloud.prevision.io>

In order to create a new account, just click on the sign up button next to the log in button. You will access the following account creation form.

Once you have filled the needed information, you will have a 30 days free but limited access to the Prevision.IO platform. In order to upgrade your free trial account into a full access one, please contact us on following email ([support@prevision.io](mailto:support@prevision.io))

Once done you can follow our *[complete guide to release a model](#)*

### 5.2 Connection

Once your account has been created, you will be able to access the prevision's Studio and Store and start creating models and deploying them.

Please note that SSO using you google/linkedin/GitHub account is available.

### 5.3 Cloud & freetrial limitations

If you are using our [cloud platform](#) using a free trial account, some limitations are set up. Here's a quick view of limitations for free accounts :

[English](#)

LOG IN. **SIGN UP.**

Register for a free 30-days trial period

First name

Last name

Country

Job title

Enterprise

Sector

Email

Password

Confirm password

☐ Je ne suis pas un robot


  
reCAPTCHA  
Confidentialité - Conditions

Fig. 1 – Signup screen

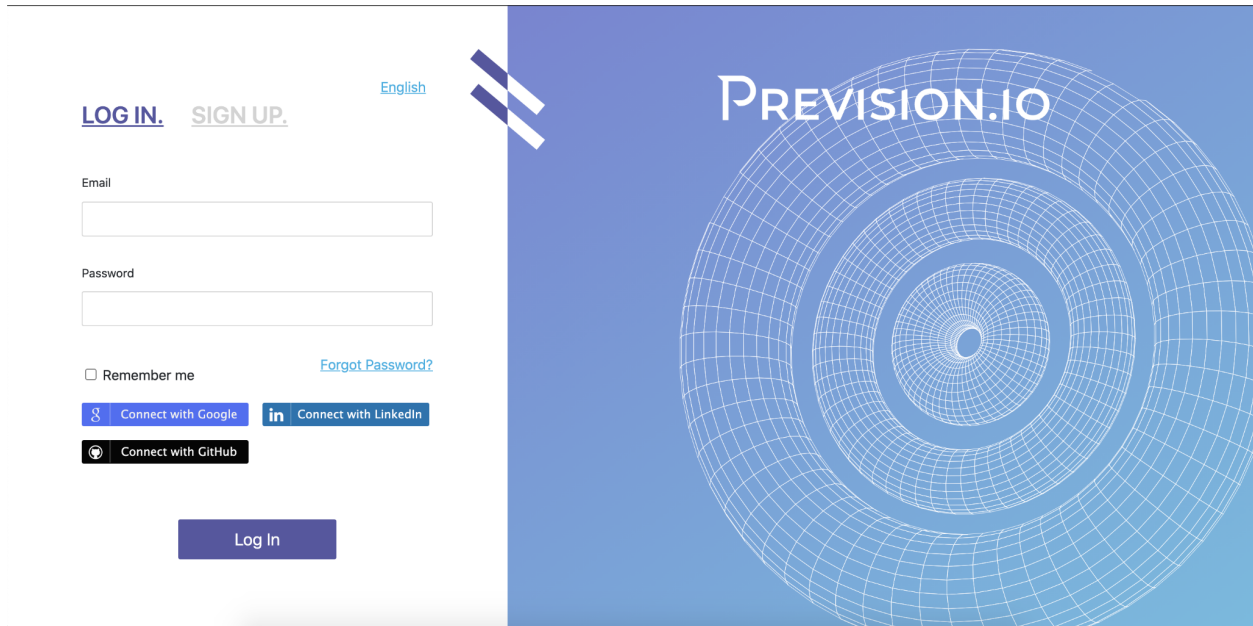


Fig. 2 – login with SSO

Tableau 1 – Freetrial Limitation

Entity	Action	Limitation
PROJECT	Create Project	Free trial users can create 2 Limited Projects
DATASETS	Add dataset from file / datasource	10 Datasets max + 1GB per dataset
IMAGE FOLDER	Add / Update / Delete image Folder in project	1 Image Folder
DATA SOURCE	Add / Update / Delete datasource in project	1 Datasource max
CONNECTOR	Add / Update / Delete connector in project	1 Connector max
USECASE	Add / Update / Delete experiment in project	5 Use Cases max
USECASE VERSION	Add / Update experiment version in project	3 Concurrent experiment versions
PREDICTION	Add / Update prediction in experiment version	2 Concurrent predictions
STORE APP	Deploy Apps	5 Concurrent deployed apps

## 5.3.1 Studio

### 5.3.1.1 Projects

In Prevision.IO studio, ressources, such as datasets or models, are scoped by project in order to structure your work and collaborate easily with people inside a project.

A project is a collection of :

- *Datas* : for importing and exporting your data from and to external database or files
- *Experiments* : to build model, evaluate them and compare them
- *Pipelines* : to set up and schedule datascience pipelines
- *Deployments* : to push models to production and monitor it
- *Collaborators* : to add and manage users

## List my Projects

All your projects are available on the homepage of your server. You can reach it with the mosaic icon on the upper-left corner of each screen

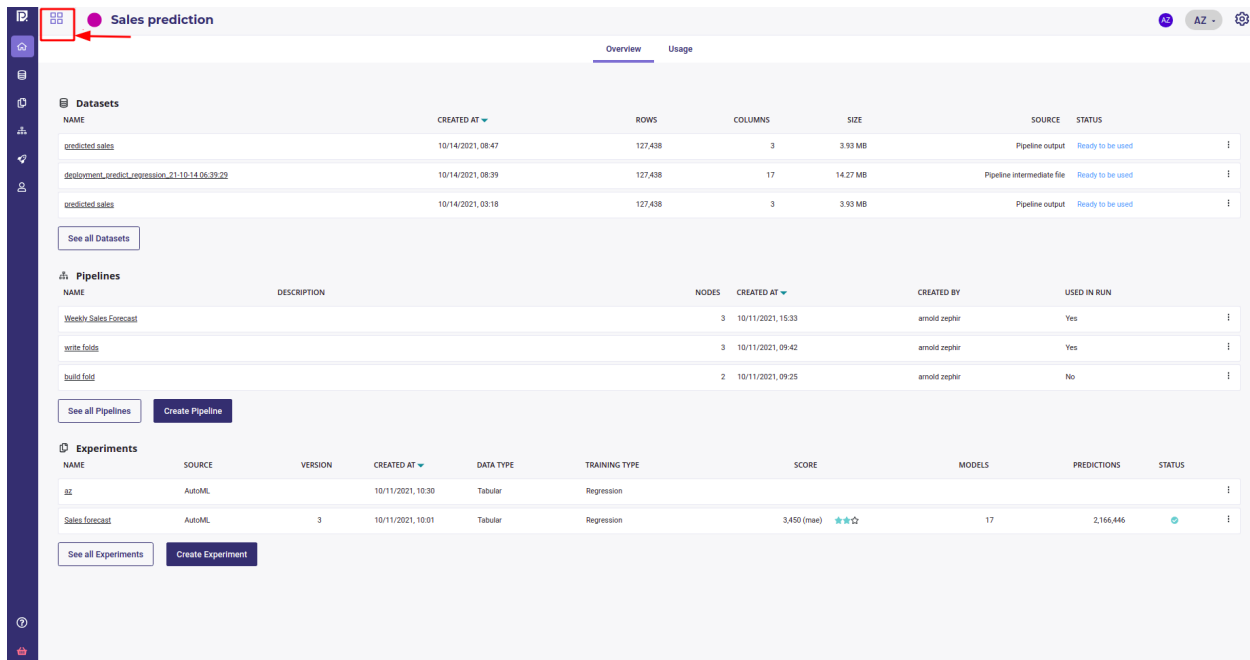


Fig. 3 – Go to the list of projects

You can switch from card view

To List view :

You will find the following information on the cards :

- project name
- created by and creating at
- description (if available)
- number of datasets/pipelines/use cases
- list of collaborators into the project and their associated role into this project

If your role has been setted up as admin into a project, an action button on top right of each card will be available. By clicking on this button you will be able to edit the project information and delete the project. Please note that deleting a project will also delete all sub project's items, such as pipelines or datasets, created on the project.

---

**Note :** Tips : you can filter projects by their names using the search bar on top right of the projects view

---

## Create a new project

In order to create a new project, you have to click on the “new project” button on top right of the “my projects” view.

You will access to the following interface :

In order to create your project you have to fulfill at least a color and a project name. You can also add a description of your project.

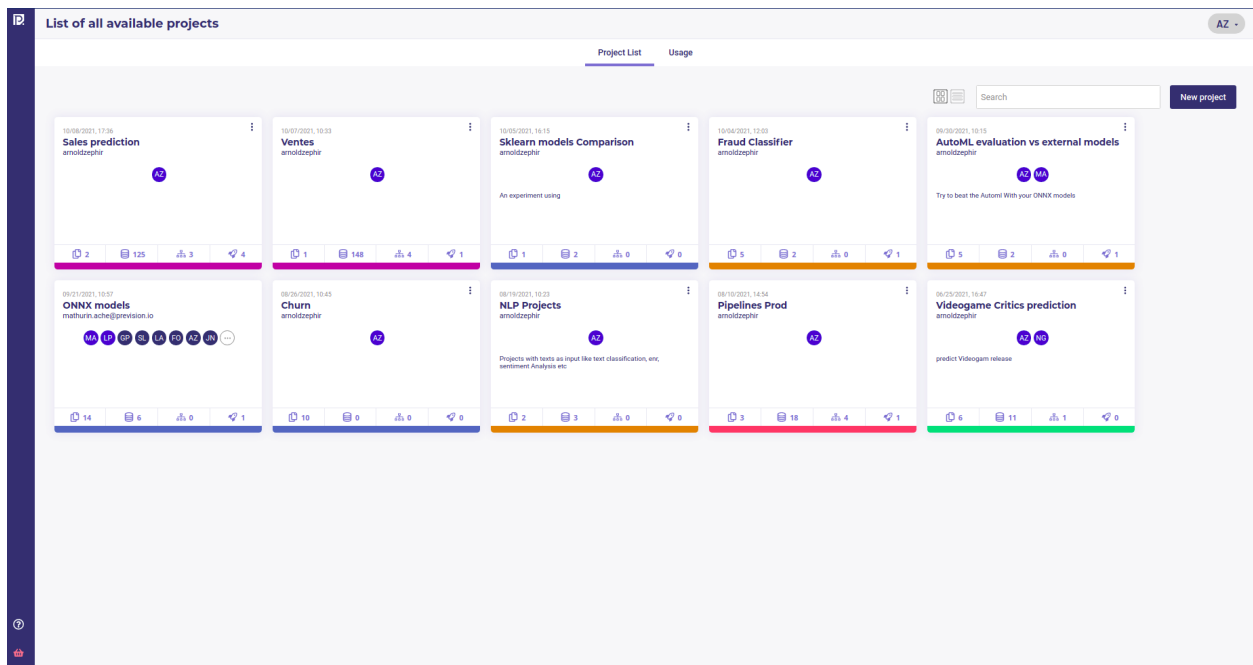


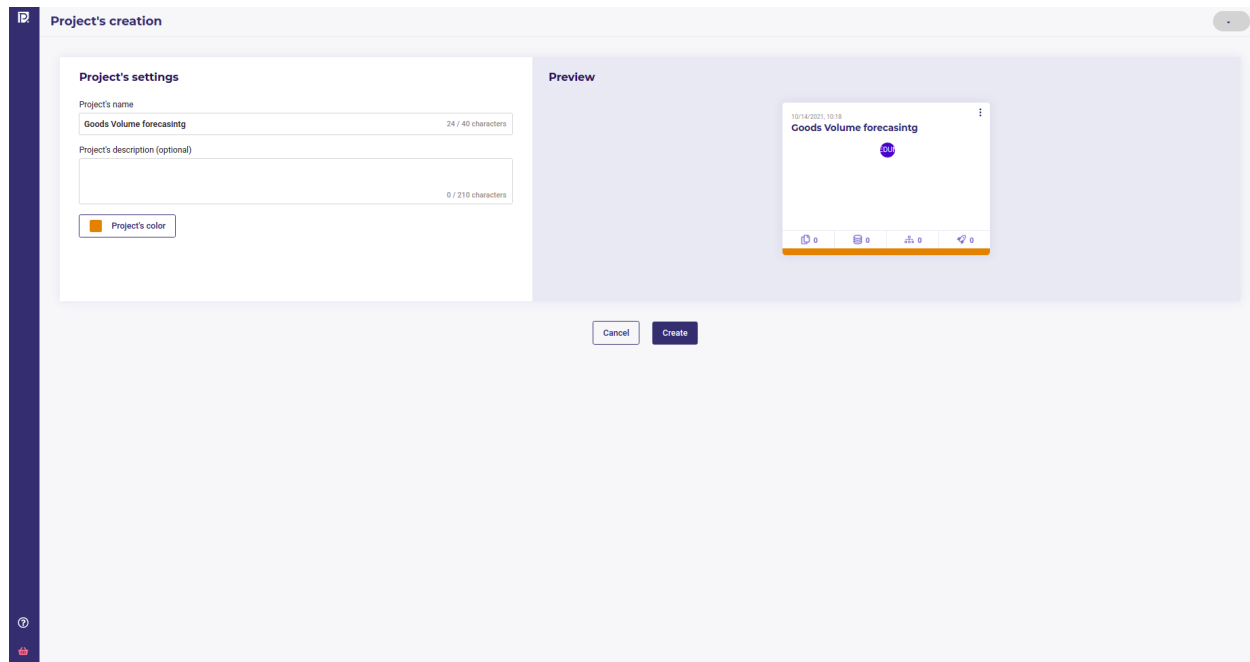
Fig. 4 – List of projects

The screenshot displays the 'List of all available projects' interface in table view. The table has columns for NAME, CREATED BY, CREATED AT, EXPERIMENTS, DATASETS, PIPELINES, and COLLABORATORS. The projects are listed in the following order:

NAME	CREATED BY	CREATED AT	EXPERIMENTS	DATASETS	PIPELINES	COLLABORATORS
Sales prediction	arnold zephir	10/08/2021, 17:36	2	125	3	AZ
Ventes	arnold zephir	10/07/2021, 10:33	1	148	4	AZ
Sklearn models Comparison	arnold zephir	10/05/2021, 16:15	1	2	0	AZ
Fraud Classifier	arnold zephir	10/04/2021, 12:03	5	2	0	AZ
AutoML evaluation vs external models	arnold zephir	09/30/2021, 10:15	5	2	0	AZ, MA
ONNX models	Mathurin ACHE	09/21/2021, 10:57	14	6	0	MA, EP, GP, LS, LA, FO, AZ, BR, ...
Churn	arnold zephir	08/26/2021, 10:45	10	0	0	AZ
NLP Projects	arnold zephir	08/19/2021, 10:23	2	3	0	AZ
Pipelines Prod	arnold zephir	08/10/2021, 14:54	3	18	4	AZ
Videogame Critics prediction	arnold zephir	06/25/2021, 16:47	6	11	1	AZ, BR

rows per page: 10

Fig. 5 – List of projects



Please note that you can at any moment, if admin role into a project has been setted up for your account, change these information by clicking on “settings” into the project menu.

All your projects will be displayed in the “my projects” view. Two different displays are list view and cards view and you can switch between one view and another by clicking on the view button you preferre next to the search bar.

## Project navigation

By entering a project, you will first be redirected to the project homepage. The following sections, including the 3 latest entries for each section, are displayed :

- Datasets : last uploaded dataset
- Pipelines : last pipeline templates
- Usecases : last experiments

Under each section you will also find a link to the dedicated page, also available through the left project main menu, and, for pipelines and experiment, a shortcut to create new ones.

Into a project you can load data, create a pipeline in order to automate some task or data transformations and train models into use cases menu. Once you enter a project by clicking on its card or on the list, the project menu will be loaded on the left navigation bar.

- Home : you will find here last items from datasets, pipelines and use cases created into the project.
- Usecases : you will find all your use cases trained into the selected project
- Data : you will find here your datasets and the connectors setted up on the project
- Pipelines : you will be able thanks to pipeline to augment your data and automate some actions
- Deployment : deploy and monitor deployed models and applications
- Collaborators : list of all project collaborators and their associated project role
- Settings : if your project role is admin, this menu is available and allows you to edit the project informations

**Axel's Project**

**Datasets**

NAME	CREATED AT	ROWS	COLUMNS	SIZE	DATA SOURCE	STATUS
regression_house_80_lite	06/23/2021, 17:14	399	21	39.67 KB		Ready to be used
french_tweets_lite	06/23/2021, 15:21	404,769	2	39.42 MB		Ready to be used
series-trafics	06/23/2021, 15:20	99,999	3	3.29 MB		Ready to be used

See all Datasets

**Pipelines**

No data for table

See all Pipelines Create Pipeline

**Usecases**

NAME	VERSION	CREATED AT	DATA TYPE	TRAINING TYPE	SCORE	MODELS	PREDICTIONS	STATUS
classif_edf_80_lite	1	06/24/2021, 10:28	Tabular	Classification	-	0	0	🟢

See all Usecases Create Usecase

Aide

## collaboration into a project

Prevision.IO studio is built in order for our users to collaborate within the projects. To do that, into the “collaborators” menu of a selected project you can manage, if your role is admin on the project, the collaborators and the right inside the project.

**Project's collaborators**

Share this project with collaborators

Collaborator's email admin Invite this collaborator search

NAME	EMAIL	ROLE
Axel Chauvin QA test	axel.chauvin@prevision.io	admin
Frédéric OGLAZA	frederic.oglaza@prevision.io	admin
Lucie Albira	lucie-externe@prevision.io	admin
nada feki	nada.feki@prevision.io	admin
Simon Levacher	simon.levacher@prevision.io	admin

rows per page: 25 1 - 5 of 5

Aide

- Add a user : by enter the email address of a Prevision.IO platform registered user you can add a collaborator
- role : if your role level is admin into the project you will be able to edit user roles
- by clicking the delete button on the right side of a user, you can disable the access to the selected user to the project

### Project roles

Into a project there are 3 levels of roles :

- End-user : in this project, the user can only access to the list of deployed models and applications and make predictions
- Viewer : you can navigate into all ressources of a project and visualize information but you can't download or create ressources
- Contributor : Viewer rights + you can create and manage resources of the project
- Admin : Contributor rights + you can manage project users and project settings

### Edit a project

You can change the following parameters of your project by clicking on settings on the main navigation of your project or, on the list/card view of your project by clicking on the action button.

- Name of the project
- Description of the project
- Color displayed on the card of the project

### Delete a project

If a project is no longer useful, you can delete it by clicking on the action button on the card/list projects view.

Warning : all ressources created into the project will be deleted with the suppression of the project with no possibility of back-up. If deleted, a project and its resources are no longer available for you but also for all users previously added to this project.

#### 5.3.1.2 Data

Data is the raw material for all experiments. All your data are scoped to a *project* and you can access them from the « Data » section on the collapsing sidebar

The datas section holds 5 kind of assets :

- datasets : input for train, predictions and pipelines
- Image folders : folders containing image
- Data sources : information about location ( database, table, folder) or remote dataset
- Exporters : information about remote table for exporting prediction

### Datasets

Datasets are tabular data resulting from a flat file upload, a Datasource or a *pipeline execution*. They are input for *training*, *predicting* and *pipeline execution*

---

#### Datasets

- may be created from file upload
- may be created from data source
- may be created from pipeline output
- may be downloaded
- may be exported with exporters
- may be used as pipeline input



The screenshot shows the 'Data' section of the Prevision.io studio. The left sidebar contains navigation links: Home, Data (selected), Experiments, Pipelines, Deployments, Collaborators, Notebooks, Help, and STORE. The main area is titled 'Customer Relation Ship' and has tabs for Datasets, Image folders, Data sources, Exporters, and Connectors. The 'Datasets' tab is active, showing a table of datasets. The table has columns: NAME, ROWS, COLUMNS, SIZE, CREATED AT, CREATED BY, SOURCE, STATUS, and ACTIONS. Two datasets are visible:

NAME	ROWS	COLUMNS	SIZE	CREATED AT	CREATED BY	SOURCE	STATUS	ACTIONS
test	2,000	13	123.96 KB	08/24/2021, 15:53	arnold zephir	File upload	Ready to be used	Compute embeddings
test	8,000	13	494.88 KB	08/24/2021, 15:53	arnold zephir	File upload	Ready to be used	Explore embeddings

At the bottom of the table, it says 'page: 25' and 'prev'.

Fig. 6 – Data section

The screenshot shows the 'Dataset list' interface for the 'Retail' dataset. The left sidebar is the same as in Fig. 6. The main area is titled 'Retail' and has tabs for Datasets, Image folders, Data sources, Exporters, and Connectors. The 'Datasets' tab is active, showing a table of datasets. The table has columns: NAME, ROWS, COLUMNS, SIZE, CREATED AT, CREATED BY, SOURCE, STATUS, and ACTIONS. There are 11 datasets listed:

NAME	ROWS	COLUMNS	SIZE	CREATED AT	CREATED BY	SOURCE	STATUS	ACTIONS
test_intermarche	86,940	23	12.07 MB	08/25/2021, 13:54	arnold zephir	File upload	Ready to be used	Compute embeddings
train_intermarche	350,470	24	47.06 MB	08/25/2021, 13:53	arnold zephir	File upload	Ready to be used	Compute embeddings
titanic_train	891	13	45.64 KB	07/01/2021, 18:35	pierre Nowak	File upload	Ready to be used	Compute embeddings
titanic_train	891	13	45.64 KB	07/01/2021, 18:35	pierre Nowak	File upload	Ready to be used	Compute embeddings
titanic_test	418	12	21.12 KB	07/01/2021, 18:35	pierre Nowak	File upload	Ready to be used	Compute embeddings
titanic_train	891	13	45.64 KB	07/01/2021, 18:35	pierre Nowak	File upload	Ready to be used	Compute embeddings
sales_timeseries	543,850	5	18.6 MB	05/26/2021, 16:06		File upload	Ready to be used	Compute embeddings
sales	543,850	5	18.6 MB	05/25/2021, 17:44		File upload	Ready to be used	Compute embeddings
output_2	402,423	4	12.34 MB	05/25/2021, 15:32		File upload	Ready to be used	Explore embeddings
ventes_produits	32,793,018	7	2.49 GB	05/24/2021, 14:15		File upload	Ready to be used	Explore embeddings
mock_kaggle	937	4	21.71 KB	05/20/2021, 15:59		File upload	Ready to be used	Compute embeddings

At the bottom of the table, it says 'rows per page: 25' and 'prev 1 - 11 of 11 next'.

Fig. 7 – Dataset list

When you click on the datasets tabs, you see a list of your dataset with :

- filter checkbox for origin of the dataset ( *pipeline output*, *Datasource*, *File upload*, *Pipeline intermediate file* )
- search box filtering on the name of datasets
- name of the datasets and information about them
- status. A dataset could be unavailable a short time after its creation due to parsing.
- a button to compute embeddings or explore them if already computed ( see [the guide about exploring data](#) )

## Create a new dataset

In order to create new *experiments* you need a dataset. They can be created by clicking on the « import dataset » button.

Fig. 8 – Dataset import

Datasets are always created from tabular data (database table or files ). You can import data from a previously created *datasource* or from a flat csv file.

For data coming from file ( upload, ftp, bucket, S3,... ) you could input the columns and decimals separator but the auto detect algorithm will work in most of cases.

When you click on the « save dataset » button, the dataset will immediately be displayed in your list of datasets but won't be available for a few seconds. Once a dataset is ready, its status will change to « Ready to be used » and you can then compute embeddings and use it from training and predicting.

## Analyse dataset

Once a dataset is ready, you got access to a dedicated page with detail about your dataset

— **General Information** : Dataset summary, feature distribution and correlation matrix of its features

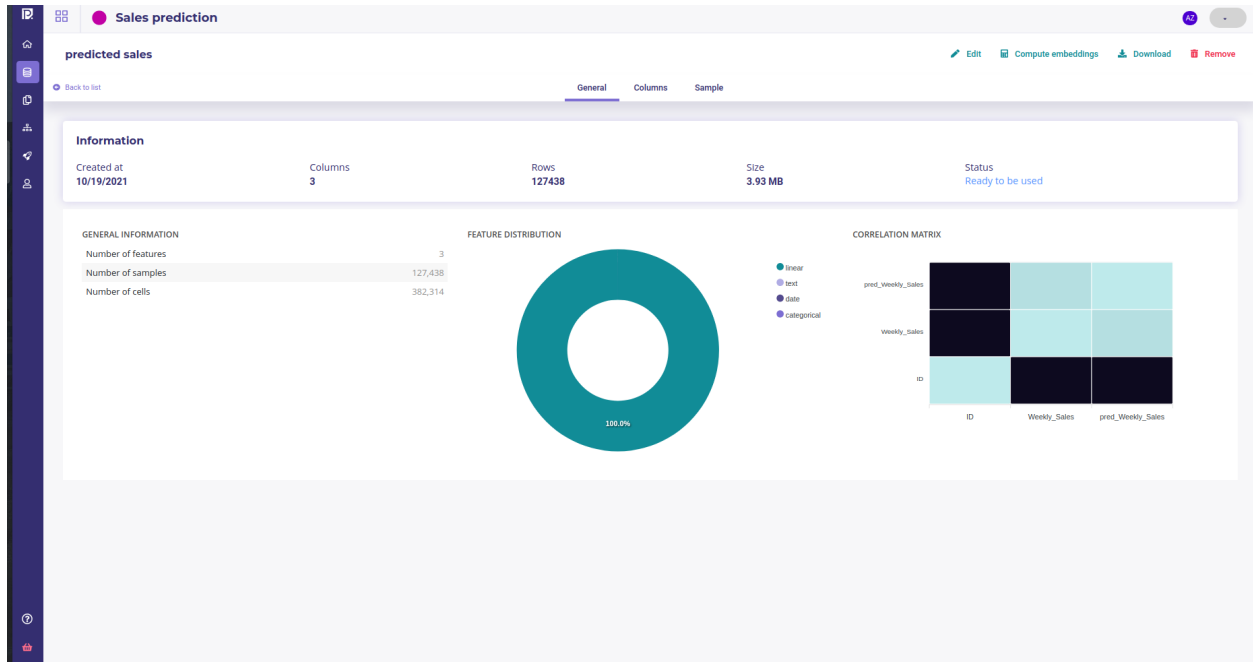


Fig. 9 – Dataset detail

— **Columns** : Information about its columns. You can click on a column to get more information about it, its distribution for example

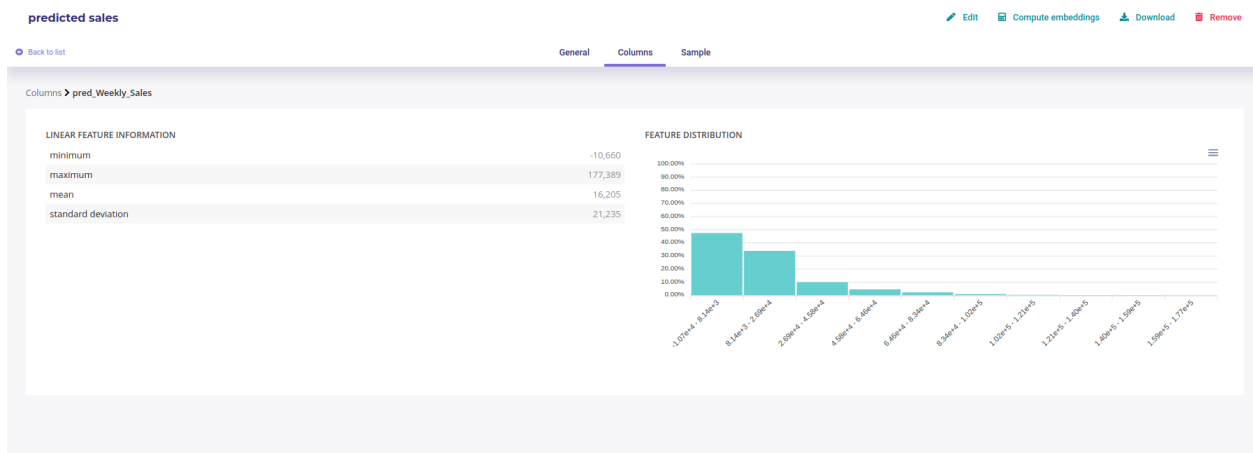
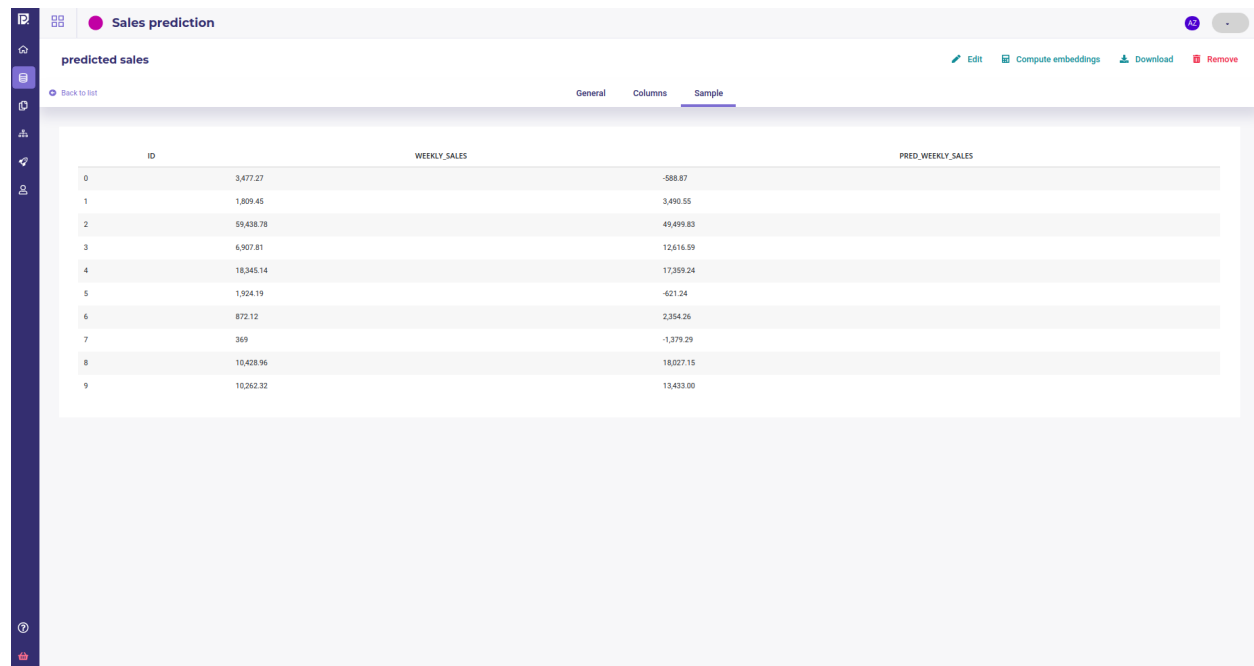


Fig. 10 – Detail of a column

— **Sample** : a very short sample of the dataset



ID	WEEKLY_SALES	PRED_WEEKLY_SALES
0	3,477.27	-588.87
1	1,809.45	3,490.55
2	59,438.78	49,499.83
3	6,907.81	12,616.59
4	18,345.14	17,359.24
5	1,924.19	-421.24
6	872.12	2,354.26
7	369	-1,379.29
8	10,428.96	18,027.15
9	10,262.32	13,433.00

Fig. 11 – Sample of a dataset

## Edit and delete

You can edit name of a dataset, download it or remove it of your storage either in the top nav menu of the dataset details page or from the dot menu in the list of your dataset.

<input type="checkbox"/>	NAME	ROWS	COLUMNS	SIZE	CREATED AT	CREATED BY	SOURCE	STATUS	ACTIONS
<input type="checkbox"/>	multiclassif_wines	1,483	12	77.54 KB	10/19/2021, 15:38	arnold zephir	File upload	Ready to be used	<a href="#">Compute embeddings</a>
<input type="checkbox"/>	predicted sales	127,438	3	3.93 MB	10/19/2021, 15:32	arnold zephir	Pipeline output	Ready to be used	<a href="#">Compute embeddings</a>
<input type="checkbox"/>	deployment_predict_regression_21-10-19 13:29:13	127,438	17	14.27 MB	10/19/2021, 15:29	arnold zephir	Pipeline intermediate file	Ready to be used	<a href="#">Compute embeddings</a>
<input type="checkbox"/>	predicted sales	127,438	3	3.93 MB	10/19/2021, 14:19	arnold zephir	Pipeline output	Ready to be used	<a href="#">Compute embeddings</a>

Fig. 12 – Edit and remove dataset

When you remove a dataset, local data are completely removed. Data source data are left untouched.

## Compute embeddings

See the : [Complete Guide for exploring data](#)

## Image folders

Image folders are storage for your image. It is source material for image experiments (classification, object detector, ...). For Images experiments you need an image folder.

### Image folders

- may be creating from file upload

- can not use datasource or connectors
- can be downloaded
- can not be exported

## Create a new image folder

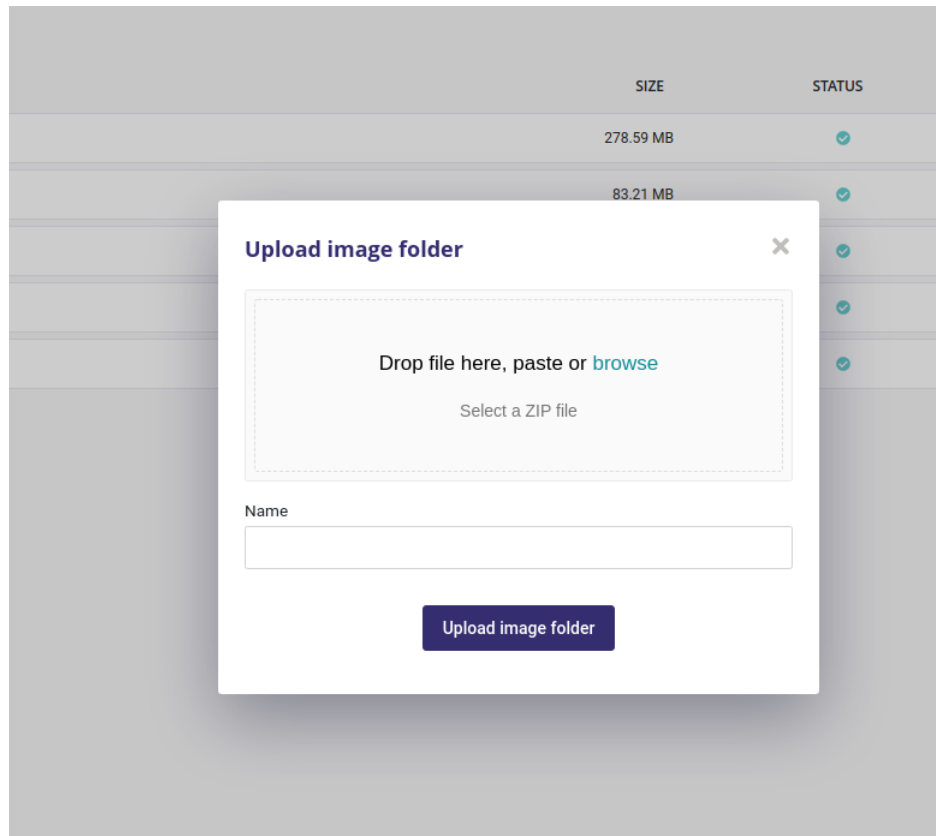


Fig. 13 – Upload a folder of image

When clicking on « Upload Image Folder » button, you can upload a zip file either from drag and dropping it or from selecting it from your local file browser. The zip file must contain only image but they can be organized into folders.

After having given a name, just click on « upload image folder » and wait. Your images will be available for *experiments* in a few seconds.

## Edit and remove

You can edit the name of of your folder from the list of image folder, using the three-dots menu on the right. Removing the image folder from your storage is available from this menu too.

## Connectors

Connectors are used to hold credentials used to access external databases or filesystems. You need to create a connector first to use Datasources and Exporters.

### connectors

- may be used for creating data source
  - may be used for creating exporter
- 

In the Prevision.io platform you can set up connectors in order to connect the application directly to your data sources and generate datasets.

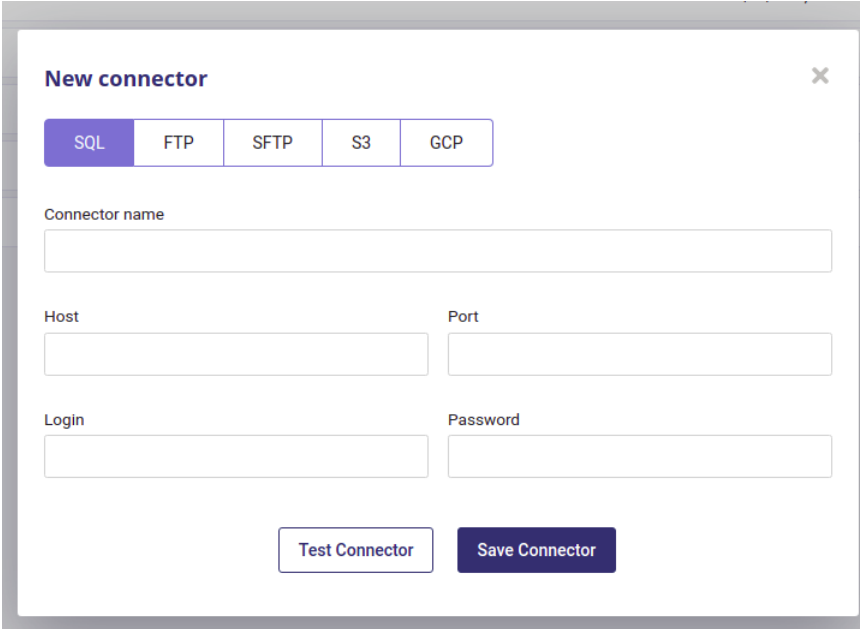
The general logic to import data in Prevision.io is the following :

- Connectors hold credentials & connection information (url, etc.)
- Datasources define how the data will be extracted (SQL query, file path)
- Datasets are used to create experiments and analyses

Several connector types are available :

- SQL databases
- FTP server
- SFTP server
- Amazon S3 datastore
- GCP

### Create Connector



The screenshot shows a modal window titled "New connector" with a close button (X) in the top right corner. Inside the modal, there are five tabs: "SQL", "FTP", "SFTP", "S3", and "GCP". The "SQL" tab is currently selected. Below the tabs, there are four input fields arranged in two rows. The first row contains "Connector name" and "Host". The second row contains "Port" and "Login". Below these fields, there is a "Password" field. At the bottom of the modal, there are two buttons: "Test Connector" and "Save Connector".

Fig. 14 – Create a new connector

By clicking on the “new connector” button, you will be able to create and configure a new connector. You will need to provide information depending on connector’s type in order for the platform to be able to connect to your database/file server.

---

**Note :** TIPS : you can test your connector when configured by clicking the “test connector” button.

---

Most of the connector require :

- an host

- a port
- a username
- a password

but [S3 Connectors](#) and [Google Cloud Platform](#) require special information provided in your AWS and GCP account.

Once connectors are added, you will find under the new connector configuration area the list of all your connectors. You can, by clicking on the action button :

- test the connector
- edit the connector
- delete the connector

Once at least one connector is well configured, you will be able to use the data sources menu in order to create CSV from your database or file server.

## Data sources

### data sources

- need a connector
- may be used input of a pipeline
- may be used to import dataset

Datasources represent « dynamic » datasets, whose data can be hosted on an external database or filesystem. Using a pre-defined connector, you can specify a query, table name or path to extract the data.

The screenshot shows a 'New datasource' modal window. It contains the following elements:

- Data source name:** A text input field containing 'CRM Gcp'.
- Connector:** A dropdown menu showing 'GCP'.
- Connector type:** A light blue button labeled 'SQL'.
- database:** A dropdown menu showing 'prevision'.
- select by query:** An unchecked checkbox.
- table:** A dropdown menu showing 'bodacc\_judgment\_temp'.
- Buttons:** 'Test Data source' (outlined) and 'Save Data source' (solid dark blue).

Fig. 15 – Create a new datasource from connector

A datasource is created from a connector thus you always need to select one when creating a new datasource. Depending on the type of connector, you then need to input :

- a db name and table name for SQL Databases
- a path to a file, starting from the root of your server, for Storage-like connector ( FTP, SFTP, Amazon S3 and GCP )

You can test your datasource then save it for later use

## Exporters

In the same way that Datasources are used to import data into Prevision.io, Exporters are used when you write the data generated in the platform to an external database or filesystem. They also require a connector, and have similar configuration options.

### Exporters

- need a connector
- may be use as output of a pipeline
- may be used to export dataset

When clicking on the **new exporter** button inside the Exporters tabs, you will be prompted to enter a name and select a previously created Connector :

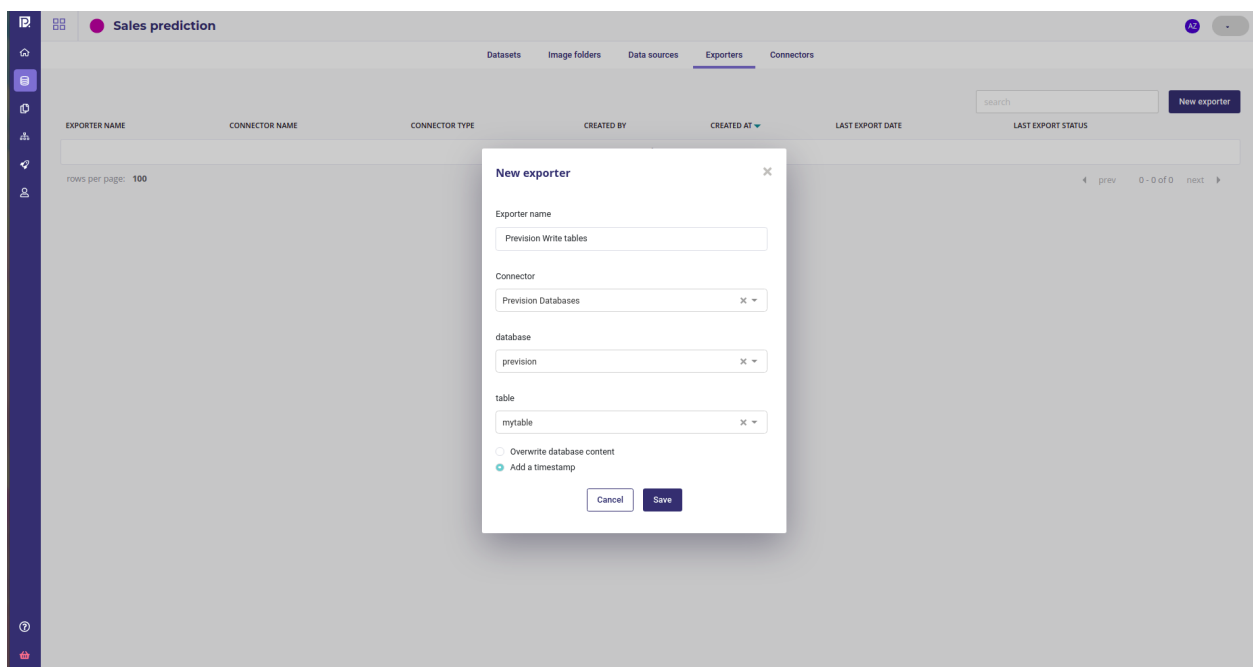


Fig. 16 – Create a exporter

When a connector is selected, depending on its type, you need to fill up more information about export destination. For example for an sql connector, you have to enter a database name and a table name.

You then have two options :

- overwriting the table each time you export data using this exporter
- create a new table with a timestamp in the name each time you export data using this exporter.

Once you clicked on the « save » button, your exporter will be available to *pipelines* as a terminal node, to push dataset or predictions into external database.

Typical examples for exporters are :

- delivering predictions to external system
- delivering transformed dataset to external system

Note that once you have an exporter, you can save any of your dataset to the exporter target from the datasets list by using the 3-dots menu



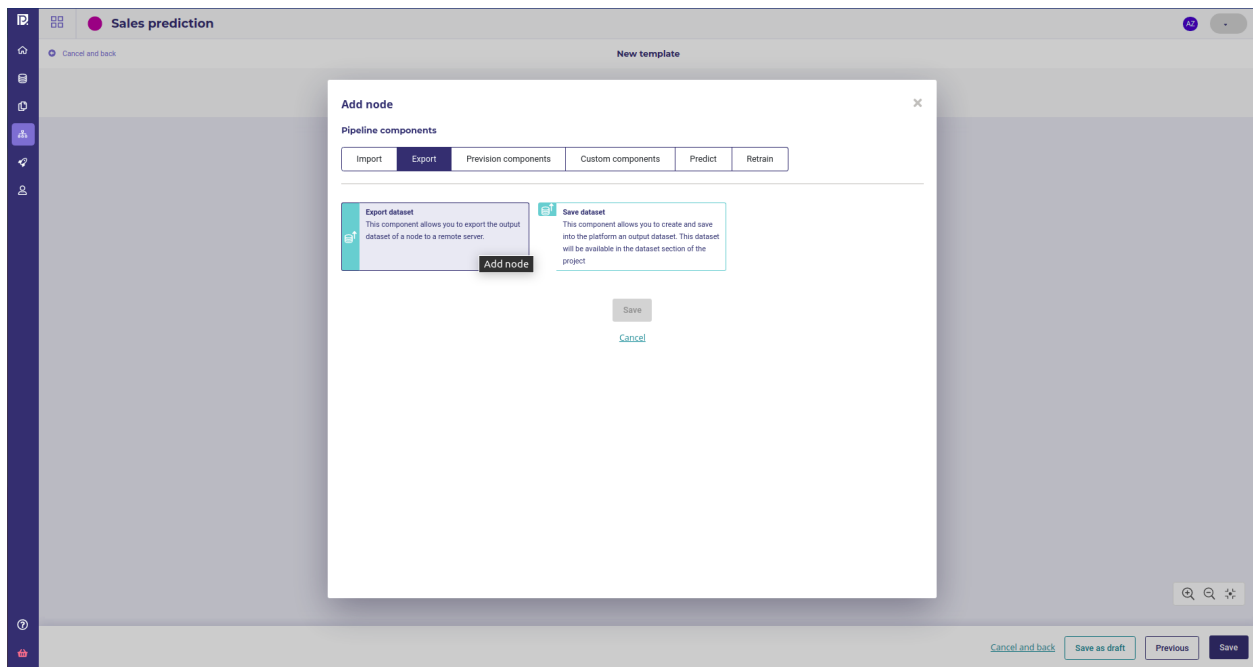


Fig. 17 – use an exporter

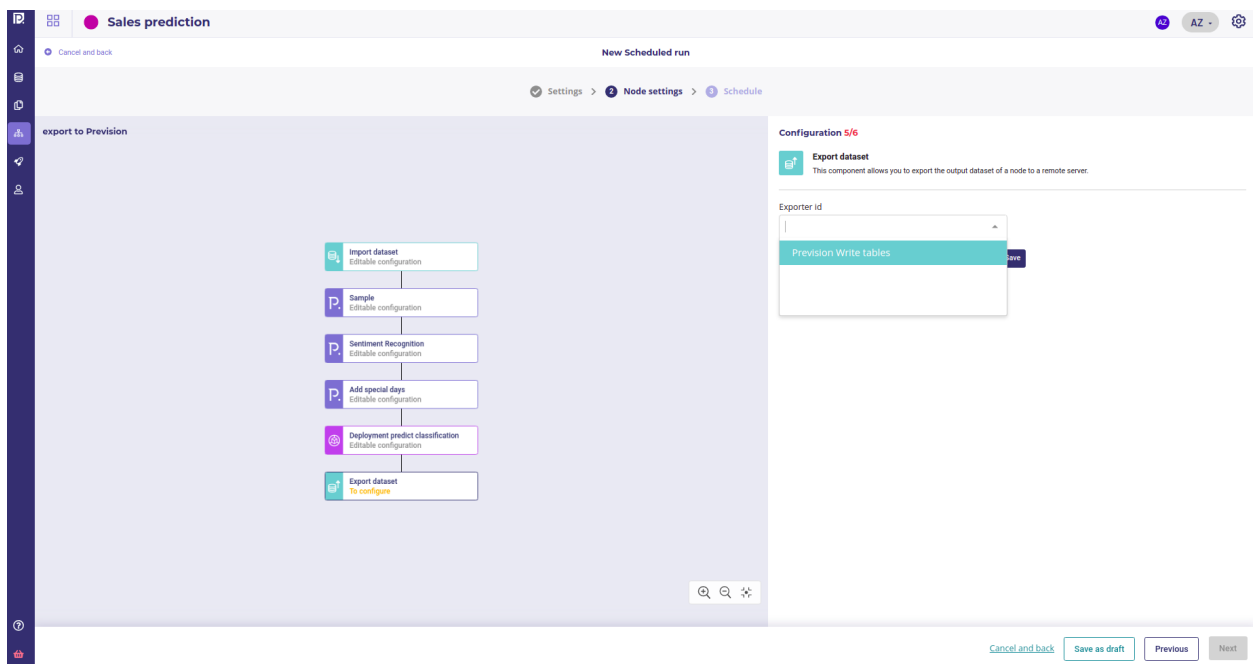


Fig. 18 – A typical pipeline : import datas, make transformations, make a predictions and export result to database

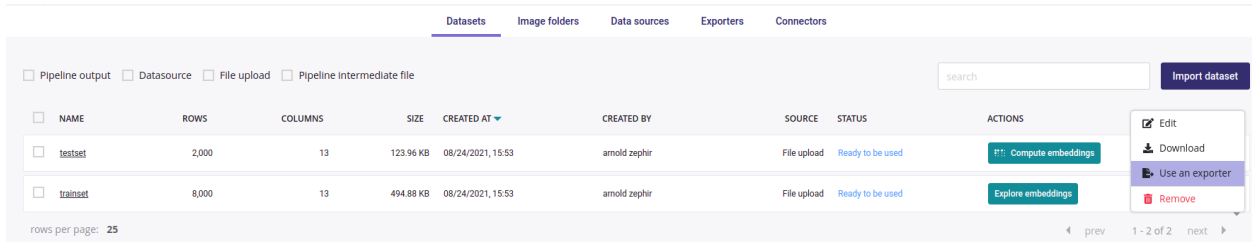


Fig. 19 – Export your dataset

### 5.3.1.3 Experiments

**Note :** An experiment represents a collection of trained *Models*, that were created either by the *AutoML Engine*, or from *External Model*. By *Versionning your experiments*, you will /studio/experiments/evaluating and compare different models against different metrics with details on predictive power of each *Features*.

This models may come from the Prevision AutoML engine or be imported in the Project as an *External Model*.

Any model of your experiments can then be used for *Deployments* as REST APIs or included in a *Pipelines*

Regarding the problematic and the data type you have, several training possibilities are available in the platform :

Tableau 2 – Experiments list and restrictions

Training type / Data type	Tabular	Time-series	Images	Available for External Model ?	Definition	Exemple
Regression	Yes	Yes	Yes	Yes	Prediction of a quantitative feature	2.39 / 3.98 / 18.39
Classification	Yes	No	Yes	Yes	Prediction of a binary quantitative feature	« Yes » / « No »
Multi Classification	Yes	No	Yes	Yes	Prediction of a qualitative feature whose cardinality is > 2	« Victory » / « Defeat » / « Tie game »
Object Detection	No	No	Yes	No	Detection from 1 to n objects per image + location	Is there a car in this image ?
Text Similarity	Yes	No	No	No	Estimate the similarity degree between two text	

### List your experiments

All your experiments are related to a *project* so in order to Create a new experiment, you need first to create a *project* .

On the homepage of your project, you can see a summary of your project ressources and access to the experiments dashboards form the left sidebar

which allows you to navigate and filter all your project's experiments from the experiments table

Each row gives you :

- the name of your experiment, that links to the experiment dashboard
- the source of the models of your experiment, AutoML or external models
- the latest version of your experiment



- the creation date and time of the experiment
- its creator ( see [Contributors](#) )
- the datatype ( [Tabular Data](#), [Images](#) or [Time series](#) )
- the training type ( Regression, Classification, Multi-classification, Object Detection or [Text similarity](#) )
- score : the choosen metrics of the last version, the type of metrics and a 3-stars evaluation
- the number of models built into the last version of your experiment
- the numbers of predictions done over the last version
- and the status ( running, paused, failed or done )

### Create a new experiment

**Note :** Experiment is a way to group several modelisation under a common target in order to compare them and track progress. An experiment may have one or more [Versionning your experiments](#) and you can change any parameter you want from version to version ( Trainset, features used, metrics,... ). The only constant between experiment are :

- the target used. Once you had selected your target, you cannot change it and muste create a new experiment if you want to try a new one
- the Engine used, AutoML or [External Model](#)

Once you had created a project, you can create a new experiment from the project Dashboard or the list of experiments, by clicking on the « Create experiment » in the bottom left corner of project dashboard or on the « New experiment » in the experiments list.

### AutoML Engine and ONNX import

The first things asked will be to choose between AutoML or external model and to give a name to your experiment. AutoML use Prevision.io engine to choose and built the best model without human intervention from the shape and type of your data. External model is the mode for importing models saved as onnx file. Both allow to evaluate, deploy and monitor your model.

The screenshot shows the 'New experiment' form in the Prevision.io interface. The form is titled 'New experiment' and has a 'Cancel and back' link at the top left. The form is divided into several sections:

- Engine:** Two buttons, 'AutoML' (selected) and 'External model'.
- Name:** A text input field with the placeholder 'Experiment name'.
- Data type:** Three buttons: 'Tabular' (selected), 'Timeseries', and 'Images'.
- Training type:** Four buttons: 'Regression' (selected), 'Classification', 'Multi-classification', and 'Text similarity'.

On the right side of the form, there is a 'What's next?' section with the following text:

**AUTOML**

The Prevision.io AutoML engine allows you to quickly benchmark and optimize a range of open source algorithms to get highly performant models.

**What do I need?**

You just need data, imported into Prevision.io as a Dataset. Your dataset just needs to contain a target column (and a temporal one, if you are working with time series), and you are good to go.

**What's next?**

Once the experiment and the models are created, you can analyze, version, and deploy them to create a prediction API endpoint, or use it with pipelines to schedule batch predictions.

At the bottom right of the form, there are two buttons: 'Cancel and back' and 'Create Experiment'.

You can then select your Data type and problem type remembering this restrictions :

Tableau 3 – Experiments list and restrictions

Training type / Data type	Tabular	Time-series	Images	Available for External Model ?	Definition	Exemple
Regression	Yes	Yes	Yes	Yes	Prediction of a quantitative feature	2.39 / 3.98 / 18.39
Classification	Yes	No	Yes	Yes	Prediction of a binary quantitative feature	« Yes » / « No »
Multi Classification	Yes	No	Yes	Yes	Prediction of a qualitative feature whose cardinality is > 2	« Victory » / « Defeat » / « Tie game »
Object Detection	No	No	Yes	No	Detection from 1 to n objects per image + location	Is there a car in this image ?
Text Similarity	Yes	No	No	No	Estimate the similarity degree between two text	

## Data type

Data type is, obviously, the type of your data :

**Indication :** Most of nlp problems should be considered as tabular data whom one or more columns are text. If any columns is detected as a textual one, Prevision AutoML engine will apply a set of standard NLP embedding technics ( tf/idf, Transformers, seq to seq...). The only limitation is that you cannot yet build generative model ( so no automatic summarisation ) but if you want to classifiante or rate docs or email, tabular data is the way to go.

As far as that goes, image are used in a tabular way too, except for the object detector. When choosing data type image, you will used a dataset whom on feature is a path to some image uploaded in your image folder. You can run Classification or Regression on Image !

- tabular : data from csv, sql database, hive database, ... suitable for classification and regression
- timeseries : when target depends on time, use a timeserie. Note that you should have data with constant timestep as much as possible and only regression are possible with timeseries
- images : if you want to build an image-based model. Note that you can mix images and standard features in the same experiment. Image has a special probleme type ( Object Detection )

## Training type

Training type is the kind of problem you want to solve :

- Regression : when you need to predict a continous value. Suitable for sales forecasting, price estimation, workforce management, ... can be used for image and text.
- classification : when target has only 2 modalities, choose classification. For example fraud detection, churn prediction, Risk management,...
- multi-classification : if your target has more than one modality. Standard example are product cross sale, Transport Mode detection, Evaluation prediction, email classification, sentiment analysis ...
- Text similarity : this training type is dedicated to retrieve doc from query. The input is tabular data with at least one column of docs ( text ) and the model will be trained to attribute later query to one of this original doc. It's useful for searching item from their description or build chatbot to answer to user questions

— Object-detection : Object detection train a model to detect some object on image, attribute a class and return a bounding box. For example you can [detect pools on satellite image](#) or [type of french cheese on a photo](#)  
You can learn more about AutoML and external models on their dedicated section :

### AutoML

Prevision.io platform can train model based on your experiment parameters. The AutoML Engine make analysis of your dataset and :

- builds the best feature engineering given your datatype ( for example : convert text and images to embedding or build lags auto on time serie )
- choose the fittest algorithm given your data
- choose the best parameters for each algorithm
- may blend and combined many model to get performance

When choosing AutoML, you can tune some configuration but the most important are those on the « Basics » tabs :

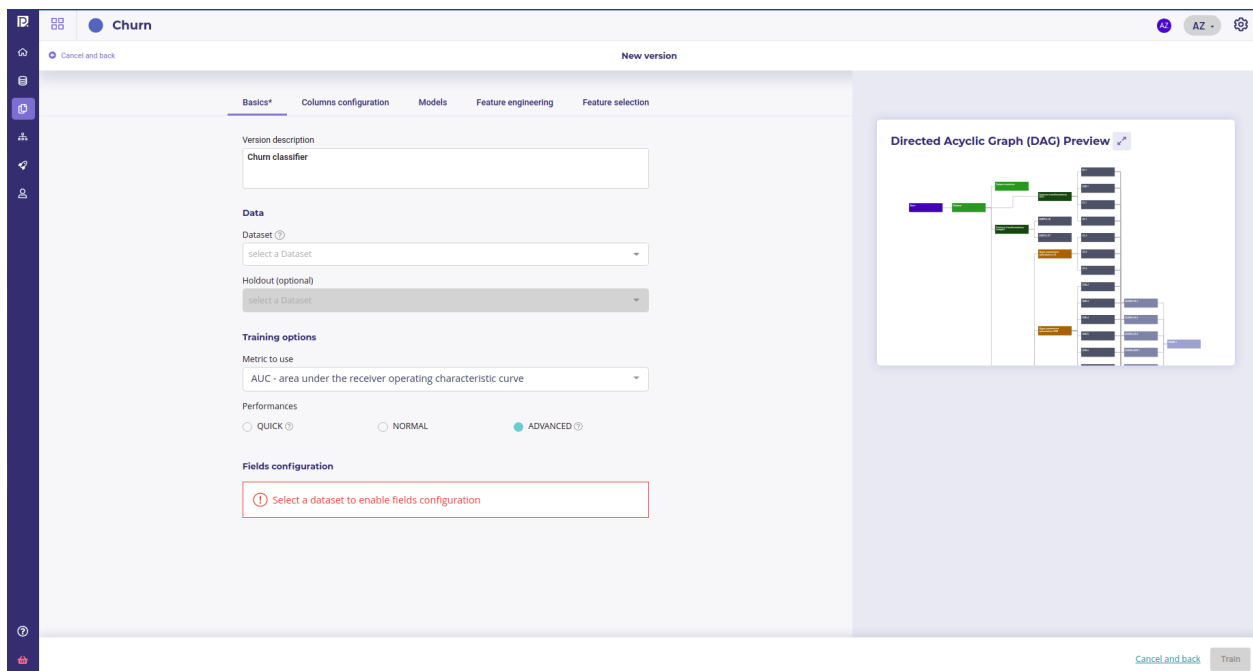


Fig. 20 – Basic configuration

- the Metric to optimise ( see [our guide on how many time you should spend on choosing the metrics](#) )
- **the performances profil :**
  - Fast : get a result as fast as possible
  - Normal : Spend more time in hyper opt
  - Advanced : get the best result ( spend many time son hyper opt and blending models )

General advice is to start project with a « Fast » profile and go for advanced train when the problem and features are completely defined.

Most of the configuration is common accross the training type and Data types except the metrics that depends on the type of problem, but some of them have specificities, especially on the metrics and the available models

Note that when you create a new experiment, you will be prompted to create a *first version*

See more about each type of training on dedicated section :

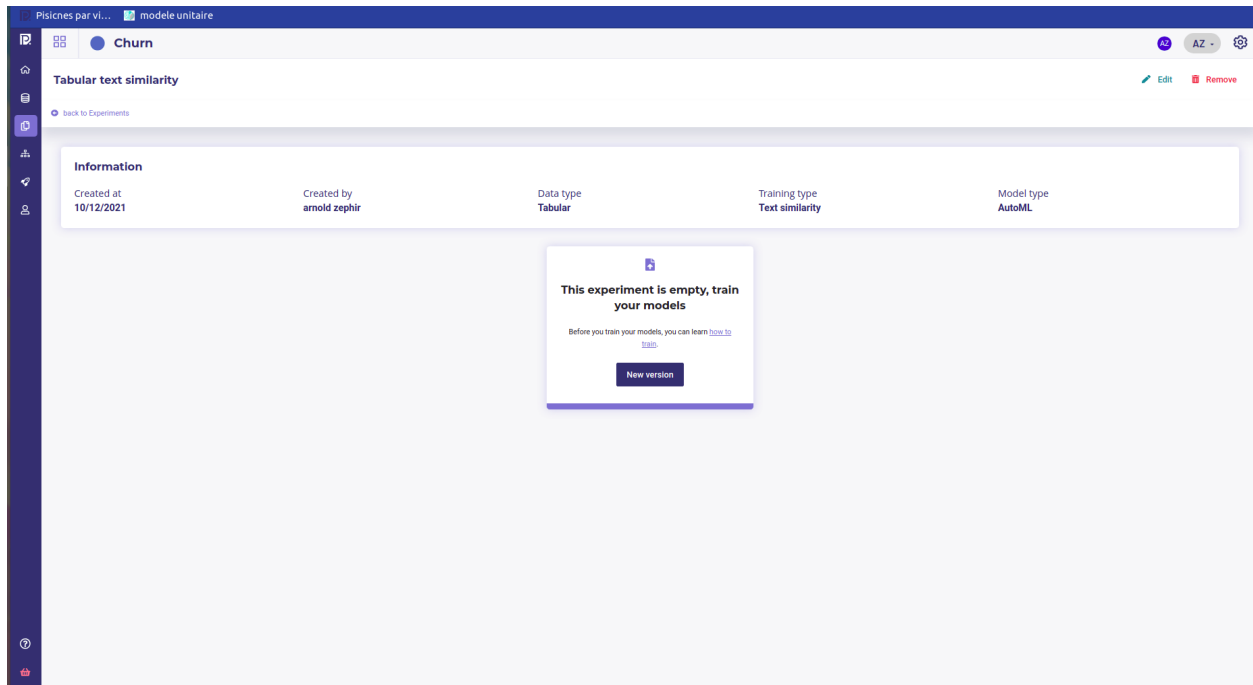


Fig. 21 – Empty experiment

## Tabular Data

Tabular data use data in columns to build model. There 3 kind of probleme type, Regression, classification, multi classification and a special one, *text similarity*

This section explains the parameters for regression, classification and multi classification. For text similarity, see the *dedicated section*

## Basics tabs

The basics tab group all the important parameters :

- dataset : the *dataset* from your datas assets to train on. This parameter is mandatory for automl
- holdout : holdout is another *dataset* that will be used to evaluate your model. It should have the same features and target than the trainset but with sample never seen in the trainset. It is optional but strongly recommended, especially at the end of your experiments, to check your model stability
- metrics : the options depends on your training type but you may choose one of the standard Machine Learning metrics for objective
- Performances : quick for qui result, Advanced for best result. Be aware that advanced option could lead to more than 24 hours of training if the trainset is big. Quick get result in less than 3 hours in most of case.
- Target : set the column of your trainset to predict. Note that the platform filter available target based on your problem type ( example : it expects the target too have 2 modalities only if the problem is a classification )
- ID Column : you can set a column as an id. It must have only unique values The column set as an id will not be used as a feature and will be repeat on subsequent prediction to serve as a join column. If you do not set an ID column, an index will be generated.
- Weight : you can set a column to be used as a Weight column for sample. The sample with large weight will be favored during the training. If you do not set weight, the system apply a balanced trainign, meaning it applies larger weight to the less frequent target modalities.

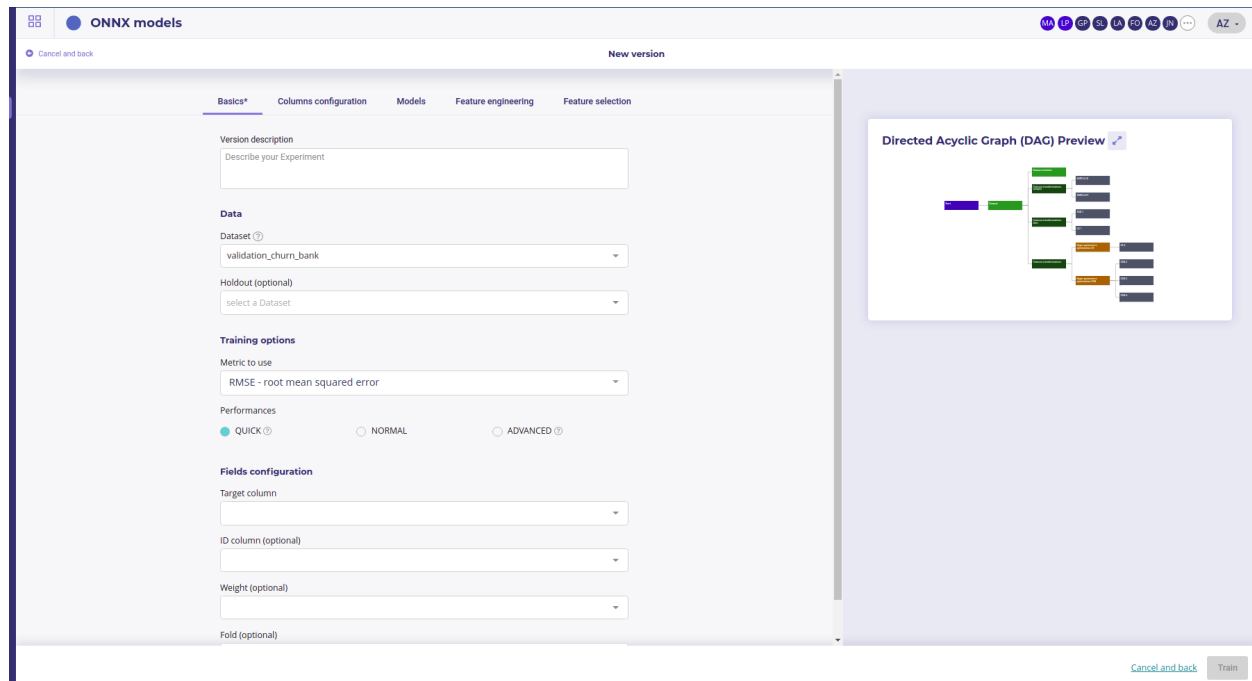


Fig. 22 – Basics configuration

- Fold : Fold column will be used to generate cross validation. If you do not know how to generate correct fold, leave this empty. Otherwise, you may put a fold number (integer) in this column and the Cross Validation will be run by splitting the dataset against this folds.

The two mandatory parameters are Dataset and Target. Once had set them up you can proceed and launch train by clicking on the **Train** button in the right-lower corner.

## Columns configuration

The second bottom left panel allows you to ignore some columns of your dataset. To do it, just deselect unwanted features. Please note that you can search by features name the columns you want to unselect and use the “select/unselect all” checkbox to apply your choice to the selection.

## Why drop features ?

In most of case, you should not drop any features. The Prevision Platform can handle up to several thousands of features and will keep those meaningful to build its models. Yet, there is two case you would drop some features :

- Some Features has too much importance in the model and you suspect that it is in fact a a covector of the target, or that you have an important leakage. If you see that your simplest model (Simple LR) perform as well as complex one and that some feature as a big feature importance, drop it
- you want to get some result fast. Dropping features allows for faster training. If you suspect that some features bears low signal, drop it at the start of your project to iterate faster

## Models

The model selection area allows you to select the type of model you want to train. You got 3 sections.



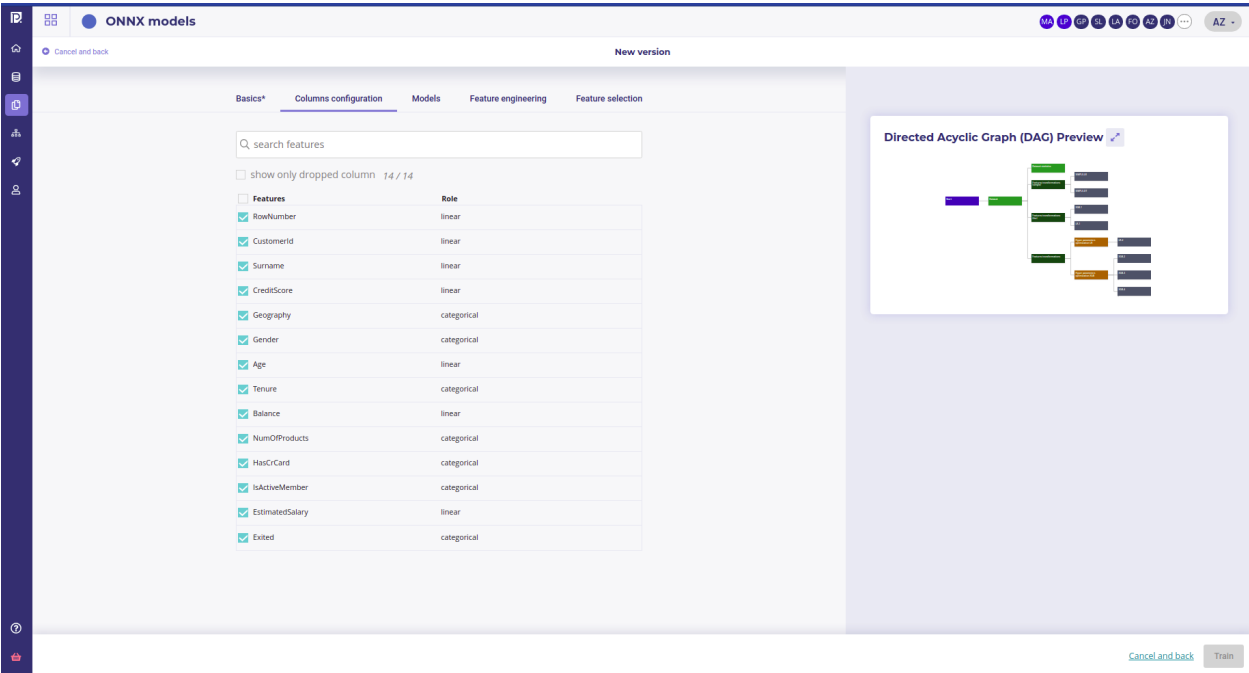


Fig. 23 – Basics configuration

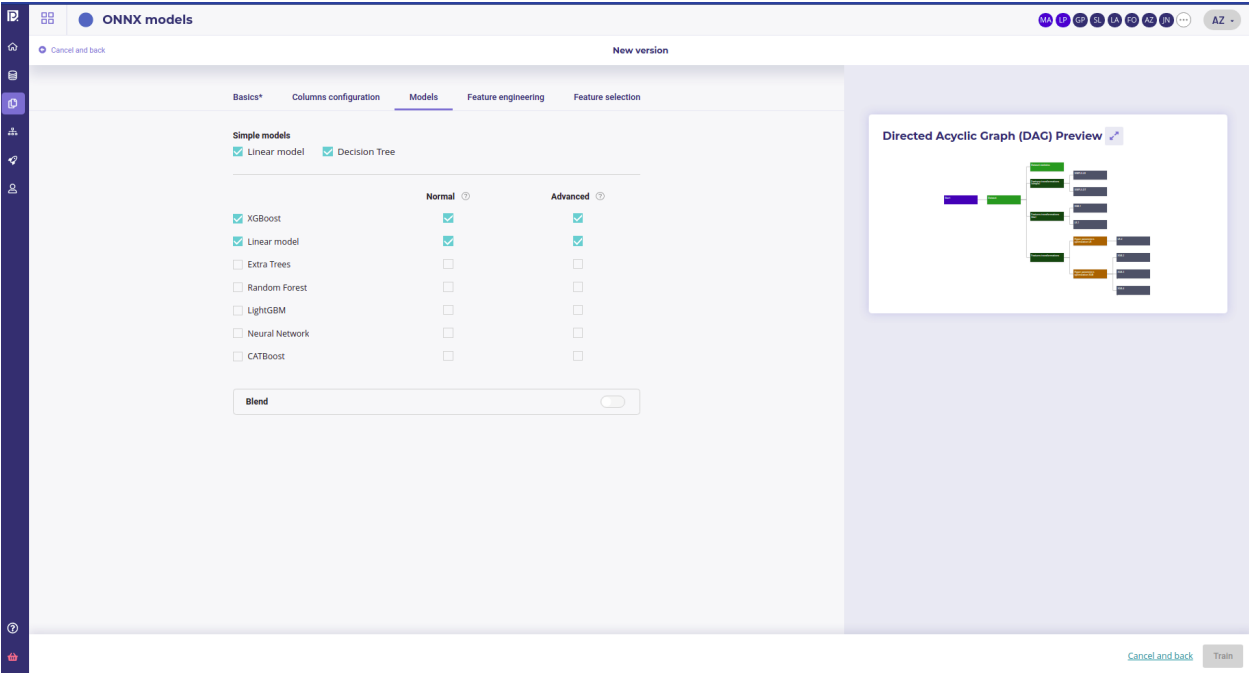


Fig. 24 – Model tab

### Simple Models

Simple models are models done with no complex optimisation and using simple algorithm, a linear regression and Random Forest with less than 5 split. They allow to check if the problem is treatable with very simple algorithm instead of fancy Machine Learning. Moreover, simple model generate :

- a Chart that explain model and is human readable
- python code to implement it
- SQL Code to implement it

You can unselect simple model but it is recommended to keep them when starting a project and watch how good they perform vs more complex model. If a simple Decision Tree performs as good as a Blend of XGBoost, or only marginally worst, favor the simple model.

### Model Selection

You can choose the algorithm type that the automl engine will use. The more you select, the longer is the train. All the selected algorithm will be used in blend

Note that only if :

- there is at least one column with text
- the probleme type is a classification or a multi classification

you can select Naive Bayes Model

### Blend

Blending model is a powerful technique to get the most performance yet it can be quite long to train. Blend use a model over all the others to merge and combine them in order to get the best performance. Switch the option if you want to blend your models but be aware that resulting train will last very long.

### Feature engineering

In this section, you could select more feature engineering ( or unselect some ).

Four kinds of feature engineering are supported by the platform. :

- 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 :
  - Statistical analysis using Term frequency-inverse document frequency (TF-IDF). Words are mapped to numerics generated using tf-idf metric. The platform has integrated fast algorithms making it possible to keep all uni-grams and bi-grams tf-idf encoding without having to apply dimension reducing. More information about TF-IDF on <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>
  - Word embedding approach using Word2Vec/Glove. Words are projected a dense vector space, where semantic distance between words are : 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](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\\_\(machine\\_learning\\_model\)](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)). The Python Package used is Sentence Transformers ([https://www.sbert.net/docs/pretrained\\_models.html](https://www.sbert.net/docs/pretrained_models.html))

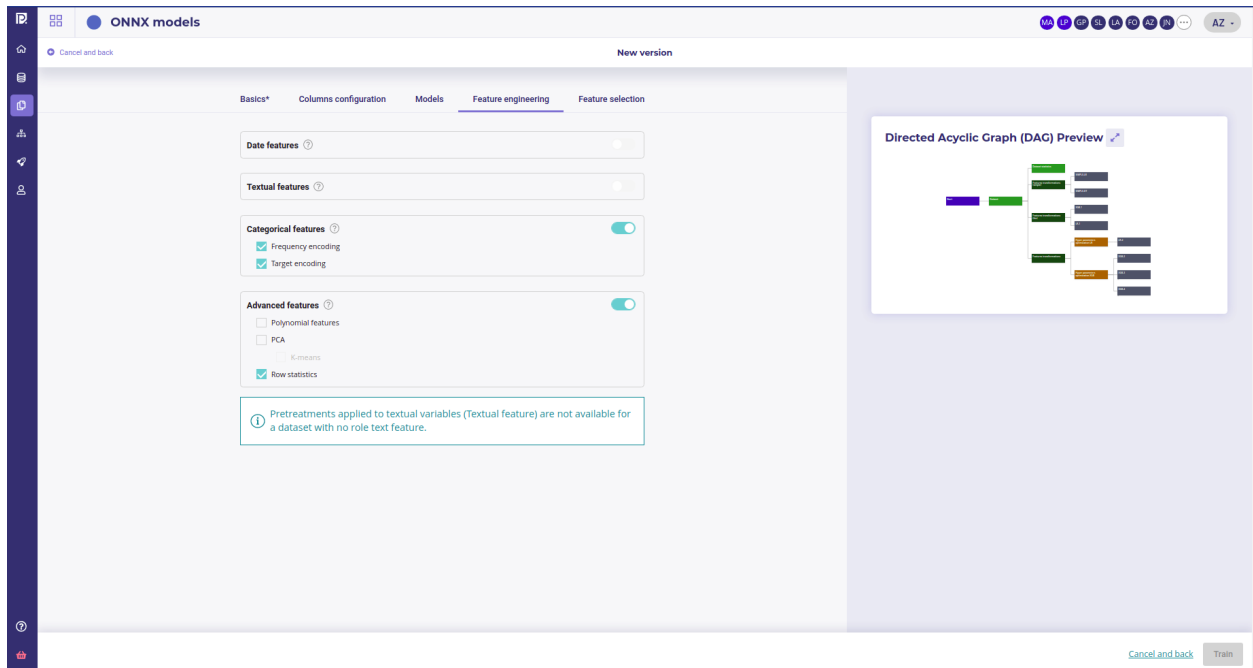


Fig. 25 – Model tab

- 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, ...)

Please note that if you don't have a feature of one of these feature types in your train dataset, the corresponding feature engineering toggle button will be disable. Also please note that textual features pretreatments only concerne advanced models and normal Naive Bayes model

## Feature selection

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.

The variable selection strategy in Prevision.io is hybrid, depends on the characteristics of the dataset and the time available.

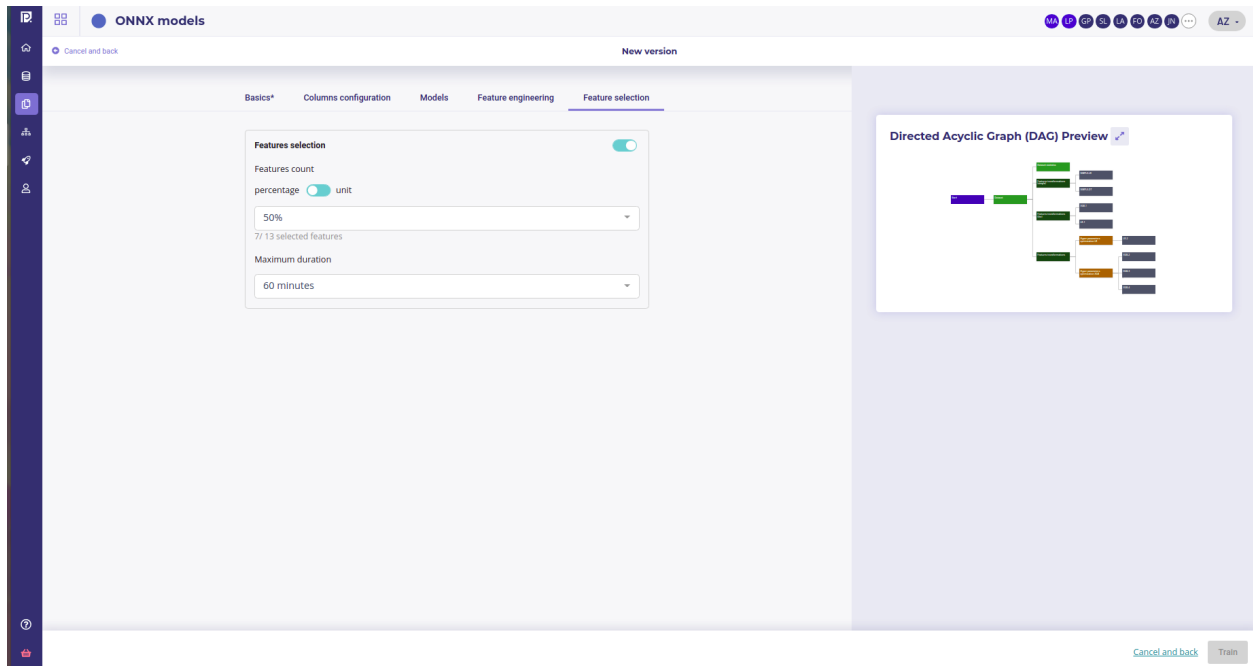


Fig. 26 – Feature Selection

1. It is hybrid because it combines both so-called filtering methods, encapsulation methods and integrated methods. The filtering methods perform the selection of entities independently of the construction of the classification model. Encapsulation methods iteratively select or eliminate a set of entities using the metric of the classification / regression model. In built-in methods, feature selection is an integral part of the classification / regression model.
2. It depends on the characteristics of the dataset and the time allotted. In fact, depending on the volume of the dataset, a small data strategy is applied for a dataset of less than 8 GB, fully in memory. Otherwise, a big data strategy is applied.
3. In a small data situation, a first filtering approach is carried out consisting in filtering the variables of zero variance, the duplicated variables, the intercorrelated variables beyond 99% and the variables correlated to the target variable beyond 99% . Depending on the time remaining available, a second so-called encapsulation method is carried out using a LASSO-type regularization on the entire dataset by cross validation with the aim of optimizing the metric selected when the use case is launched.
4. In a big data situation, as time permits, several row and column samplings are carried out and the stages of filtering, encapsulation method and integrated methods completed by a reinforcement learning strategy are successively launched. . The variables are then ranked in order of priority according to the different approaches tested and the top variables, at the threshold defined by the user, are sent to the various algorithms of Prevision.io.

## Text similarity

Even if considered as a training type for tabular data type, text similarity experiments are particular and need specific training options. Text similarity models allow to retrieve textual documents from a query. For example, from « Red shoes for girls » query, your model should return a corresponding item.

Fig. 27 – Text Similarity parameters

## Creating a Text similarity Model

In order to train a text similarity model you **must have** a trainset ( dataset dropdown menu ) with :

- a description column : some column with text that describes items you want to query (*Description column* dropdown menu )
- an id column : only column with unique ID could be selected (*ID column*)

To get better evaluation you **should have** a query dataset ( queries dropdown menu ) with :

- a textual column containing user queries that should have match with some item description ( *query column* dropdown menu)
- a column with the id of the item whose description should have match the query (*Matching ID column in the description dataset* dropdown menu)

Your queries dataset **could have** its own ID column (*ID Column* dropdown menu )

Note that the drop down to select column only appear when you had selected Dataset and/or a Queries

You then have to select a metric :

- **Accuracy at k** : Is the real item corresponding to a query present in the search result, among the k items returned ? The value is a percentage calculated on a set of queries.
- **Mean Reciprocal Rank (MRR) at k** : Similar to accuracy at k. However the score for each query is divided by the rank of appearance of the corresponding item. Example : If for a query the corresponding item appears in third position in the returned list, then the score will be  $\frac{1}{3}$ . If it appears in second position the score will be  $\frac{1}{2}$ , in first position the score will be 1, etc. [https://en.wikipedia.org/wiki/Mean\\_reciprocal\\_rank](https://en.wikipedia.org/wiki/Mean_reciprocal_rank)
- **K results** : the number of query like items that the tool must return during a search. Value between 1 and 100.

## Text similarities models options

Text similarity module has its own modeling techniques and is composed of 2 kinds of models :



- embedding model to make a vector representation of your data
- search models to find proximity between your queries and product database

**quora\_items\_short** [Edit](#) [Compute embeddings](#) [Download](#) [Remove](#)

[Back to list](#) [General](#) [Columns](#) [Sample](#)

ITEM_DESC	ITEM_ID
A direct competitor of mine in a geographically based service market approached me to buy their business. What should I think of this?	251,113
Who is the best TV actor/actress?	77,217
Which is better in term of Performance & Power, laptop i7 or desktop i7 for Graphic designing & Website Designing?	116,204
Why does Ladder Logic fall short when tasks become complex and why isn't it the best for modular PLC programming?	185,384
Was Hitler a Nazi?	138,632
What is my vocal type? I'm a male and my vocal range is A2-C6?	480,557
Is it possible to buy a 30 lakh car for a 24 year old with a current salary of 15k?	258,133
When you are an adult, what do you call a girlfriend?	246,590
How can I reduce breast size?	217,402
Does 1,000000 exist?	328,290

Fig. 28 – A dataset with items and their description

  **nlp**

**quora\_queries**

[Back to list](#) [General](#) [Columns](#) [Sample](#)

TRUE_ITEM_ID	QUERY
241,981	Is a PhD in chemistry worth pursuing?
96,178	Who is the most badass footballer ever?
429,697	What are things which I can export from India?
532,049	What is the best question and answer platform?
184,530	What are some good books for web developers to learn about design?
95,514	What are some activities that people (adults) with various disabilities can all enjoy?
395,794	Why is my Yahoo! sign in failing?
67,925	What are some examples of controversial topics in special education?
150,661	How many times do you have to smoke weed before you get high?
404,641	Can I do an MBA after BE?

Fig. 29 – A dataset with user queries and the item id that should have match

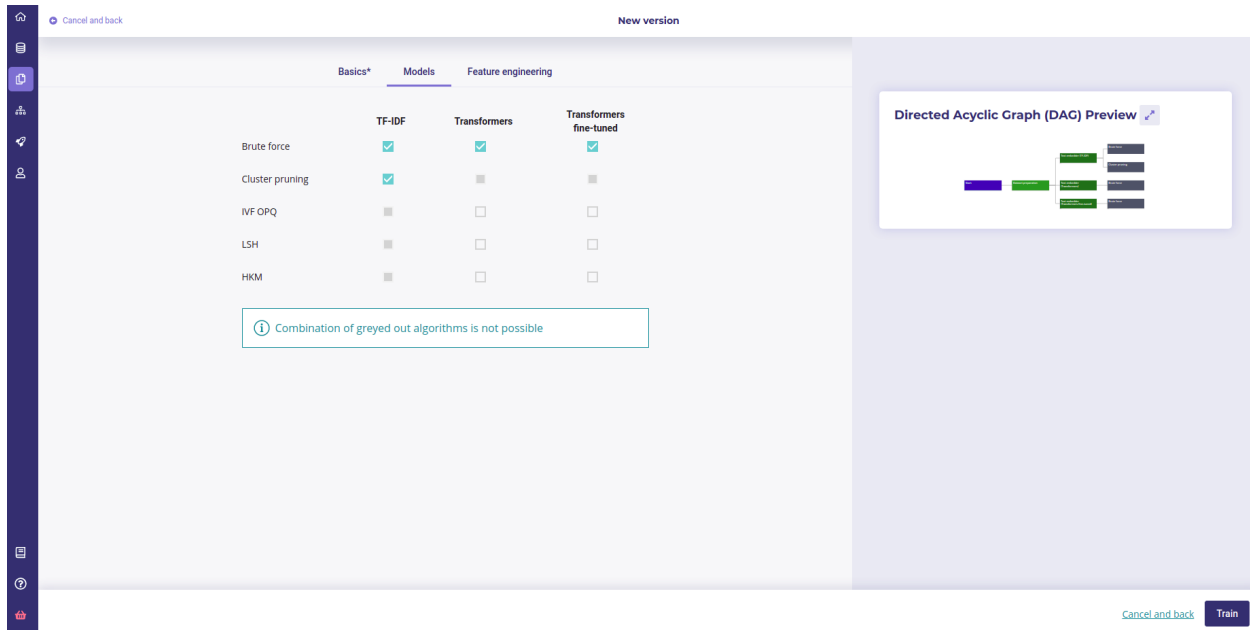


Fig. 30 – Available algorithm for text similarity models

## Embedding model / word vectorization

**Term Frequency - Inverse Document Frequency (TF-IDF)** : Model representing a text only according to the occurrence of words. Words not very present in the corpus of texts will have a greater impact. <https://fr.wikipedia.org/wiki/TF-IDF>

**Transform** : Model representing a text according to the meaning of words. In particular, the same word will have a different representation according to the other words surrounding it. [https://en.wikipedia.org/wiki/Transformer\\_\(machine\\_learning\\_model\)](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model))

**Transformer feature-based** : Transformer that has been trained upstream on a large volume of data, but has not been re-trained on the corpus in question.

**Fine-tuned transform** : A transform that has been trained on a large volume of data and then re-trained on the text corpus in question.

## Search models

**Brute Force** : Exhaustive search, i.e. each query is compared to the set of item descriptions.

**Locality sensitive hashing (LSH)** : exhaustive search. Vectors are compressed to speed up distance calculations. [https://fr.wikipedia.org/wiki/Locality\\_sensitive\\_hashing](https://fr.wikipedia.org/wiki/Locality_sensitive_hashing)

**Cluster Pruning** : non-exhaustive research. Item descriptions are grouped by cluster according to their similarity. Each query is compared only to the queries of the closest group. <https://nlp.stanford.edu/IR-book/html/htmledition/cluster-pruning-1.html>

**Hierarchical k-means (HKM)** : non-exhaustive research. The idea is the same as for the previous model, but the method used to group the items is different.

**Inverted File and Optimized Product Quantization (IVFOPQ)** : non-exhaustive search. The idea is the same as for the two previous models, but the method used to group the items is different. Vectors are also compressed to speed up distance calculations.

Please note that in order to guarantee the performance of IVF-OPQ models, a minimum of 1000 unique IDs in the train dataset is required.

### Text similarity Preprocessing

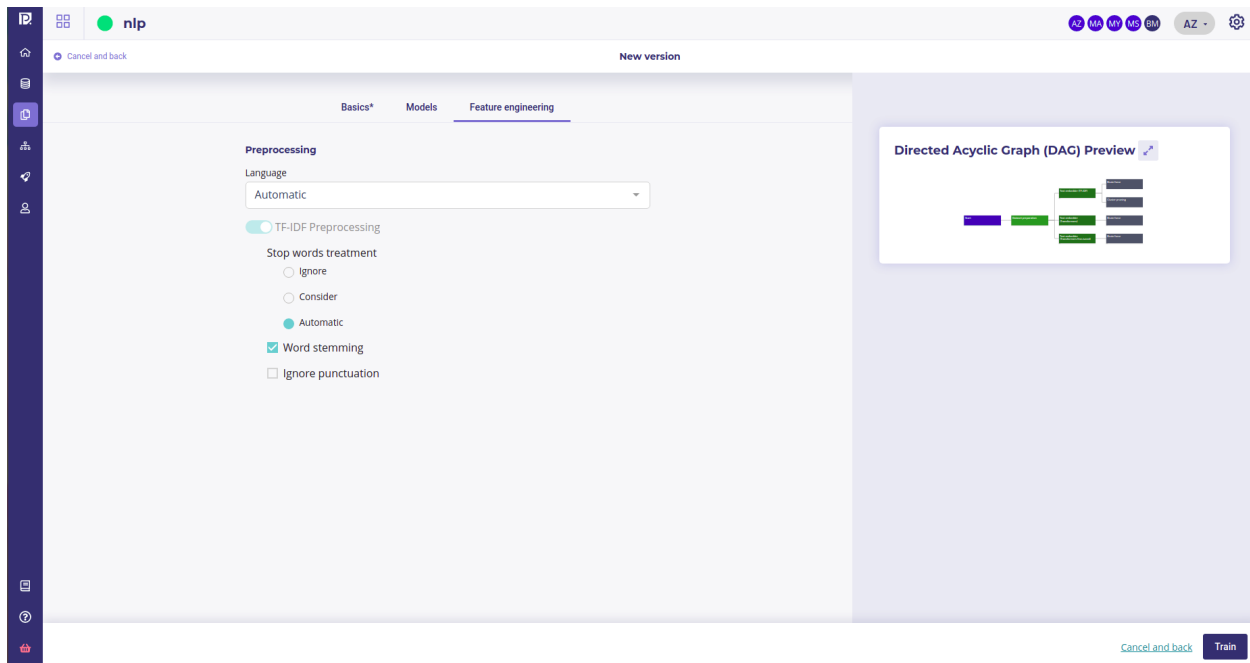


Fig. 31 – Feature engineering for text similarities

Several preprocessing options are available :

- Language : you can force the training dataset language to english or french or, let the platform determines by itself between these two languages
- Stop words treatment : you can choose if the platform has to ignore or consider the stopwords during the training. As for the language, you can also let the system makes it own decision by selecting “automatic”
- Word stemming : **stemming** is the process of reducing inflected (or sometimes derived) words to their **word stem**, base or **root** form—generally a written word form.
- Ignore punctuation : by activating this option, the punctuation will not be considered during the training

### Watching my Text similarity experiments

Text similarity experiments will be available in the same way that standard tabular data experiments, by clickin on it the list of your experiments yet it has some specificities.

First one, when you select a model, is the model evaluation chart. It shows the performance evolution along the expected rank

Second difference is the prediction tab, which is slightly different from other :

### Time series

In the prevision.io platform you have the possibility to train time series experiment in order to do forecasting predictions. By selecting in the new experiment screen the timeseries data type you will access the timeseries experiment



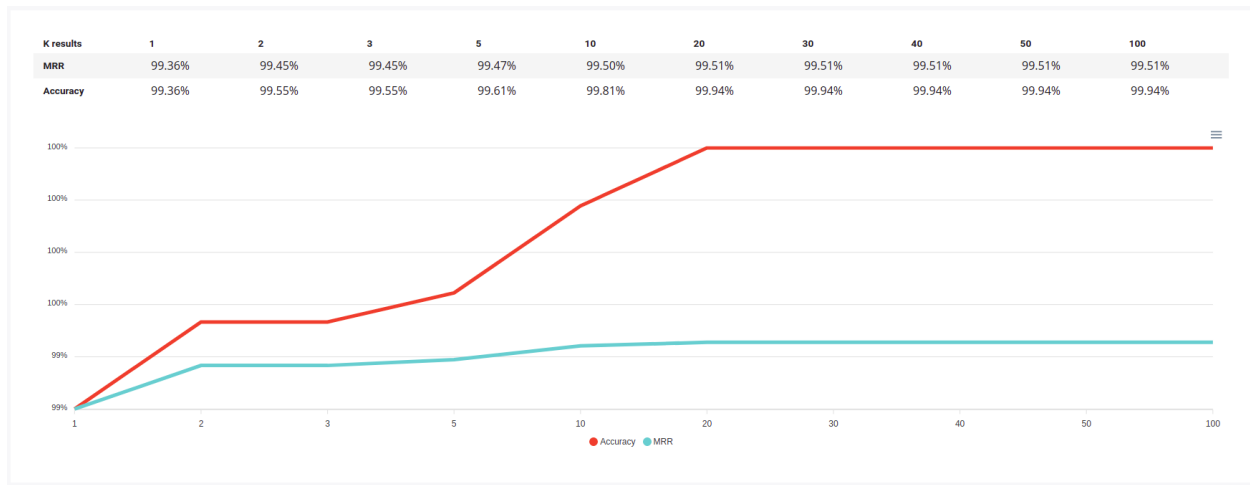


Fig. 32 – Text similarity models

**Prediction Bulk**

SELECT A MODEL:

SELECT A DATASET:

K RESULTS:

DESCRIPTION COLUMN:

MATCHING ID COLUMN (OPTIONAL):

ID COLUMN (OPTIONAL):

**User Generated Predictions**

<input type="checkbox"/>	NAME	CREATED AT	MODEL	SCORE	VALIDATION SCORE	ROW COUNT	DURATION	CREATED BY	STATUS
<input type="checkbox"/>	Netflix_Catalog	10/19/2021, 10:51	<a href="#">brute_force-tf_idf</a>	0.9994				arnold zephir	<input type="radio"/>

rows per page: 25

1 - 1 of 1

Fig. 33 – Txt similarity predictions

configuration.



studio/experiments/automl/img/slected\_ts.png


### Timeserie experiment configuration

Time series is very similar to tabular experiment 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 contains 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](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) \* End 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 stamp 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)



studio/experiments/automl/img/ts\_fields.png

Please note that advanced options work the same way than for tabular experiments. Please read the corresponding readthedoc section in order to configure your time series experiment.

### Images

In the prevision.io platform, you can train several experiments type using images :

regression classification multi-classification image detection

For the first three kinds of training, the user flow is similar to the corresponding tabular experiments. You will just need in addition to add an image folder corresponding to the train dataset in the new experiment screen.

The screenshot shows the Prevision.io configuration interface with three main sections:

- 1. DATA TYPE:** Contains three options: Tabular (with a bar chart icon), Timeseries (with a line graph icon), and Images (with a photo icon). The 'Images' option is highlighted with a blue border.
- 2. TRAINING TYPE:** Contains five options: Regression (with a line graph icon), Classification (with a scatter plot icon), Multi-Classification (with a scatter plot icon), and Object-Detection (with a bounding box icon). The 'Regression' option is highlighted with a blue border.
- 3. NAME & DATASET SELECTION:** Contains a 'Next step' button and three input fields:
  - Usecase name:** A text input field with the placeholder 'set usecase name'.
  - Usecase description (optional):** A text input field with the placeholder 'set usecase description'.
  - Dataset ?:** A dropdown menu with the placeholder 'select a Dataset'.
  - Image Folder:** A dropdown menu with the placeholder 'select a Folder'.

## Image experiments configuration

The image experiments training workflow is similar to the tabular corresponding experiments (except for image detection). Please refer to the tabular datatype training read the doc section in order to get information about the train settings. However this similarity, some differences are notable. On the field configuration options, an image path is required. This image path is the link between the tabular train dataset and the corresponding images.

The screenshot shows a 'Fields configuration' dialog box with two input fields:

- Target column:** A dropdown menu with a downward arrow.
- Image path:** A dropdown menu with a downward arrow.

## Image detection experiments

In the prevision.io platform, a particular kind of image experiment allows you to train models that are able to recognize and boxing on an image a particular object.

In order to train image detection experiments you will need to have an image folder and a tabular document including :

- the image path
- the object label
- the bounding box coordinates

The image detection experiment training configuration is simpler than for other training. In advance options, only training performances choice between quick, normal & advanced is available.

### Fields configuration

Image path

Object class column

Top

Bottom

Left

Right

## External Model

When running some *Experiments* you can either use the Automl engine over the data connected or upload your own pretrained models.

Both will benefit from same Prevision.io features :

- evaluation,
- batch prediction,
- deployment,
- pipeline integration,
- monitoring

External model import uses the *Standardized ONNX Format* and most of the standard ML library have module for export :

Tableau 4 – Onnx exporter for standard lib

library	exporter	link
sklearn	sklearn-onnx	<a href="http://onnx.ai/sklearn-onnx/">http://onnx.ai/sklearn-onnx/</a>
Tensor-flow	tensorflow-onnx	<a href="https://github.com/onnx/tensorflow-onnx/">https://github.com/onnx/tensorflow-onnx/</a>
Pytorch	torch-onnx	<a href="https://pytorch.org/docs/stable/onnx.html">https://pytorch.org/docs/stable/onnx.html</a>
XGBoost	sklearn-onnx	<a href="http://www.xavierdupre.fr/app/sklearn-onnx/helpsphinx/auto_tutorial/plot_gexternal_xgboost.html">http://www.xavierdupre.fr/app/sklearn-onnx/helpsphinx/auto_tutorial/plot_gexternal_xgboost.html</a>
XGBoost	onnxmltools	<a href="https://github.com/onnx/onnxmltools">https://github.com/onnx/onnxmltools</a>
LightGBM	skl2onnx	<a href="http://www.xavierdupre.fr/app/sklearn-onnx/helpsphinx/auto_tutorial/plot_gexternal_lightgbm.html">http://www.xavierdupre.fr/app/sklearn-onnx/helpsphinx/auto_tutorial/plot_gexternal_lightgbm.html</a>
LightGBM	onnxmltools	<a href="https://github.com/onnx/onnxmltools">https://github.com/onnx/onnxmltools</a>
CatBoost	onnxmltools	<a href="https://github.com/onnx/onnxmltools">https://github.com/onnx/onnxmltools</a>

If you prefer reading code than document to kickstart your project, you may use this *provided boilerplate* . This code builds a basic classifier and creates the needed files to use with Prevision Platform :

- a Trainset for trying Automl
- an holdout file to evaluate each iteration of your experiments
- an onnx file
- a yaml configuration file ( see below )

```
git clone https://github.com/previsionio/prevision-onnx-templates.git
cd prevision-onnx-templates
python3.8 -m venv env
source env/bin/activate
pip install -r requirements.txt
python sktoonnx.py
```

## Constraints

Current version of Prevision Platform only supports *Tabular Data* and only for classification, regression and multi-classification (no *Text similarity* nor *Images* )

## Prerequisites

For importing a model as an experiments you should at least provide :

- an holdout file to evaluate the model. This holdout must have the same features and target column than the model was trained on.
- the onnx file of the model ( it must be a classification, a regression or a multiclassification )
- a config file in the yaml format

You could provide a trainset too. The trainset is going to be used for computing drift ( and thus, expected distribution of features and target in production ) if you deploy your model. If you provide both a trainset and holdout :

- score is computed from the holdout
- drift is computed from trainset

---

**Note :** You could, and probaly should, import many onnx models in one experiment in order to compare them side by side.

---

### How to get the Onnx File ?

To get you onnx file, you need first to build a model, with standard Machine Learning Frameworks like sklearn or XGBoost, and then export them with exporter modules.

You can get any onnx file from anybody or any tools as long as :

- a config file describing the inputs is provided
- the output of the model is an array of string (binary classification or multi classification) , an array of numbers ( regression ) of both (a multi classification )

For example :

```
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from skl2onnx import convert_sklearn
from skl2onnx.common.data_types import FloatTensorType

clf = RandomForestClassifier(max_depth=50, verbose=1, n_estimators=200, max_
    ↳ features=1)
clf.fit(X_train, y_train)

initial_type = [('float_input', FloatTensorType([None, np.array(X_train).shape[1]]))]
onx = convert_sklearn(clf, initial_types=initial_type)

with open(join(OUTPUT_PATH, "classif_fraud.onnx"), "wb") as f:
    f.write(onx.SerializeToString())
```

### How to make the yaml config File ?

The config file is a standard yaml file.

You must always provide the list of the names of the inputs and, if the model is a classifier, you must provide the name of the class too.

Beware that the name of the class will be cast as a string so if your class are 1.0 and 0.0 ( because some int has been converted ) , the name of the class must be « 1.0 » and « 0.0 »

```
---
class_names:
- A
- B
input:
```

(suite sur la page suivante)

(suite de la page précédente)

```

- DeviceType_desktop
- DeviceType_mobile
- "Code Produit_C"
- "Code Produit_H"
- "Code Produit_R"
- "Code Produit_S"

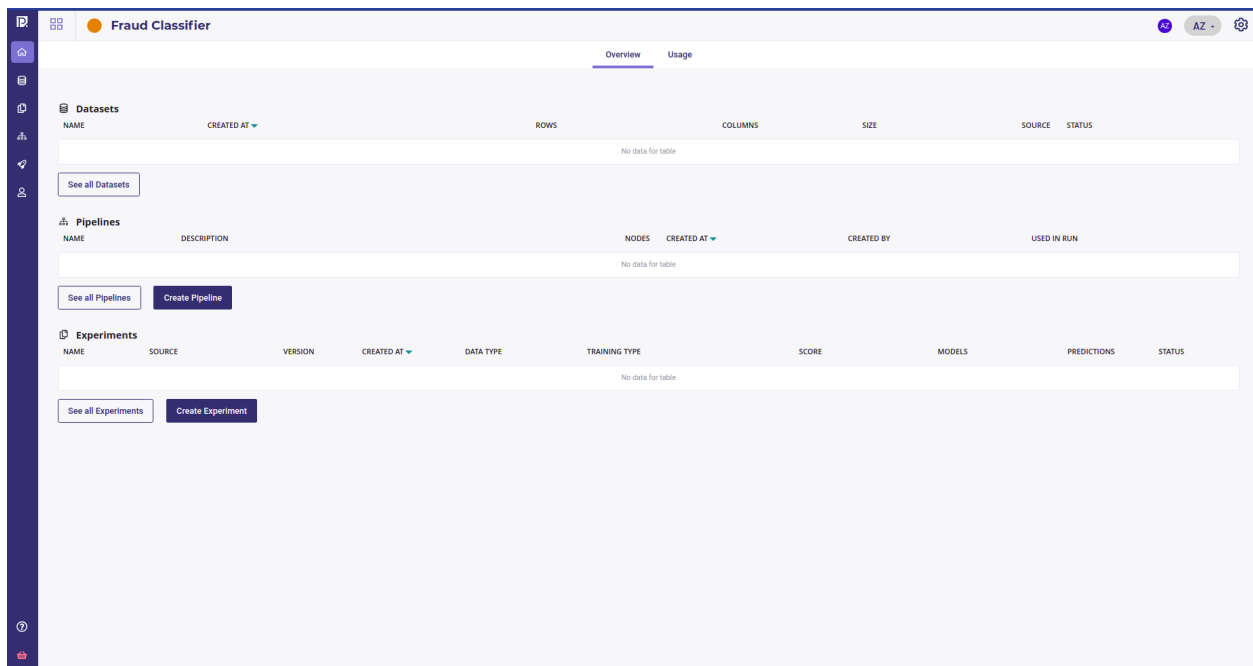
```

You could refer to the [Prevision.io Github](#) to see how to generate an yaml config file from your model and training data.

## Importing an External Model

Once you got all the required assets ( holdout file, onnx file and yaml file), you can launch a new experiment in your project.

Go to your project homepage and create a new experiment with the « Create experiment » button :



In order to import your model, select « External model ». The Timeseries, Images and Text similarity options will be muted as they are not supported yet. Assign a name to your experiment and click on « Create experiment »

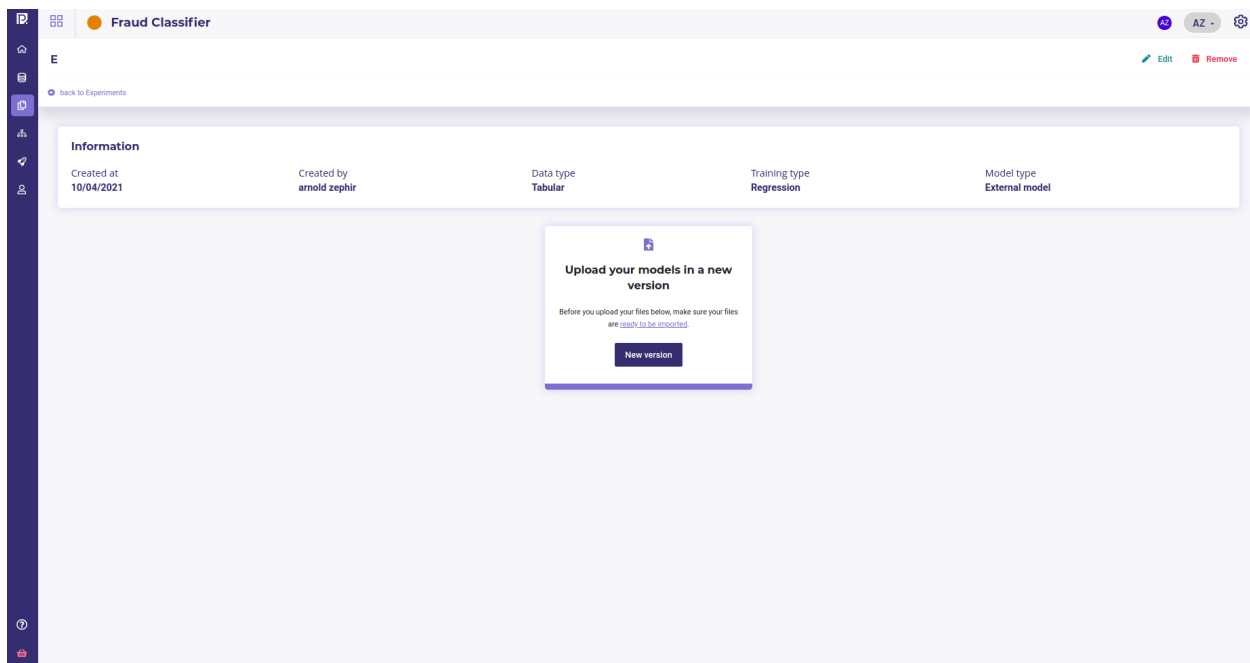
When you create an experiment, you need to create a first version ( see [Versionning your experiments](#) )

Describe your version and select :

- the holdout file to evaluate
- the metrics to use for evaluation
- the target ( only when you create the first version. Target will always be the same in all version of your experiment )

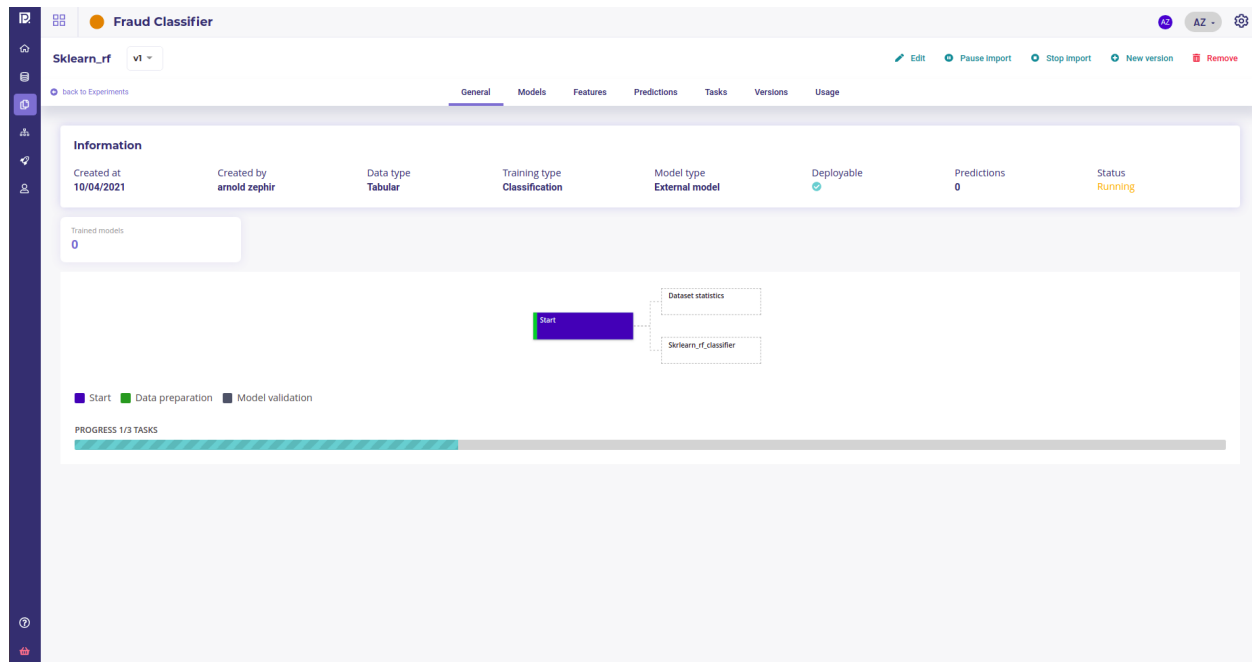
On the next screen you will be asked to provide an onnx file and a yaml file **for each model your upload**. You can upload as much model as you want as long as they have the same target.

When all ressources are uploaded, you can click on the create button on the bottom right of the screen. All the uploaded models will be evaluating on the holdout dataset upon the metrics you selected before.









In a few minutes, the experiments should be available on the experiment page.

Now you can `/studio/experiments/evaluating` and proceed to [Deployments](#)

And get a more details about each Problem type :

Whatever your choice, when you create a new experiment, you will be prompt to create the first *version* of it.

When clicking on **new version**, you will enter the configuration screen, that depends on the engine you choose

For the first version only, you need to fill the target field to set the common target for all versions of your experiment.

Once every mandatory parameters are fill ( see each training type doc for an explanation of parameters ), you can click on « train » to launch a train and start modelisation.

## Import External models into your experiment

If you already have models built form others frameworks, so called *external models* you can import them to benefit from PrevisionIO evaluation and monitoring tools.

See the [dedicated page](#)

## Inspect your experiment and evaluate your models

Once an experiment has at least one version, you can get some details about it on its corresponding dashboard by clicking on its name in the list of experiments.

## Models

By clicking on the models menu of top experiment navigation, you will access the model list trained for this experiment version. you will also, at the bottom of the page, find information regarding the model selected for this train.

5.3. Cloud & freetrial limitations	51
------------------------------------	----

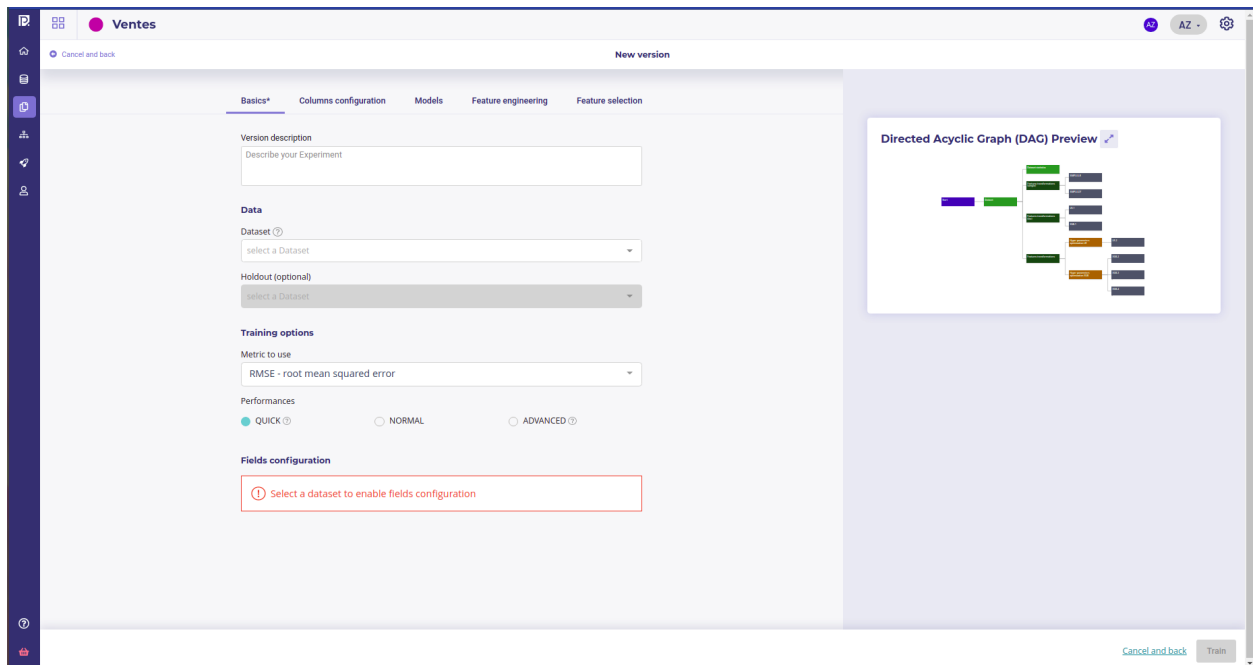


Fig. 34 – automl configuration screen, with the graph of tasks that is gonna be executed

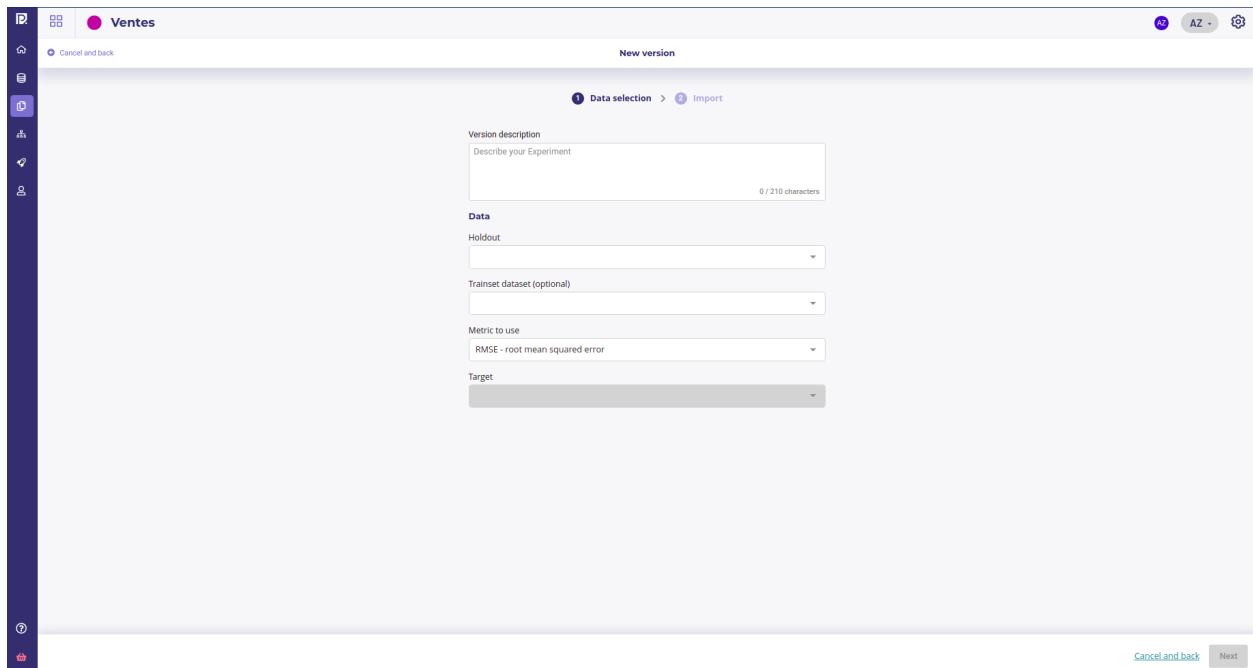


Fig. 35 – The *external model* configuration screen

Triplet more v1 Edit New version Remove

back to Experiments General Models Features Predictions Tasks Versions Report Usage

MODELS DONE

NAME	TECHNOLOGY	TYPE	SCORE	TRAINING DURATIONS	PREDICT DURATIONS	ARRIVAL TIME
CB-1 <small>Best performance</small>	CATBoost	Base	7,911 ± 2,516	12m 34.6s	48ms	10/20/2021, 14:22:02
LGB-1	LightGBM	Base	8,008 ± 2,454	1m 10.4s	93ms	10/20/2021, 14:12:08
XGB-1	XGBoost	Base	8,204 ± 2,473	4m 10.5s	81ms	10/20/2021, 14:18:04
XGB-2	XGBoost	Base	8,424 ± 1,040	8m 0.2s	311ms	10/20/2021, 14:26:23
LGB-2	LightGBM	Base	8,456 ± 1,069	3m 34.5s	255ms	10/20/2021, 14:18:32
RF-1	Random Forest	Base	8,544 ± 2,603	4m 37.7s	224ms	10/20/2021, 14:15:45
XGB-3	XGBoost	Base	8,577 ± 1,361	9m 50s	292ms	10/20/2021, 14:42:53
CB-4	CATBoost	Base	8,588 ± 1,468	23m 11.9s	262ms	10/20/2021, 16:09:14
XGB-4	XGBoost	Base	8,729 ± 628	9m 28s	237ms	10/20/2021, 14:39:21
LGB-4	LightGBM	Base	8,785 ± 1,116	4m 8.2s	273ms	10/20/2021, 14:24:45
CB-3	CATBoost	Base	8,902 ± 1,476	24m 43s	481ms	10/20/2021, 16:05:36
LGB-3	LightGBM	Base	8,951 ± 1,031	6m 54.9s	248ms	10/20/2021, 14:26:03

Fig. 36 – Model List

By clicking on the model name in the list, you will be redirected to the model detail page. Please note that a toggle button is available on the right side of the list for each model. This toggle allows you to tag a model as deployable. In order to know how to deploy a model, please go to the dedicated section.

Each model page is specific to the datatype/training type you choose for the experiment training. Screens and functionality for each training type will be explained in the following sections. You can access a model page by two ways :

- by clicking on a graph entry from the general experiment page
- by clicking on a list entry from the models top navigation bar entry

Then you will land on the selected model page splitted in different parts regarding the training type.

For each kind of tabular training type, the model general information will be displayed on the top of the screen. Three sections will be available.

Models > XGB-3 Download CV

MODEL INFORMATIONS		HYPERPARAMETERS <span>Download</span>		SELECTED FEATURE ENGINEERINGS	
technology	XGB	colsample_bytree	0.8451704283003703	✓ Frequency encoding	
score	8,577	eta	0.05	✓ Linear feature scaling	
metric	rmse	eval_metric	rmse		
metric standard deviation	1,361	max_depth	7		
train duration	9m 50s	min_child_weight	15		
predict response time	292ms	objective	reg:linear		
deployable	Yes	reg_lambda	0.5236229827587133		
model used for blend	No	silent	1		
arrival time	10/20/2021, 14:42:53	subsample	0.843057816759945		
		feature_selected	["freq", "lin"]		
		seed	33479		
		num_boost_round	1079		

Fig. 37 – Model detail

- Model information : information about the trained model such as the selected metric and the model score
- Hyperparameters : downloadable list of hyperparameters applied on this model during the training
- Selected feature engineerings (for regression, classification & multi-classification) : features engineerings applied during the training
- Preprocessing (for text similarity experiments) : list of pre-processing applied on textual features

Please note that for following experiments types, the general information parts is different than from others :

- Image detection experiments : no feature engineering
- text similarity experiments : preprocessing are displayed instead of feature engineering

## Model page - graphs explanation

In order to better understand the selected model, several graphical analyses are displayed on a model page. Depending on the nature of the experiment, the displayed graphs change. Here an overview of displayed analysis depending on the experiment type.

Tableau 5 – Type of Training. Tabular Data

chart	re-gres-sion	clas-sifi-ca-tion	multi-classification	text similarity	Time se-ries	Image regres-sion	Image classifi-cation	Image multi-classification	Image detec-tion
Scatter plot graph	Yes	No	No	No	Yes	Yes	No	No	No
Residual errors distribution	Yes	No	No	No	Yes	Yes	No	No	No
Score table (textual)	Yes	No	No	No	Yes	Yes	No	No	No
Residual errors distribution	No	No	No	No	No	No	No	No	No
Score table (overall)	No	No	Yes	No	No	No	No	Yes	No
Cost matrix	No	Yes	No	No	No	No	Yes	No	No
Density chart	No	Yes	No	No	No	No	Yes	No	No
Confusion matrix	No	Yes	Yes	No	No	No	Yes	Yes	No
Score table (by class)	No	Yes	Yes	No	No	No	Yes	Yes	No
Gain chart	No	Yes	No	No	No	No	Yes	No	No
Decision chart	No	Yes	No	No	No	No	Yes	No	No
lift per bin	No	Yes	No	No	No	No	Yes	No	No
Cumulated lift	No	Yes	No	No	No	No	Yes	No	No
ROC curve	No	Yes	Yes	No	No	No	Yes	Yes	No
Accuracy VS K results	No	No	No	Yes	No	No	No	No	No

Please note that you can download every graph displayed in the interface by clicking on the top right button of each graph and selecting the format you want.

## Scatter plot graph

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

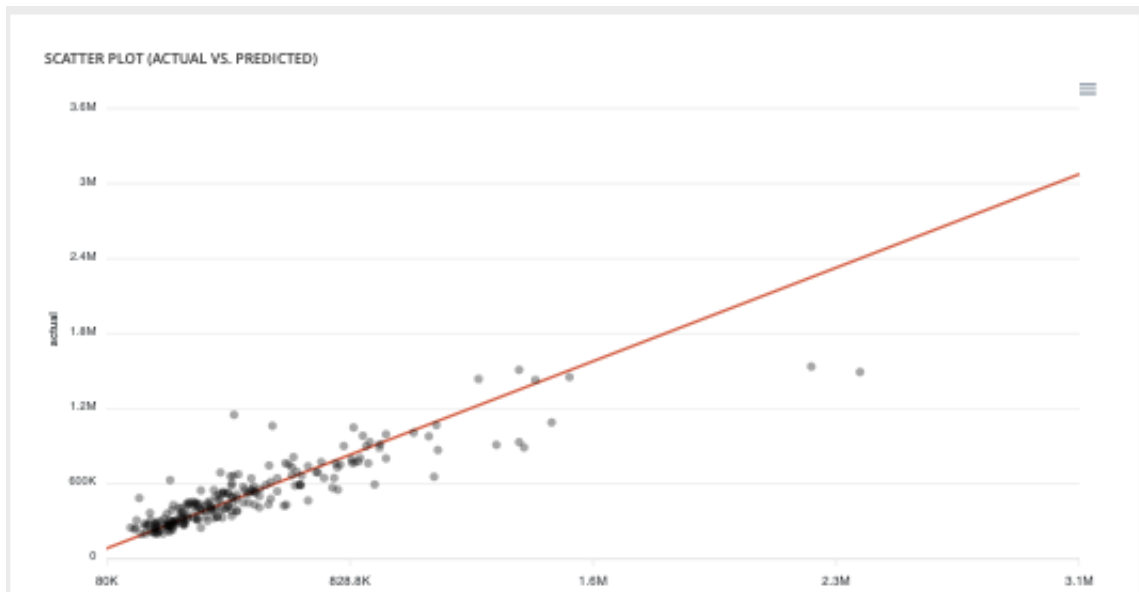
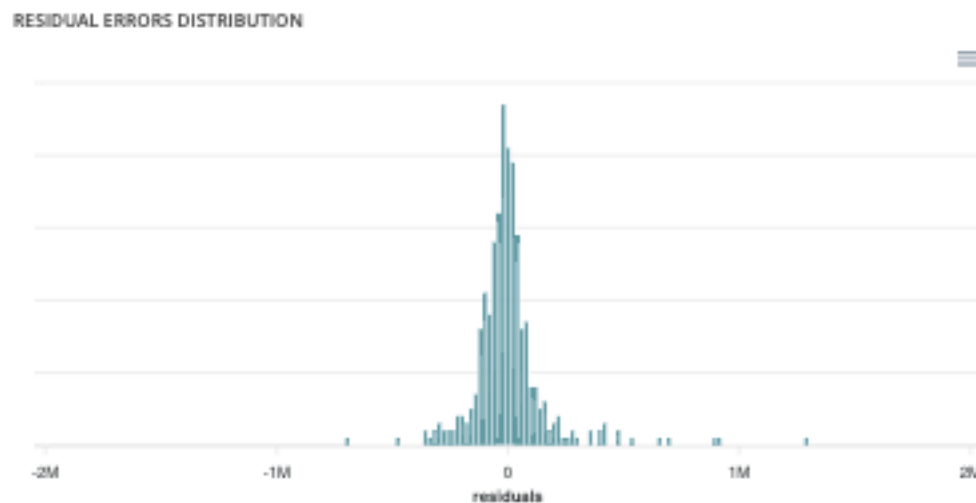


Fig. 38 – Scatter plot

## Residual errors distribution

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



## Score table (textual)

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	27,223,899,929
Root mean squared error	164,997
Mean absolute error	94,770
R2	99.95%
Mean absolute percentage error	18.55%

## 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.



## 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.

## 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.

## Confusion matrix

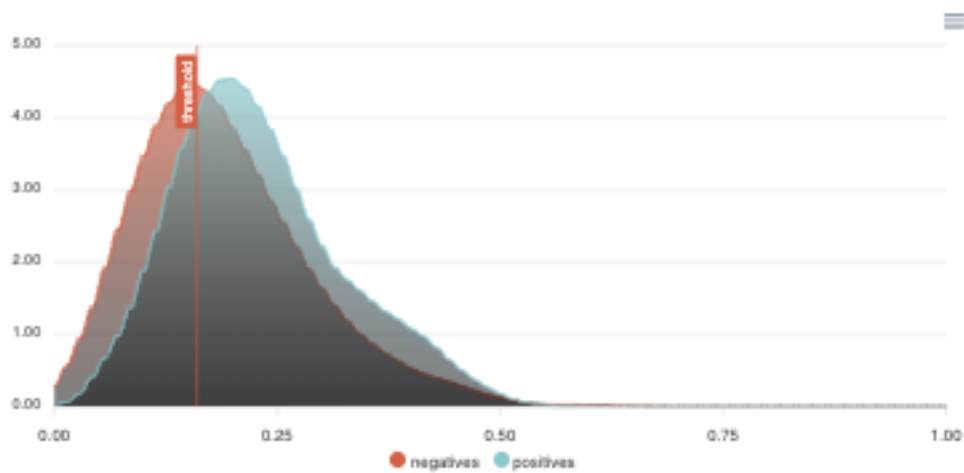
The confusion matrix helps to understand the distribution of true positives, false positives, true negatives and false negatives according to the probability threshold. The boxes in the matrix are darker for large quantities and lighter for



## COST MATRIX

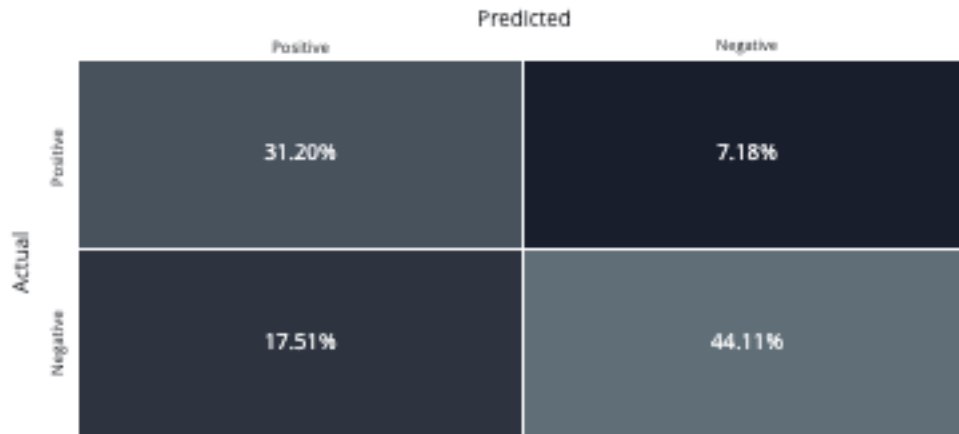
value = true/1	predict = true/1	gain = 10 <input type="text"/>	expected = 0.248
	predict = false/0	gain = -5 <input type="text"/>	expected = -0.804
value = false/1	predict = true/1	gain = -5 <input type="text"/>	expected = -0.088
	predict = false/0	gain = 5 <input type="text"/>	expected = 3.984

DENSITY CHART



small quantities.

CONFUSION MATRIX



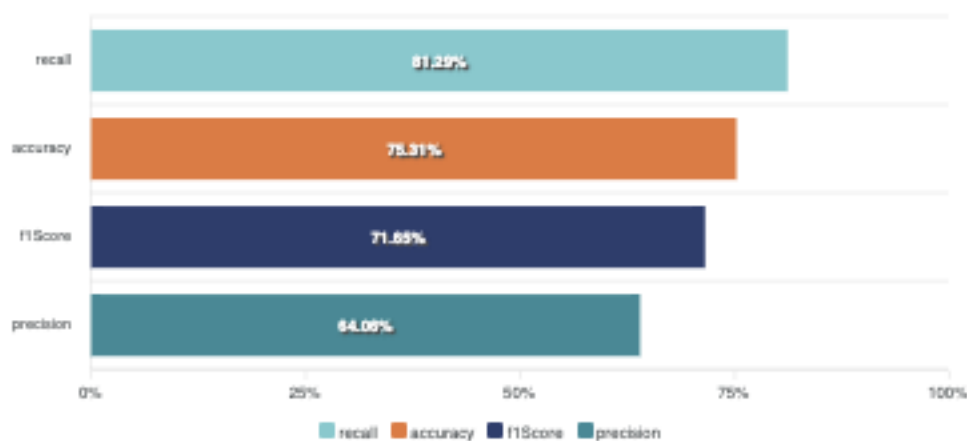
Ideally, most classified individuals should be located on the diagonal of your matrix.

### Score table (graphical)

Among the displayed metrics, we have :

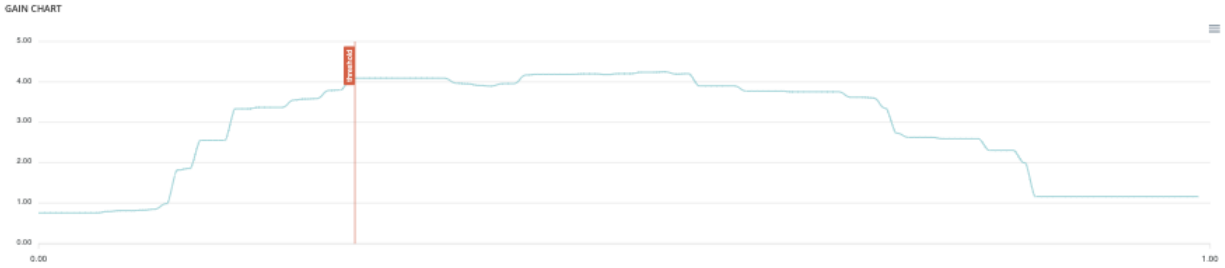
- Accuracy : The sum of true positives and true negatives divided by the number of individuals
- F1-Score : Harmonic mean of the precision and the recall
- Precision : True positives divided by the sum of positives
- Recall : True positives divided by the sum of true positives and false negatives

SCORE TABLE



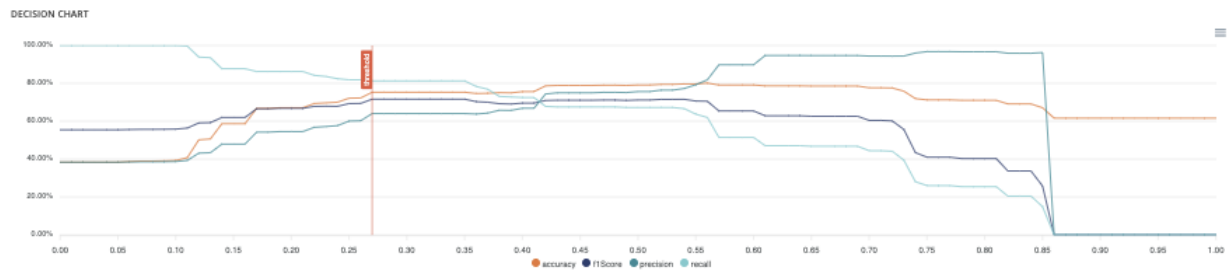
### Gain chart

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

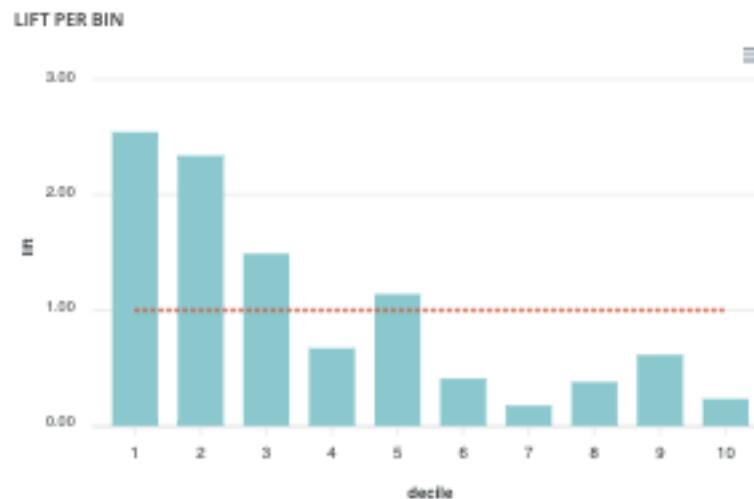
The decision graph allows you to quickly visualize all the proposed metrics, regardless of the probability threshold. Thus, one can visualize at what point the maximum of each metric is reached, making it possible for one to choose its selection threshold.



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.

## Lift per bin

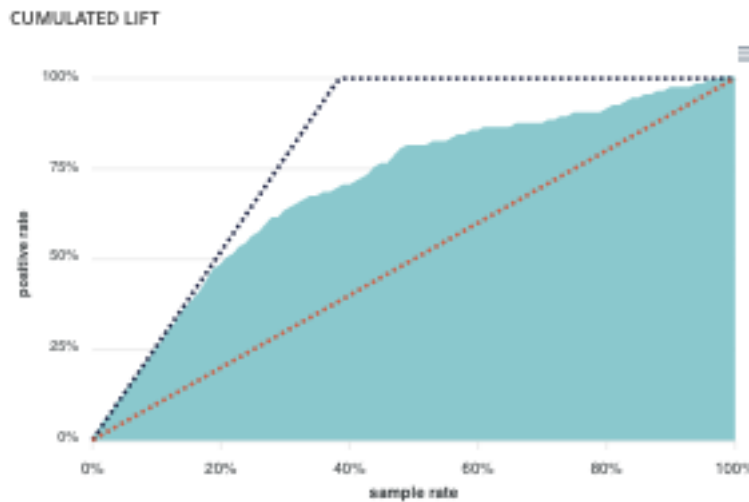
The predictions are sorted in descending order and the lift of each decile (bin) is indicated in the graph. Example : A lift of 4 means that there are 4 times more positives in the considered decile than on average in the population.



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).

## ROC curve

The ROC curve illustrates the overall performance of the classifier (more info : [https://en.wikipedia.org/wiki/Receiver\\_operating\\_characteristic](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.

## Accuracy VS K results

this graph shows the evolution of accuracy and MRR for several value of K results

## Features

In this section you will find any information relative to the dataset used during the train.

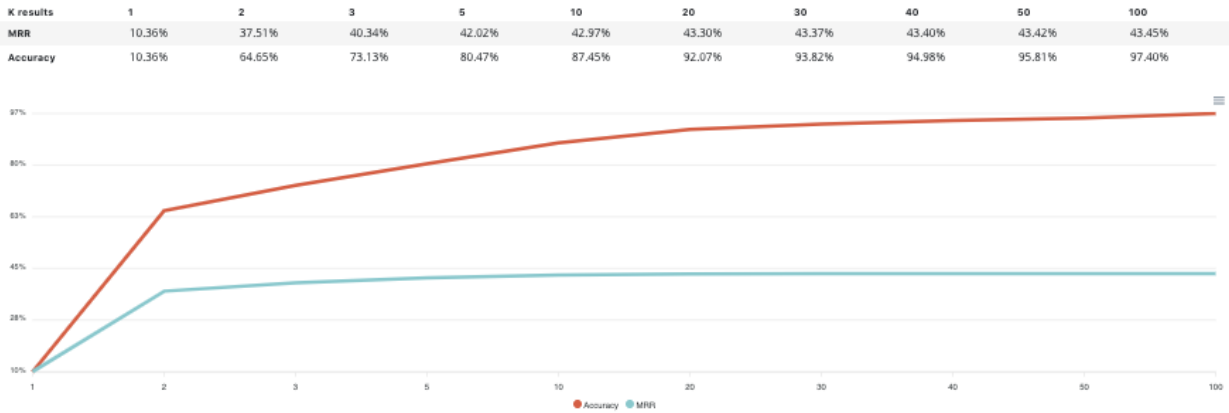
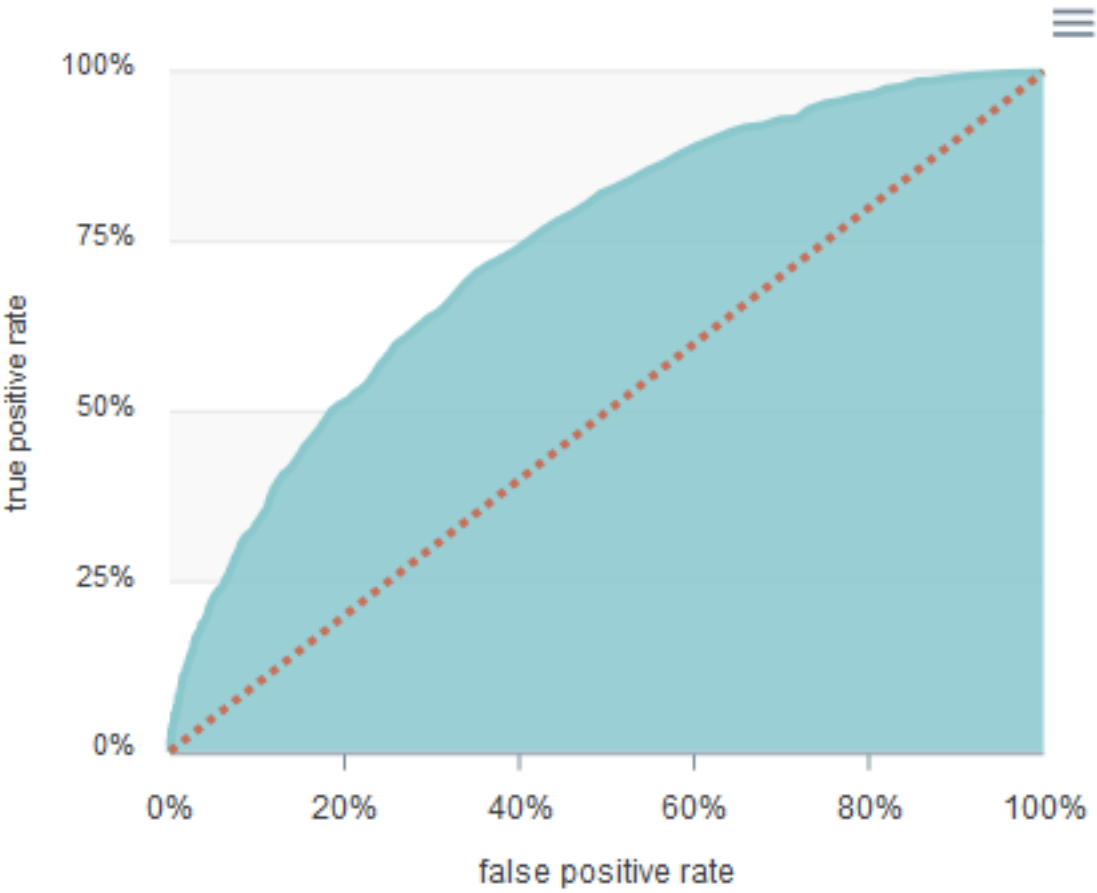
On the top panel, you will find generic information about the dataset used during the train such as the number of columns, number of samples and number of cells or the usecases using this dataset.

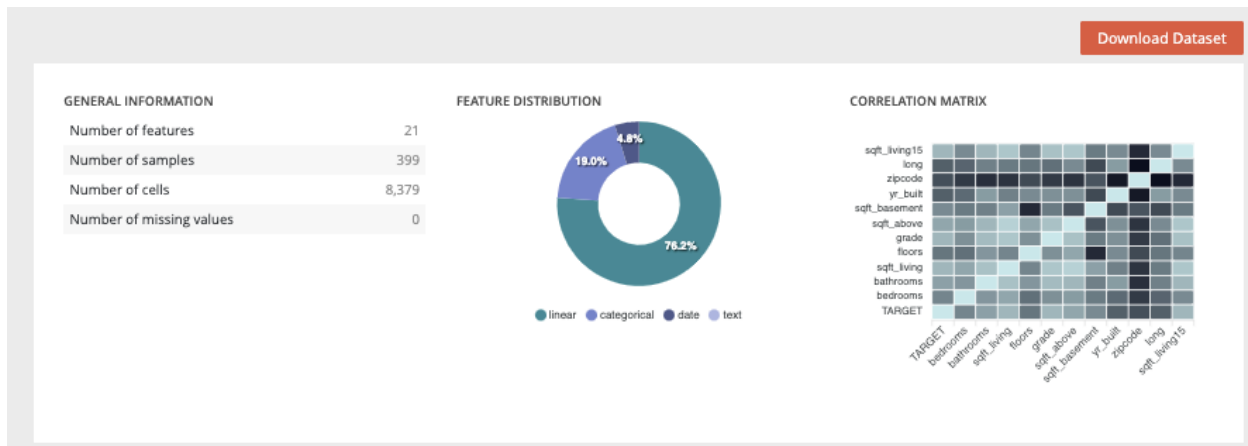
You can also download the dataset used for the training by clicking on the “download dataset” button on top of the page.

Two graph are also displayed showing :

- the feature distribution regarding the feature type (linear, categorical, date or text). This distribution is automatically calculated when uploading a dataset into the platform
- correlation matrix showing the correlation coefficients between variables. Each cell in the table shows the correlation between two variables

ROC CURVE (AUC = 0.7358)





Under this top panel, two tabs are available :

- Features analysis : table displaying features information calculated after the upload of the dataset such as the % of missing value.
- Dropped features : In this tab, you will find a list of all features you dropped for the usecase training during the usecase configuration

By clicking on a feature name you will be redirected to feature detail page

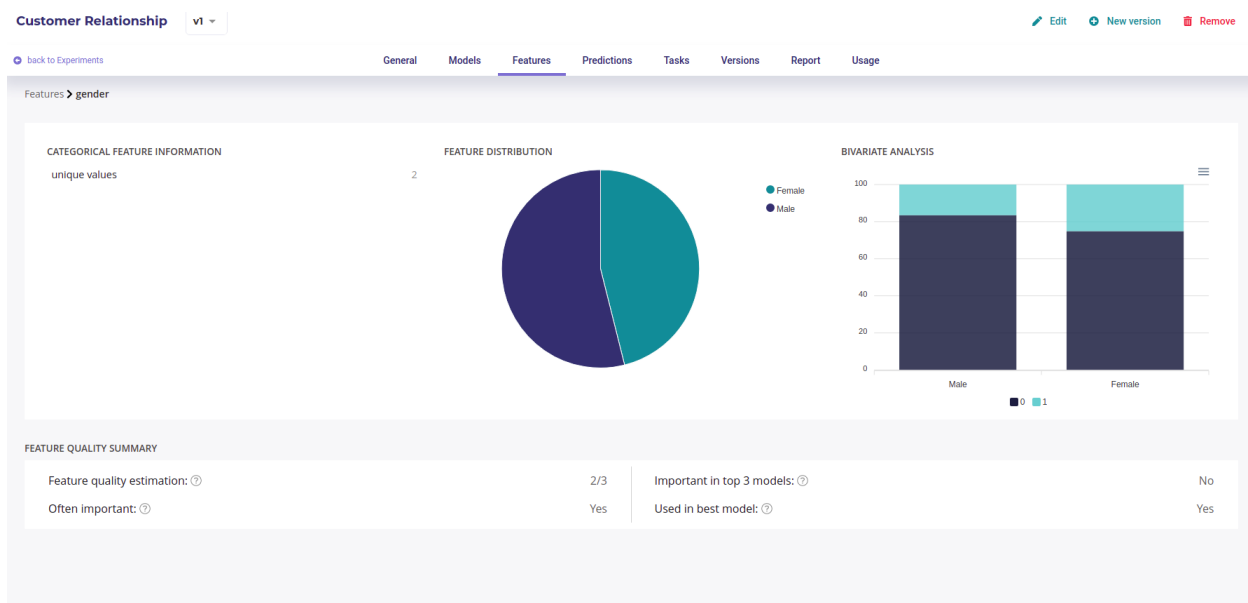


Fig. 39 – Feature detail

The feature detail has some statistics chart about the features, its relation to the target and its signal power for the submitted problem

## Predictions

The predictions menu allows you to do bulk predictions using a previously loaded [dataset](#) and see holdout predictions made during training.

In order to do a new prediction, you have to first select a model from the dedicated dropdown list and then a dataset uploaded on the project. Then, by clicking on the “launch prediction” button, the system will compute and generate a

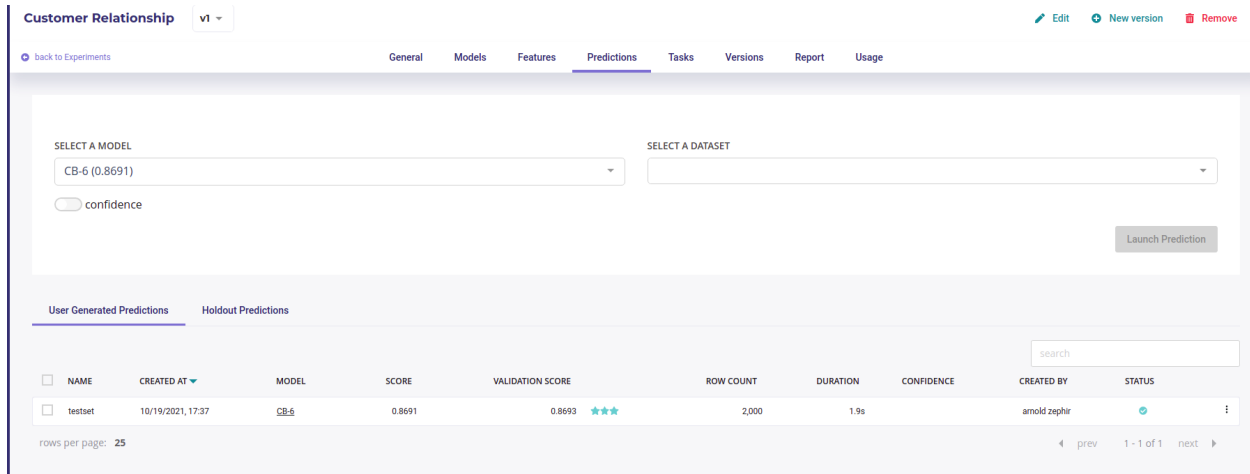


Fig. 40 – Make and see predictions

prediction file downloadable by clicking on the right side button on the prediction list below.

## Versioning your experiments

In the Prevision.IO platform you can iterate versions of your experiments.

All the version keep one common thing : the target of your training, that defines your experiment. Yet from one version to the others you can change any parameters to see how it impacts performance and stability of your model.

To do that, three possibilities :

- On the experiment list of a project, by clicking on the right side action button of the experiment you want to iterate and select “new version”
- On any pages of a experiment by clicking on the top right “actions” button and select new version
- On the “Version” menu of a experiment, by clicking on the action button right side of a version listed and select “new version”

Then, on the version menu of a experiment, you will find the list of all trained versions for this experiment. By clicking on the version number, left side of this list, you will access the selected experiment version page. You can also navigate through versions by using the dropdown list top left of the information banner on any page of a experiment.

After clicking on a new version button, you will be redirected to the experiment configuration menu. The experiment version configuration you selected for your iteration will be automatically loaded in order for you to know what configuration was done and select the changement you want to apply.

---

**Note :** when creating a new experiment or a new version, add a description to your experiment at the first screen of new experiment configuration. It will help you finding the version you want to work with later.

---

The default dashboard is those of the **last version** of your experiment. If you have many *version of your experiment*, you can change it with the dropdown menu on the top left corner.

The front page of experiment dashboard shows you :

- General : general information and comparison of your models in terms of performances
- Models : list view of the created models and information about the *trained models*
- Features : information about the way *the features* are used for the training and the configuration of the feature engineering
- Prediction : create *bulk predict* using CSV files and view all bulk predictions done for this usecase

Basics\*

Columns configuration

Models

Feature engineering

Feature selection

Version description

Describe your Experiment

Data

Dataset ?

trainset

Holdout (optional)

testset

Training options

Metric to use

AUC - area under the receiver operating characteristic curve

Performances

☐ QUICK ?
☒ NORMAL
☐ ADVANCED ?

Fields configuration

Target column

target

ID column (optional)

customerid

Fig. 41 – The target stays the same and cannot be changed along versions of an experiment

TESTS 23/06

SL

S3\_titanic v2

Actions

General

Models

Features

Predictions

Tasks

Versions

Report

VERSION	DESCRIPTION	CREATED AT	CREATED BY	SCORE	MODELS	PREDICTIONS	STATUS
V2		06/28/2021, 16:06	Simon Levacher	-	0	0	
V1		06/25/2021, 14:42	Axel Chauvin QA test	0.9587 (auc) ★★	8	0	

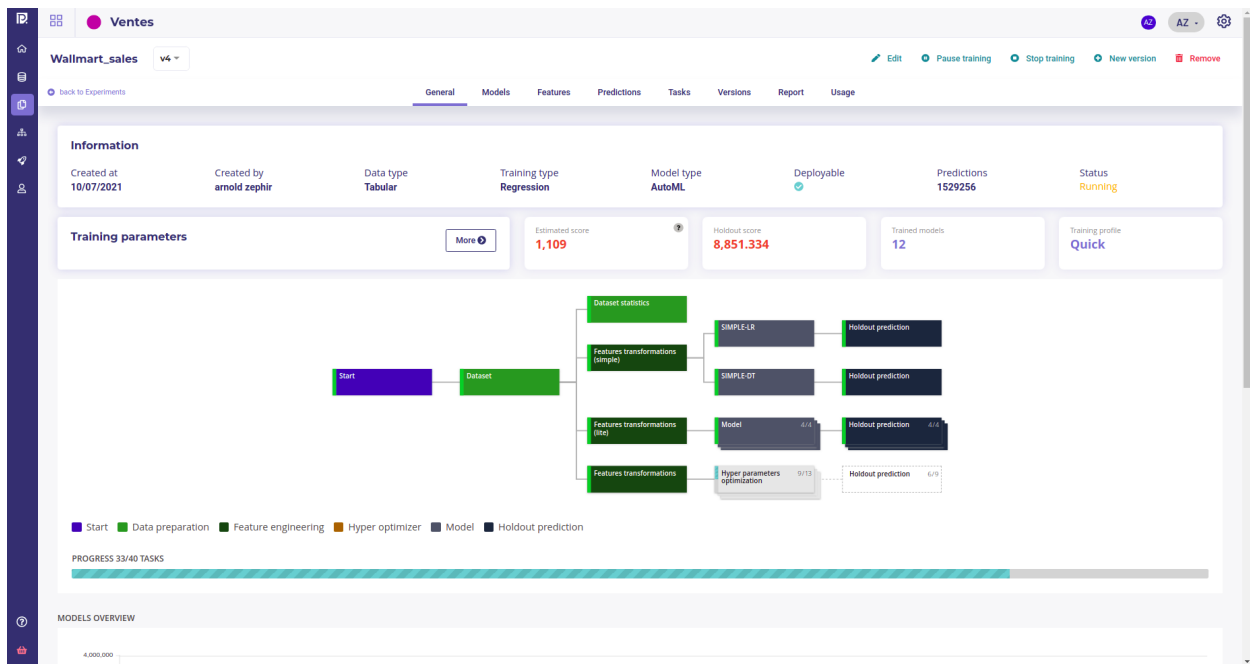
rows per page: 25

prev

1 - 2 of 2

next





- Task : Graph and listing of all operations done during training
- Versions : list of *all version* of the selected usecase
- Report : generate PDF reports explaining the models/usecases

The information header gives you the important information regarding your usecase. You can navigate through the versions using the dropdown list on the left side of this panel.

- Action button : on the top-right corner of the page are the actions buttons allowing you to :
  - edit the name and description of the experiment version
  - create a new *version*
  - delete the experiment
- Under the information panel, cards displaying information regarding your usecase are displayed. Please note that the holdout score card will be displayed only if a holdout was selected during training configuration
- Two graphs are displayed on the general page of a usecase showing : - The models ranked by score. By clicking on a model chart bar, you can access to the selected model details - Models score vs. estimated prediction time

Please note that for object detection, the general screen is quite different from the other use cases types. On the image detection general menu you will find a sample of images used during the train in orange, the predicted bounding boxes using cross validation and in blue, the true bounding boxes.

## Tasks

In this menu you will find an overview of all tasks made by the platform during the usecase training and their status. The aim of this screen is to help you to better understand the operations made during the training and, if errors occurred, at which level it happened. When a task failed, you can access logs by clicking on the **logs button** that appears.

Two views are available :

- Liste view : list all single operations done
- DAG view : graphical view of single operations and their relation

You can switch between these views by clicking on the execution graph / tasks board switch.

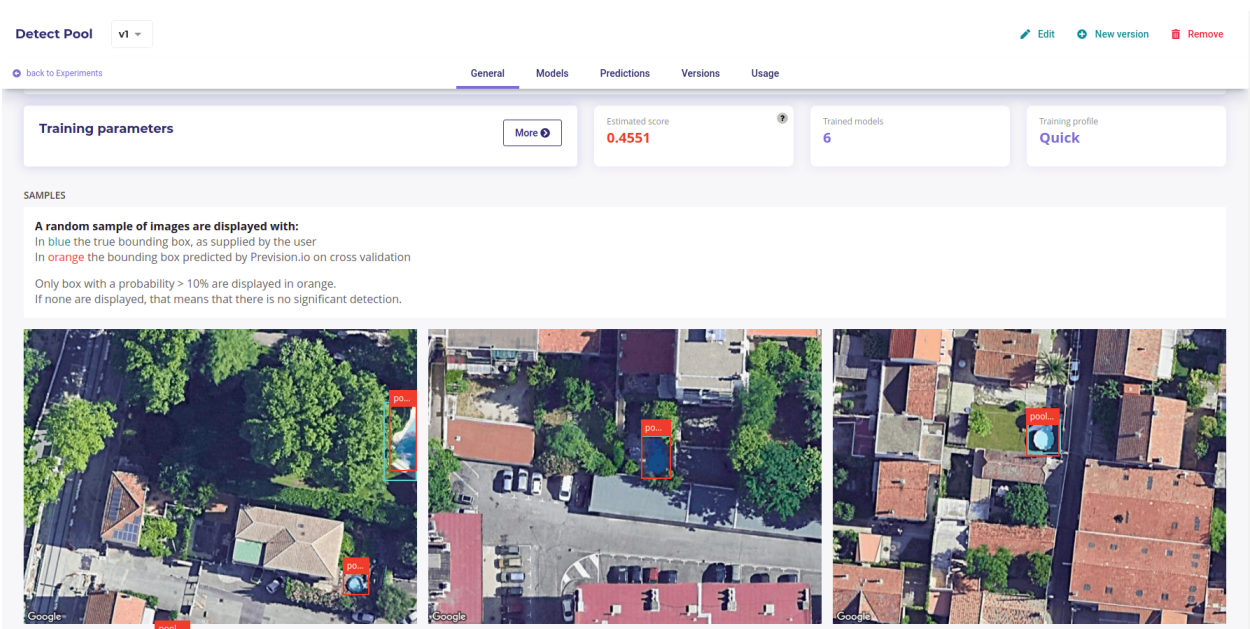


Fig. 42 – Object detector dashboard

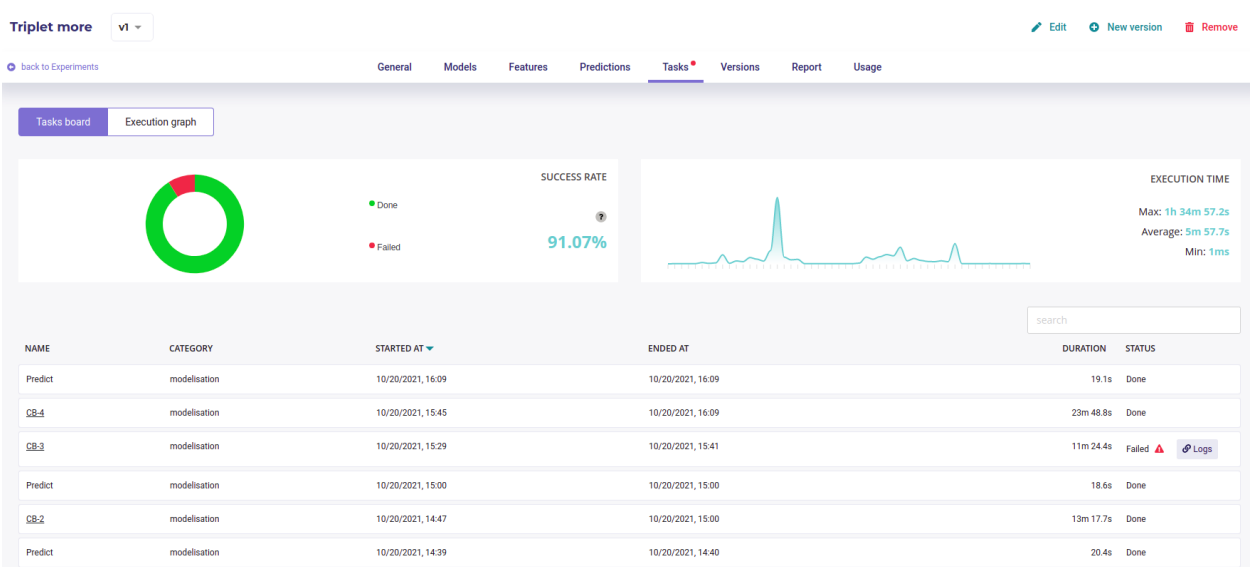


Fig. 43 – All the tasks executed

## Reports

In this menu, you can generate PDF reports regarding models from the usecase. To do that, once on the dedicated model menu, you will have to choose from the drop down the models you want to appear in the generated report and the feature importance count. You also can select explanations by check/uncheck the show explanation checkbox. Then, by clicking on the generate button, you will get an overview of the report. By clicking on the “print” button on the top of the overview, you will download the PDF report.

### 5.3.1.4 Pipelines

In Prevision.IO platform you can automate several actions using the pipeline editor. We will see in this chapter the possibilities of the pipelines and its components and how the editor works. In order to execute a pipeline, several requirements need to be fulfilled :

- first, you have to create your own template using the pipeline editor. This template includes generic components with no configuration required in this step. This allows you to create a generic template and apply it several times on different experiments by configuring the component.
- then, you will be able to configure the pipeline run by choosing an already created pipeline template and by configuring the nodes to your experiment. You also can choose to run the pipeline manually or automatically by using the scheduler.

Optionally, you can create and load into the platform your own components and use them into pipelines.

Before going into detail about the creation of pipelines itself, let’s have a look at the pipeline components existing in the platform and their purpose. This overview will help you to better understand the possibilities of Prevision.IO pipelines

### Pipeline components

Pipeline components can be considered as steps, or nodes, of the pipeline. Several categories of components are available in order to easily find them when building a pipeline template.

Some Components are provided on the shelf when you subscribe a Prevision.io plan.

The screenshot shows the 'Pipelines Prod' interface with a table of pipeline components. The table has columns for NAME, NODE TYPE, CREATED BY, CREATED AT, and DESCRIPTION. The components listed include 'Import dataset', 'Import dataset from datasource', 'Export dataset', 'deployment\_predict\_on\_regression', 'deployment\_predict\_multiclassification', 'deployment\_predict\_classification', 'Add country kpi', 'Concatenate 2 columns', 'Sentiment Analysis', 'Add siren siret', 'Add POI data', 'Add weather', 'Sample', 'Add special days', 'Pad column', 'Filter outliers', 'Sentiment Product Review', 'Add weather city day', 'Sentiment Recognition', and 'Add siren kpi'.

NAME	NODE TYPE	CREATED BY	CREATED AT	DESCRIPTION
Import dataset	Import	Prevision	08/09/2021, 13:46	This component will import a dataset, previously load into the dataset s...
Import dataset from datasource	Import	Prevision	08/09/2021, 13:46	This component allows you to generate a dataset from an external sour...
Export dataset	Export	Prevision	08/12/2021, 15:02	This component allows you to create and save into the platform an Out...
deployment_predict_on_regression	Predict	Prevision	08/09/2021, 13:46	This component allows you to automatize the generation of prediction ...
deployment_predict_multiclassification	Predict	Prevision	08/09/2021, 13:46	This component allows you to automatize the generation of prediction ...
deployment_predict_classification	Predict	Prevision	08/09/2021, 13:46	This component allows you to automatize the generation of prediction ...
Add country kpi	Prevision components	Prevision	08/09/2021, 13:46	This component will add different KPI for countries
Concatenate 2 columns	Prevision components	Prevision	08/09/2021, 13:46	This component will concatenate 2 columns using a selectable character
Sentiment Analysis	Prevision components	Prevision	08/09/2021, 13:46	This component will compute text sentiment analysis probability
Add siren siret	Prevision components	Prevision	08/09/2021, 13:46	This component will add siren and siret static features for each siret
Add POI data	Prevision components	Prevision	08/09/2021, 13:46	This component will add the number of electric charging stations within...
Add weather	Prevision components	Prevision	08/09/2021, 13:46	This component will add weather features depending on temporality an...
Sample	Prevision components	Prevision	08/09/2021, 13:46	This component will sample the input dataset according to the selectabl...
Add special days	Prevision components	Prevision	08/09/2021, 13:46	This component will add special days in country for each date
Pad column	Prevision components	Prevision	08/09/2021, 13:46	This component will pad column with "char" to "length".
Filter outliers	Prevision components	Prevision	08/09/2021, 13:46	This component will filter rows where values in numerical columns fall o...
Sentiment Product Review	Prevision components	Prevision	08/09/2021, 13:46	This component will compute text sentiment product review probability (...)
Add weather city day	Prevision components	Prevision	08/09/2021, 13:46	This component will add weather features in cities for each date
Sentiment Recognition	Prevision components	Prevision	08/09/2021, 13:46	This component will compute text sentiment analysis class
Add siren kpi	Prevision components	Prevision	08/09/2021, 13:46	This component will add yearly accounts for companies, based on date ...

- Import : All component relative to the import of data
- Export : All component relative to the export of data
- Prevision component : Various component developed by Prevision.IO datascientists for their various projects
- Custom component : Components developed by you or your team
- Predict : All components relative to the automatisisation of the predictions
- Retrain : All components relative to the automatisisation of model training

Each component has a description helping you to choose the ones suitable for your needs. You can access all components by clicking on the pipelines menu on the side project's menu and, navigate to the pipeline components menu.

The screenshot shows the 'Projet de Simon - test edit' interface with a table of pipeline components. The table has columns for NAME, NODE TYPE, CREATED BY, CREATED AT, and DESCRIPTION. The components listed include 'Import dataset', 'Import dataset from datasource', and 'Export dataset'.

NAME	NODE TYPE	CREATED BY	CREATED AT	DESCRIPTION
Import dataset	Import	Prevision	08/09/2021, 13:46	This component will import a dataset, previously load into the dataset s...
Import dataset from datasource	Import	Prevision	08/09/2021, 13:46	This component allows you to generate a dataset from an external sour...
Export dataset	Export	Prevision	08/09/2021, 13:46	This component allows you to create and save into the platform an Out...

## Building you own component

You can build and use your own component for custom dataset transform.

A boilerplate with more guide is available on the [Prevision.io public repo](#)

To use your own component you need a gitlab or a github account ( and it needs to be setup in your account page )

Once done, your component may be use in any Pipeline Template.

## Pipeline templates

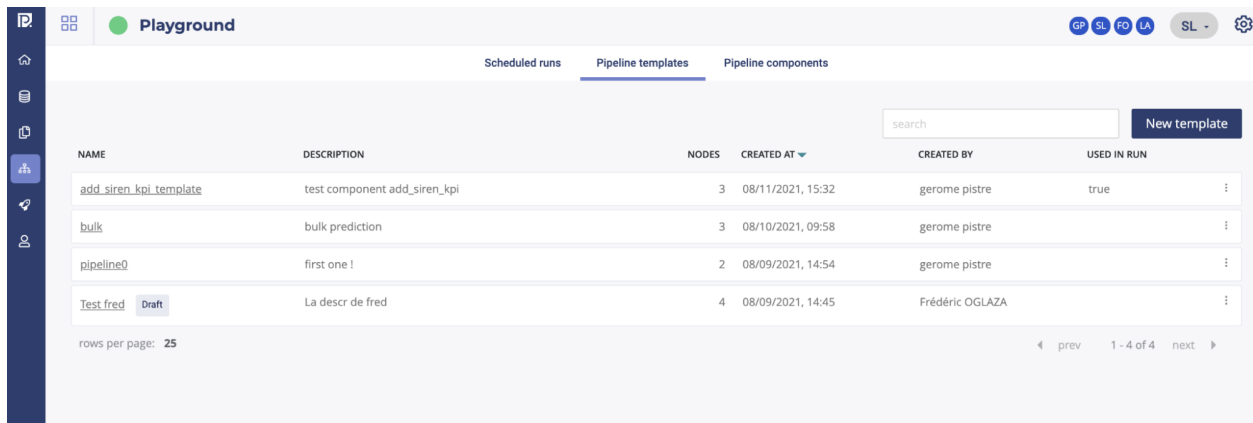
Pipeline Template are a tool for describing operations to schedule. In a pipeline template, you do not input any parameters. Instead you build a pipeline by linking nodes together and setting some placeholder.

Your pipeline template will then be used in the schedule step, where you, or someone else, is going to fill the placeholder with their inputs.

For example, you can define a template that « add an integer to the age column ». The value of this integer will fill in when a scheduler use your template.

A template may be used in different scheduled run with different parameters each time.

In order to create a new pipeline template, you have to navigate through the pipeline template menu.



You will then access the pipeline template liste of the project. By clicking on the “new template” button, you will access to the pipeline template editor.

The first step is to set up the name of your pipeline template and, optionally, a description.

**New template**

1 Settings > 2 Template

Name

0 / 40 characters

Description (optional)

0 / 210 characters

Once ( and only once ) the required information is fulfilled, you will be able to reach the next step by clicking on the next button bottom right.

Then, the pipeline graphical editor will be visible and you will be able to start the creation of your pipeline template. In order to add your first node, you have to click on the “+” button in the center of the graphical area.

Cancel and back

New template

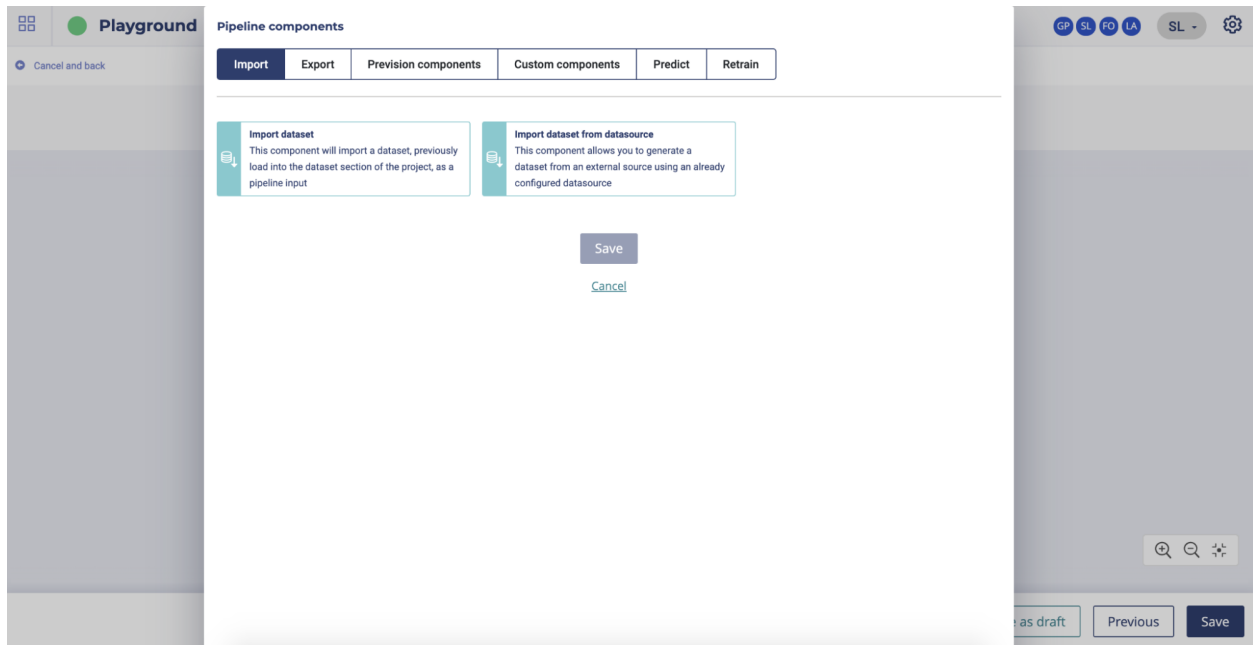
✓ Settings > 2 Template

🔍 🔍 ⚙️

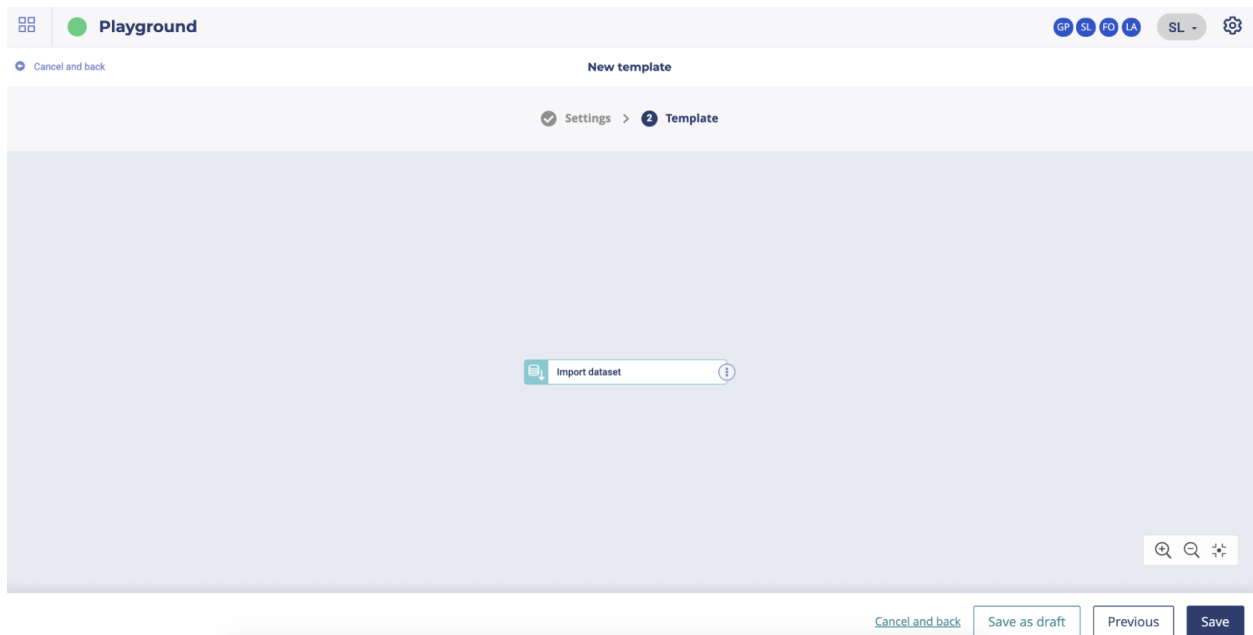
Cancel and back Save as draft Previous Save

Then a popup window will appear allowing you to select the first node. In this popup, the components are classified in several categories. You can navigate through nodes categories by clicking on the top bar menu. In order to add a node, you simply have to click on it and save. Please note that the eligible nodes are colored unless ineligible ones that are in gray. This will help you through the creation of your pipeline in order to be sûre that the final result is conform to

what the platform is expected.



The selected node will be visible in the graphical view. Please note that you can save as draft your pipeline template in order to finish it later by clicking on the bottom right “save as draft” button.



Several actions are possible on the newly added node. You can access to the possible actions by clicking on the more action button on the right side of the node.



Four actions are possible :

- add a node after the selected one
- switch this node for another
- view settings of the node
- delete the node

Please note that some special nodes can have limited actions.

Once your pipeline template is finished, you can save it by clicking on the bottom right “save” button. You will be then redirected to the pipeline template list.

## Scheduled runs

A scheduled run is the combination of a pipeline template and a specific schedule. It allows to trigger some pipeline at regular interval.

Once you had defined a pipeline template, you can use schedule it for running :

- Once
- periodically forever
- periodically for a defined period of time

## When to used scheduled runs ?

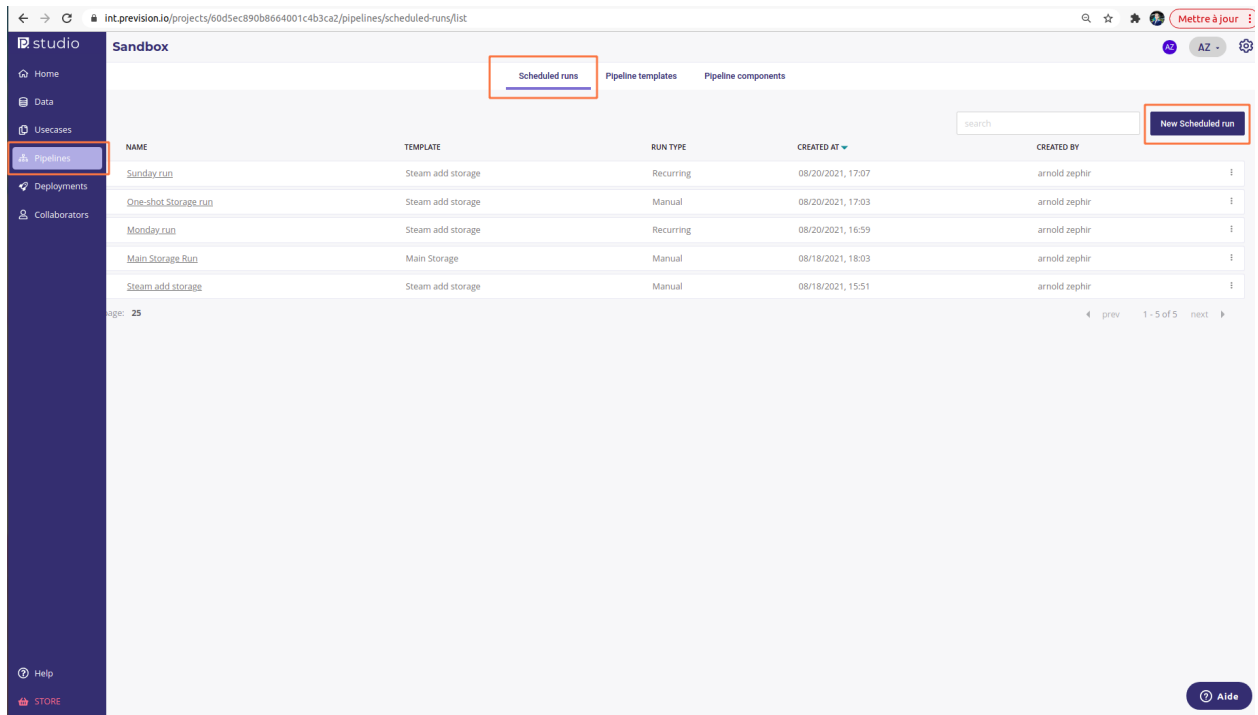
In most of case, scheduled runs are used :

- for delivering prediction on a periodic schedule. For example sending a list of churner to sales team each monday
- for retraining a model, for example each first day of a quarter
- for computing engineered features and pushing them to others teams on a regular basis

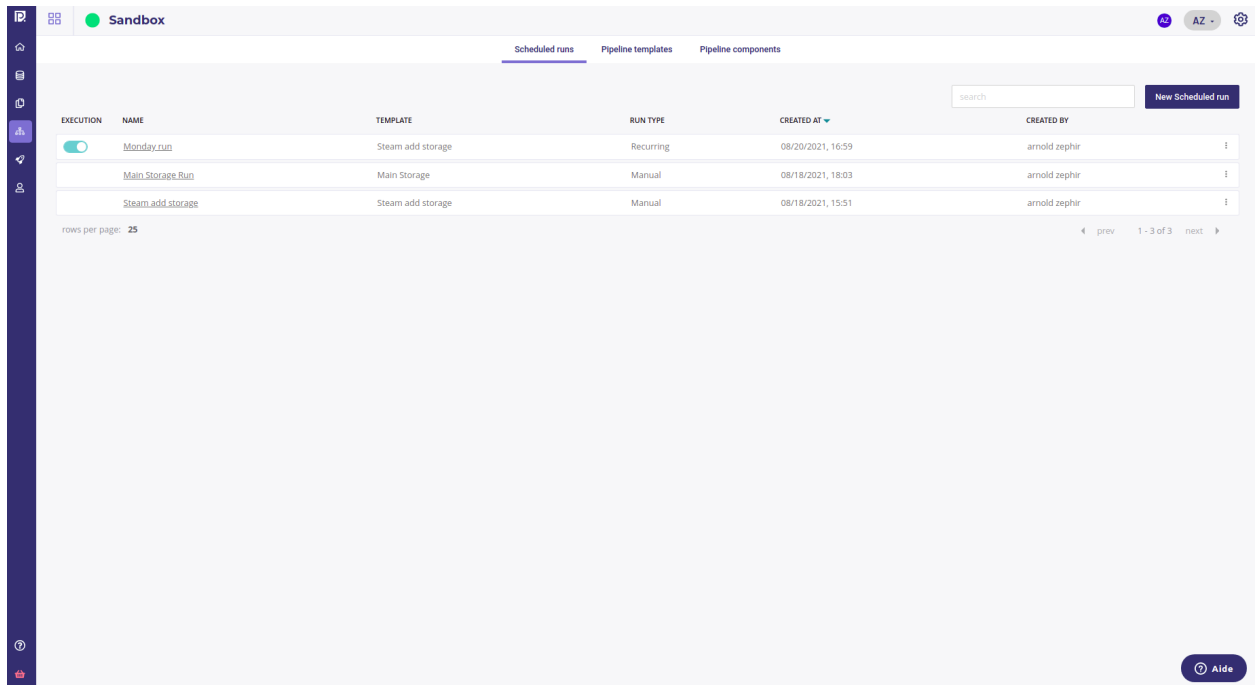
## Create a new scheduled running

Scheduled run are available in the Pipeline section of a project, available under the « Scheduled runs » tab.





From the main page, you can view a list of all your scheduled runs and create a new one by clicking on « New Scheduled run » button



First step is to give a name and some description and select a pipeline template.

Select the template you want to fill in and click on « next » on the right lower corner.

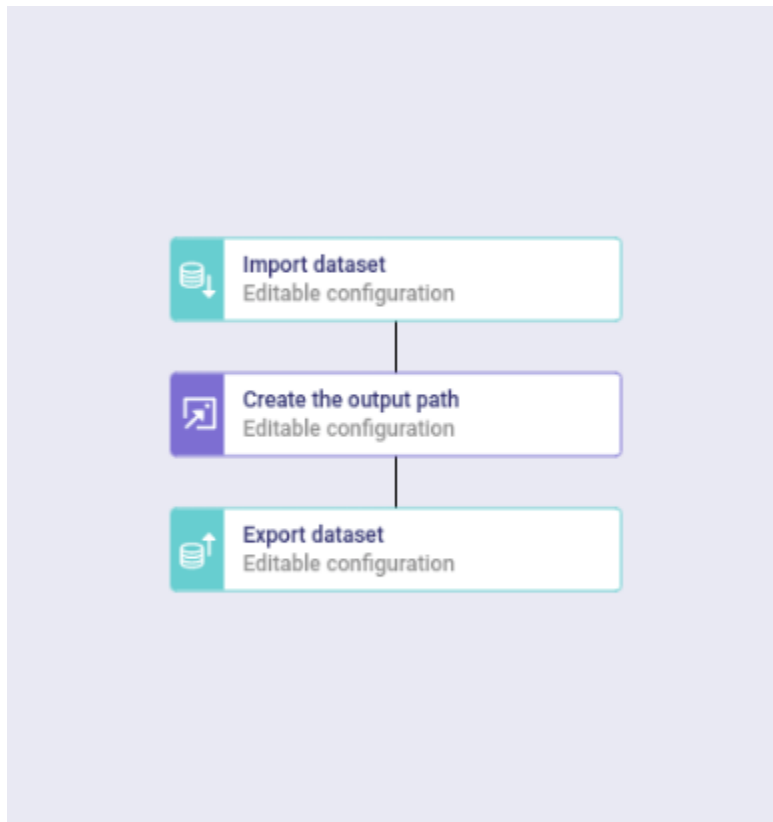
**Note :** Pipeline templates were created in a previous step, with some input inside nodes to fill in. « Scheduled runs »

is the place were you are to fill them up.

The next screen is were you fill all the parameter of your pipeline. For each node with one or more parameter to provide, a *to configure* yellow label is displayed on the node.

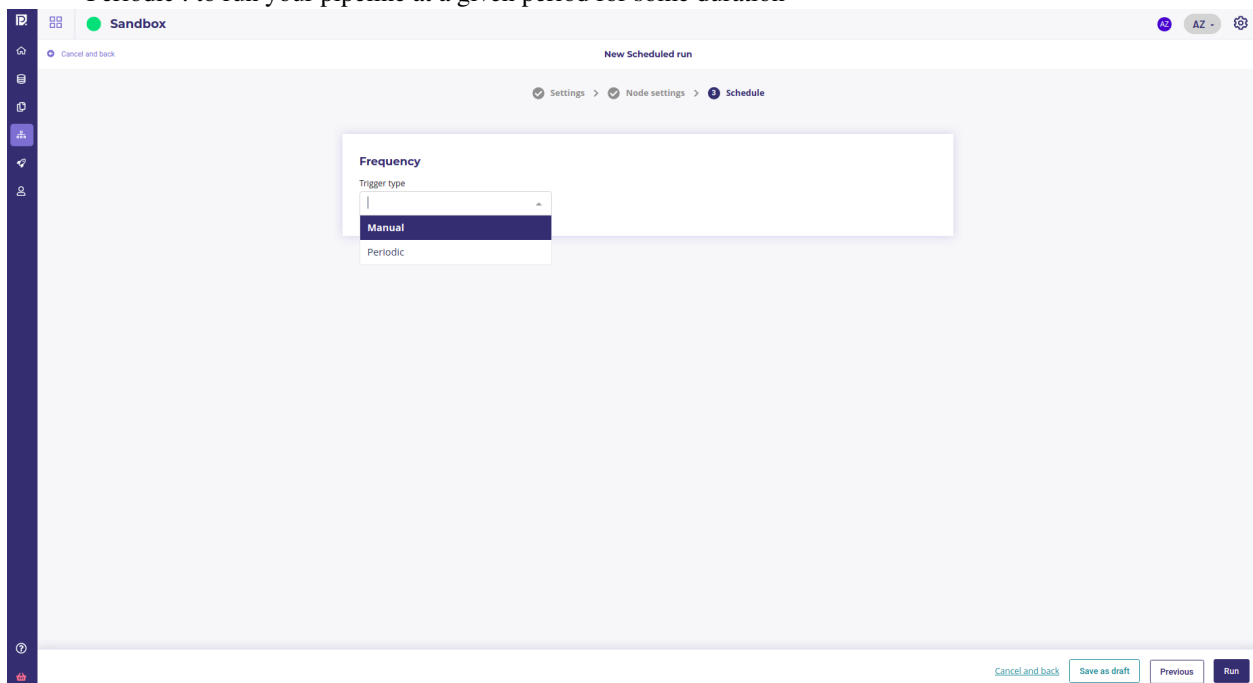
To enter a parameter, click on each node and fill the input. **Click on save to save them** ( parameters are not taking into account till you click on save ). To know more about a parameter, you may go to the « pipeline components » tab and click on the component to get input description.

Once you filled all the node parameters, there should be no more yellow label :



You can click on « next » and choose the Trigger type :

- Manual : to run your pipeline once (useful for testing ). The pipeline will be run as soon as you click on run ( it can be run as much as you want later )
- Periodic : to run your pipeline at a given period for some duration



If you choose « periodic » you will be prompt to input some information :

- hour ( and minutes ) for hourly run. The pipeline will be launch every day, every x hours. for Example, if you input 2 hours, the pipeline will be ran twelve time a day ( every 2 hours )
- day period, hour and minutes of execution for daily run.

**Note :** Note that « Day » is the number of day between each run. If you input « 5 »,the pipeline will run every 5 days.

- day of the week ( at least one ), hour and minut for a weekly period
- months period, day, hour and minute for monthly period
- a crontab expression ( see [crontab guide](#) for syntax ) if you select advanced mode

The screenshot shows the 'New Scheduled run' configuration page in the Prevision.io interface. The 'Frequency' section is active, with 'Trigger type' set to 'Periodic'. The 'Period' dropdown menu is open, showing options: Hourly, Daily, Weekly (selected), Monthly, and Advanced. To the right of the dropdown, there are input fields for 'Hour' (set to 12am) and 'Minute' (set to 0). Below the dropdown is a 'Duration (optional)' field. At the bottom right, there are buttons: 'Cancel and back', 'Save as draft', 'Previous', and 'Run'.

Once period input, you may select a start and end of run. The pipeline will only run on between the date you selected.

**Frequency**

Trigger type  
Periodic

Period

Hour: 12am Minute: 0

Cancel and back Save as draft Previous Run

## See all your scheduled run

All your scheduled run are available under the Scheduled run tab ( in the pipeline section ) :

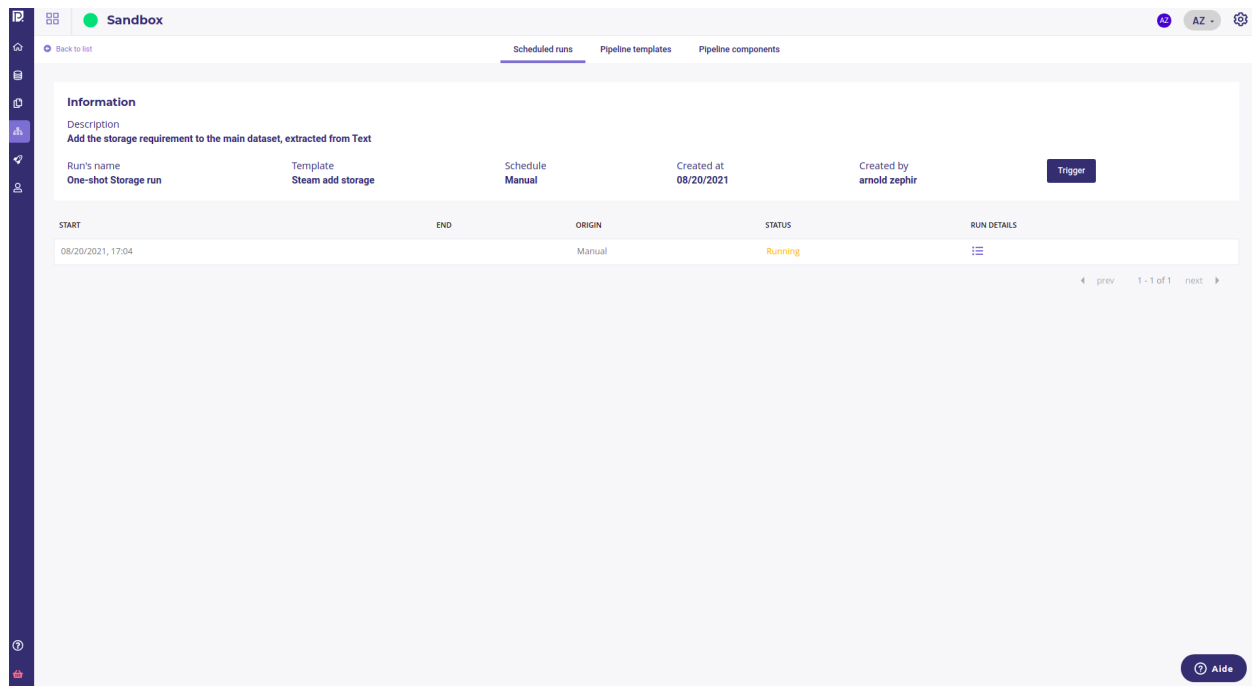
EXECUTION	NAME	TEMPLATE	RUN TYPE	CREATED AT	CREATED BY
<input checked="" type="checkbox"/>	dailyprod	retrain pop charts	Recurring	08/17/2021, 14:48	arnold zephir
<input checked="" type="checkbox"/>	weeklytrain	Retrain billboard	Recurring	08/17/2021, 14:46	arnold zephir
<input type="checkbox"/>	weeklyimmo	Immo_demo	Recurring	08/11/2021, 09:19	arnold zephir

rows per page: 25

The scheduled run with a periodic schedule have a switch in order to disable or enable them. A disabled Run will obviously not run.

The « Manual » Scheduled run does not have Execution switch. Instead you need to trigger them manually.

## See run state, trigger a run



When clicking on a Scheduled Run in the list, you get details about its executions :

- either previous executions, with a « done » status or a « failed » status
- current execution status (« running »)

If an execution failed, you can get more informations by clicking on the « run detail » icon. For all execution that succeeded, the dataset produced are available in your « data » Section with the tag « pipeline output »

Each Scheduler Run can be trigger to run once and immediately by using the « Trigger » button.

## Edit a run

You cannot edit a scheduled run, you have to create a new one.

## Delete a Scheduled run

In the list of Scheduled Run , you can use the « More actions » icons to remove a scheduled run. Note that instead of removing it, you can disabled it by using the « execution » switch.

### 5.3.1.5 Deployments

Deployments are the last step of a :doc:`Machine Learning Project </studio/projects>` and are used to

- share model across your organisation
- start monitoring you models

Deployments need *experiments with at least one model*. You can deploy any model of any version of your experiment.

#### Once an experiment is deployed...

- you can *schedule batch prediction*
- you can monitor performance of one main model and one challenger

- external users can use your model for unit prediction from an url
- external application can call your model from a REST API

Deployments are scoped to a project and available from the **deployments section** on the collapsing sidebar. When entering the deployment section, you will see a list of each of your deployment and two status :

- deployed : are models built and available over API
- running : does API reponses to request

The url column links to a page where human can call the model over a simple form in order to test it.

## Create a new Deployments

Creating a new deployment is done by clicking on the **deploy a new experiment** button under the **deployments experiments** tab.

Fig. 44 – Create a new deployment

In addition to give it a name and a description, you must :

- select one of your experiment
- select a version
- select a model from this version of your experiment

and you could select another version and another model of the same experiment taht will be deployed as Challenger model. This one will be called each time you main model is called and it's response will be recorded next to those of the main model in order to compare them and maybe be switch them.

When deploying a new experiment you need to grant access :

- public : everybody can call your model
- Instance collaborators : everybody on the instance can call your model ( note : every user on cloud.prevision.io share the same instance )
- Project collaborators : only your project's collaborators can call your model

Once you click on the deploy buttont, the model you chose will be deployed. You can check its status in the list of deployments or in the deployment page.

## Inspect and monitor a deployed experiment

The deployment page is available as soon as the experiment is deployed. On each deployment page lie five sections.

### General

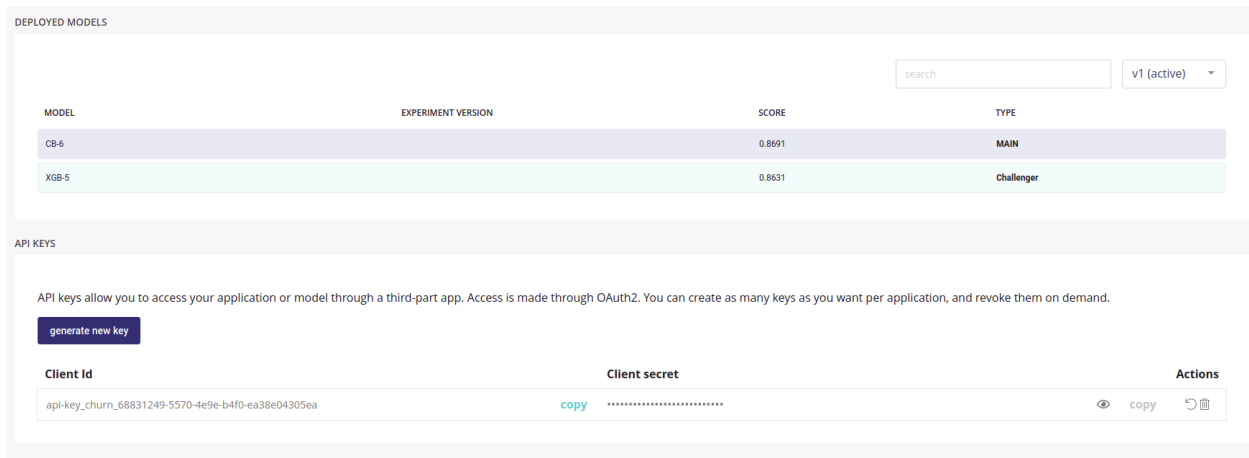


Fig. 45 – General info about your deployment

The general tab will show you :

- Creation date of your deployment
- Current version of your deployment
- A link for using the model with a simple form
- deployment status
- main model status
- challenger model status
- a link to the documentation of model API
- the access you granted

And

- a drift chart : a chart showing distribution of the training data vs distribution of the data predicted since deployed
- a summary of the models deployed
- an API key generator than you can share to others applications in order to call your model

### Monitoring

The monitoring section will show you a selection of chart of :

- the data sent to your models
- the prediction done by your models, both main and challenger if you deployed a challenger model
- the drift of your input data

Monitoring this chart let you decide when your model becomes obsolete and you should schedule a retrain.

### Predictions

The Predictions tab is where you will find all the predictions scheduled in a *pipeline* done with your deployed experiment. When an experiment is deployed, it can be used in a pipeline and this, can be schedule, to be deliver in an





Fig. 46 – Model monitoring

Best Forecast Sales model v1

Back to list

General Monitoring Predictions Versions Usage

VERSION	CREATED AT	ROWS	TYPE	MODEL	SCORE	DURATION	STATUS	DOWNLOAD
V1	10/11/2021, 15:41	127438	Main	LGB-3	9,725	8s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	5.8s	Done	<a href="#">Download</a>
V1	10/11/2021, 16:01	127438	Main	LGB-3	9,725	7.4s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	5.7s	Done	<a href="#">Download</a>
V1	10/11/2021, 17:02	127438	Main	LGB-3	9,725	7.6s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	6s	Done	<a href="#">Download</a>
V1	10/11/2021, 18:03	127438	Main	LGB-3	9,725	7.3s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	5.3s	Done	<a href="#">Download</a>
V1	10/11/2021, 19:01	127438	Main	LGB-3	9,725	7.7s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	6.2s	Done	<a href="#">Download</a>
V1	10/11/2021, 20:05	127438	Main	LGB-3	9,725	7.4s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	6.5s	Done	<a href="#">Download</a>
V1	10/11/2021, 21:05	127438	Main	LGB-3	9,725	7.7s	Done	<a href="#">Download</a>
			Challenger	XGB-4	8,970	6.6s	Done	<a href="#">Download</a>

Fig. 47 – Predictions list

external database each monday for example. Each time a prediction is ran in a pipeline, a file is generated both for the main model and the challenger model and they can be downloaded for further inspection from this section.

## Versions

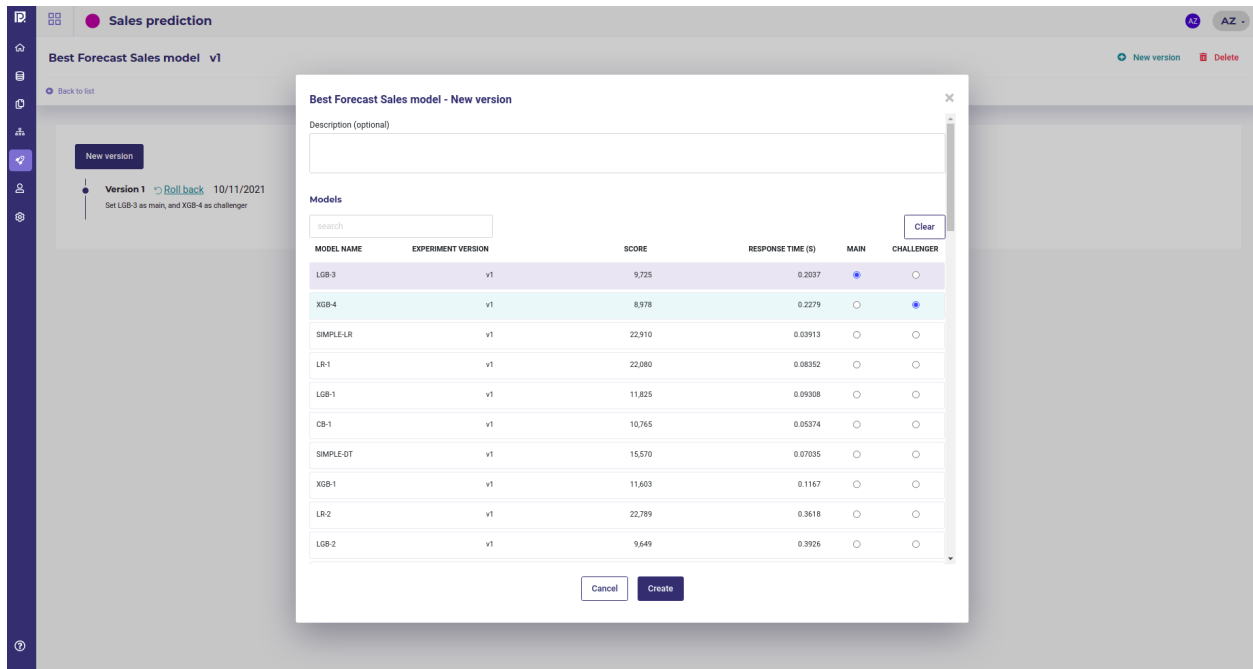


Fig. 48 – Managing version of a deployment

At any moment you can change the model of your experiment that is called, picking it in any version of your experiment. By doing so, you will replace the models called when someone use your deployed experiment without anyone noticing or having to change its client call.

When creating a new version you can change both the main model and the challenger model. You can rollback to a previous version of your deployment just by clicking on the Rollback link. The version clicked will then be used again to do predictions.

## Usage

The usage section displays CPU and memoru usage of your deployed experiment.

### Edit and remove a deployment

You can not edit a deployment but you can remove it either from the list of deployment, with the 3-dots menu on the right of the list items, or on the individual page of each deployment with the Delete button.

#### 5.3.1.6 Contributors

By clicking on contributors on the main menu of a project, you will access the list of all users of the project. If you have sufficient rights (« Admin ») on this project, you will be able to add & delete users from the project and modify users' rights.

## Roles & rules

- Viewer : you can access to all pages (except project settings) with no possibility of creation or edition
- Contributor : viewer rights + you can edit and create resources inside the project
- Admin : contributor rights + you can manage users and modify project properties

## Add & delete

If you are admin in a project, you can manage users into your project.

To add a user, you have to enter the collaborator email into the top left field, set his right using the dropdown menu and click on the “invite this collaborator”. Please note that you can only invite collaborators that already have a prevision.io account.

To change the rights of a user into the project, you just have to select the new role using the dropdown. To be sure that the project and users properties can be managed, at least one collaborator have to be admin of the project.

To remove a collaborator from the project, use the trash button on the left side of the list.

**Project's collaborators**  
Share this project with collaborators

Collaborator's email  admin

NAME	EMAIL	ROLE	
Axel Chauvin QA test	axel.chauvin@prevision.io	admin	
Simon Levacher	simon.levacher@prevision.io	admin	

## Project settings

If you are admin on a project, the project settings button is enabled and, by clicking on it, you will access the project setting page.

**Project's settings**  
Edit the project's settings

**SETTINGS**

Project's name  
QA - Multiclassification 24 / 40 characters

Project's description (optional)  
0 / 210 characters

Set a color to this project

**PREVIEW**

**QA - Multiclassification**  
axel.chauvin@prevision.io  
05/18/2021, 11:06

2 4

Collaborators:

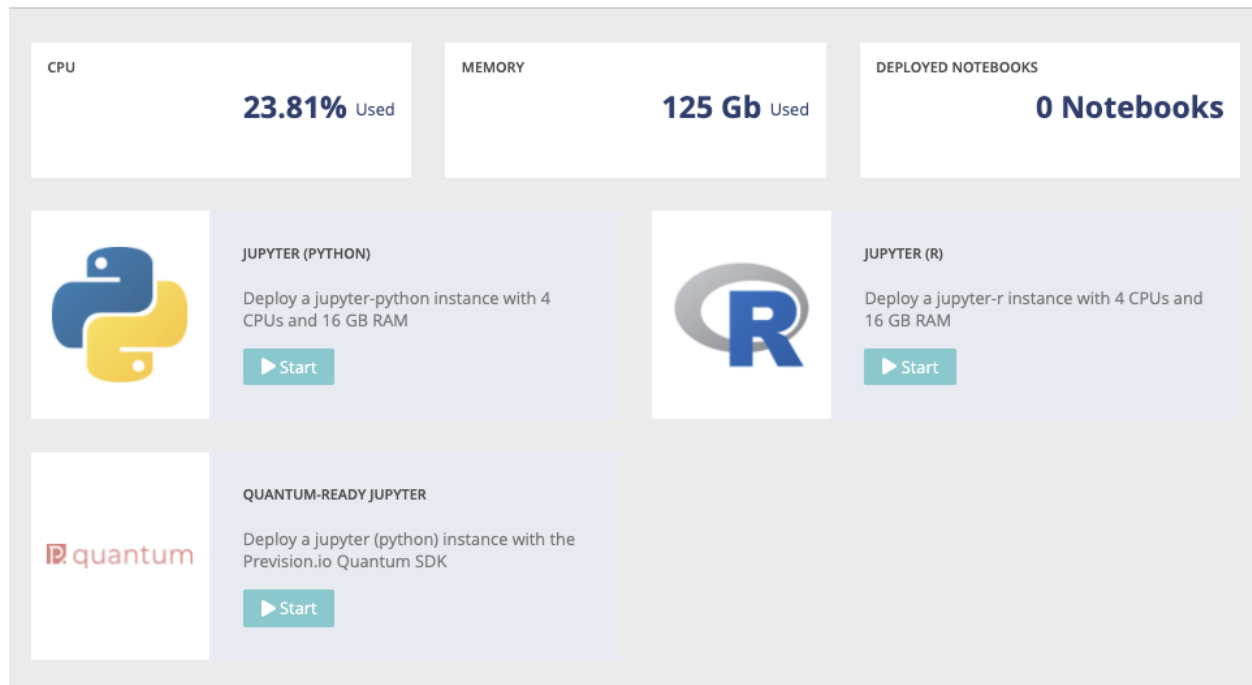
On this page, you can :

- update name, description and color of your project
- delete the project. Please note that if you delete a project, all resources linked to the project will be deleted (experiments, datasets, deployed models, etc.)

### 5.3.1.7 Notebooks

#### Introduction

Prevision.io offers various tools to enable data science use cases to be carried out. Among these tools, there are notebooks and production tools. Notebooks are not scoped into projects. You can access notebooks by clicking on the notebook button on the left main menu. Then you will be redirected to the following page.



#### Jupyter (python)

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](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/>

#### Jupyter (R studio)

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

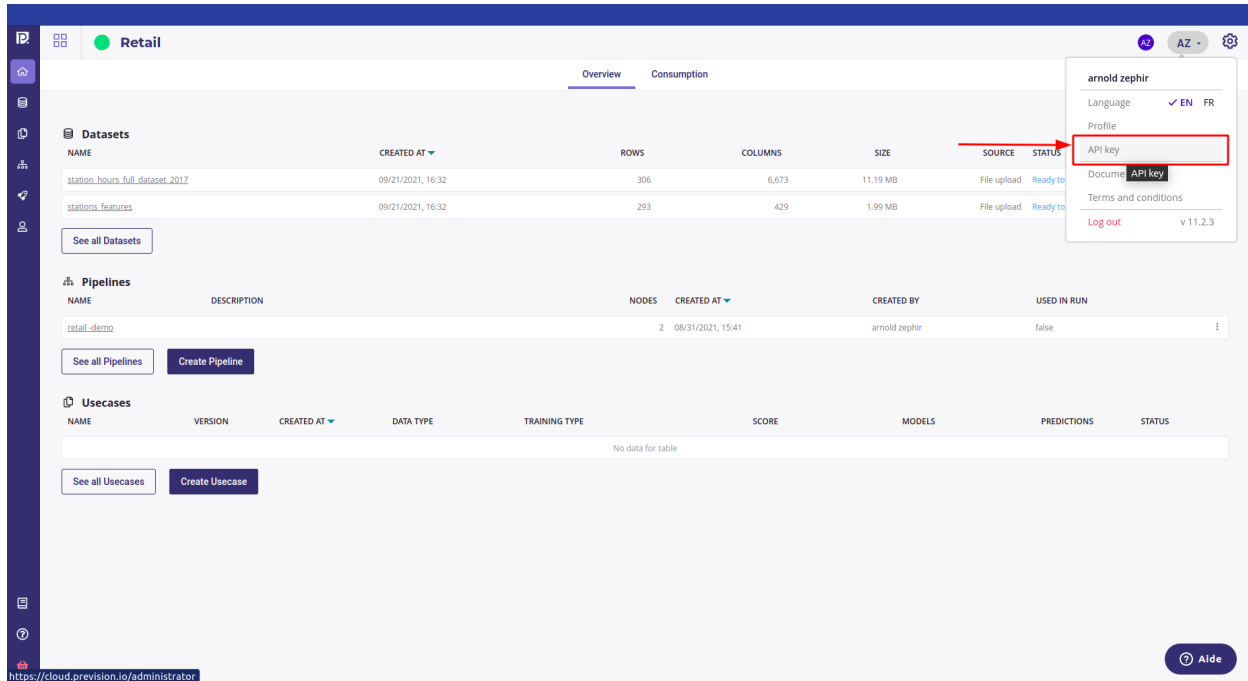
Note that a set of packages is preinstalled on your instance (list : [https://previsionio.readthedocs.io/fr/latest/\\_static/ressources/packages\\_R.txt](https://previsionio.readthedocs.io/fr/latest/_static/ressources/packages_R.txt)), particularly the previsionio package that encapsulates the functions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

## 5.3.2 API

### 5.3.2.1 Using The API

You can access every object generated by Prevision Platform from the [API](#).

To run the Code below you need your Master Token. It is available in the API Key page of your user settings :



You can use the native urllib module to parse API. This page show how to use API with raw python but we suggest to use the SDK ( [Python](#) and [R](#) ) for an higher level of abstraction.

First, import native python 3 urllib.request and set up your Token ( warning : if you have an on promise server or custom dedicated domain, you need to replace the url « cloud.prevision.io » with your own )

```

1 import urllib.request
2 import pandas as pd
3 import ssl
4 import json
5
6 MASTER_TOKEN="<YOUR_MASTER_TOKEN>"
7
8 BASE_PATH ="https://cloud.prevision.io/ext/v1"
9
10 projectsurl = f"{BASE_PATH}/projects"
11 request = urllib.request.Request(projectsurl)
12 request.add_header('Authorization',MASTER_TOKEN )
13 # Disable SSL check
14 projectslist = urllib.request.urlopen(request, context=ssl.SSLContext()).read()
15 projectslist = json.loads(projectslist)

```

### 5.3.3 SDK

The following parts contain information about Prevision.io SDKs, which allow to access to the platform's features programmatically.

If your looking for the full documentation of the software's APIs you will find it [here](#).

#### 5.3.3.1 Using the Python SDK

##### Standard Machine Learning Worflow with Prevision SDK

Here is a list of standard and common workflows you can acheive with Prevision Python SDK. You may read the [Prevision Public github](#) or the [API reference](#) too.

##### External Model

You can get code for runing this guide on the [Getting started guide](#)

First import all the modules

```
1 import previsionio as pio
2 import yaml
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.pipeline import make_pipeline
5 from sklearn.preprocessing import OrdinalEncoder
6 from sklearn.neighbors import KNeighborsClassifier
7 from skl2onnx import convert_sklearn
8 from skl2onnx.common.data_types import FloatTensorType
9 import numpy as np
10 import logging
```

Setup your token account ( see *Using The API* ) and some parameter for your project, like its name, the name of the datasets...

Note that you always must create a Project for hosting datasets and experiments.

```
1 import os
2 from os.path import join
3 from dotenv import load_dotenv
4
5 load_dotenv()
6
7 PROJECT_NAME="Sklearn models Comparison"
8 TRAINSET_NAME="fraud_train"
9 HOLDOUT_NAME="fraud_holdout"
10 INPUT_PATH=join("data","assets")
11 TARGET = 'fraude'
12
13
14 pio.client.init_client(
15     token=os.environ['PIO_MASTER_TOKEN'],
16     prevision_url=os.environ['DOMAIN'])
```

Create a New project, or reuse an existing one

```

1 projects_list = pio.Project.list()
2 # Create a new Project or using the old one
3
4 if PROJECT_NAME not in [p.name for p in projects_list] :
5     project = pio.Project.new(name=PROJECT_NAME, description="An experiment using ")
6 else :
7     project = [p for p in projects_list if p.name==PROJECT_NAME] [0]

```

Add the dataset to the projects or get the existing one if already uploaded ( the dataset will be automatically uploaded to your account when you create them )

```

1 datasets_list = project.list_datasets()
2 for d in datasets_list:
3     if TRAINSET_NAME in [d.name for d in datasets_list] :
4         train = [d for d in datasets_list if d.name==TRAINSET_NAME] [0]
5     else :
6         train = project.create_dataset(file_name=join(INPUT_PATH, "trainset_fraud.csv"),
7         ↪name='fraud_train')
8
9     if HOLDOUT_NAME in [d.name for d in datasets_list] :
10        test = [d for d in datasets_list if d.name==HOLDOUT_NAME] [0]
11    else :
12        test = project.create_dataset(file_name=join(INPUT_PATH, "holdout_fraud.csv"),
13        ↪name='fraud_holdout')

```

Beware to converting the data to the right type before making your dataset

```

1 train_data = train.data.astype(np.float32)
2 test_data = test.data.astype(np.float32)
3
4 X_train = train_data.drop(TARGET, axis=1)
5 y_train = train_data[TARGET]

```

Then train some classifiers ( you may upload many models at once ) and create a yaml file to hold the models configuration.

```

1 classifiers=[ {
2     "name": "lrsklearn",
3     "algo": LogisticRegression(max_iter=3000)
4 },
5 {
6     "name": "knnsk",
7     "algo": KNeighborsClassifier(3)
8 }
9 ]
10
11 initial_type = [('float_input', FloatTensorType([None, X_train.shape[1]]))]
12
13
14 config={}
15 config["class_names"] = [str(c) for c in set(y_train)]
16 config["input"] = [str(feature) for feature in X_train.columns]
17 with open(join(INPUT_PATH, 'logreg_fraude.yaml'), 'w') as f:
18     yaml.dump(config, f)

```

Sklearn Pipeline are supported so you may build any pipeline you want as long as you provide the right config file. Convert each of your model to an onnx file once fitted :

```
1 for clf in classifiers :
2     logging
3     clr = make_pipeline(OrdinalEncoder(),clf["algo"])
4     clr.fit(X_train, y_train )
5
6     onx = convert_sklearn(clr, initial_types=initial_type)
7     with open(join(INPUT_PATH,f'{clf["name"]}_logreg_fraude.onnx'), 'wb') as f:
8         f.write(onx.SerializeToString())
```

And last, use the `Project create_external_classification` method to upload all your models at once in the same experiment

---

**Note :** You can upload many onnx file in the same experiment in order to benchmark them. To do that you must provide a list of tuple, one for each onnx file with :

- a name
- the path to your onnx file
- the path to your config file ( often the same for each model

```
1 external_models=[(clf["name"], join(INPUT_PATH,f'{clf["name"]}_logreg_fraude.onnx'),
2 ↪ join(INPUT_PATH,'logreg_fraude.yaml')) for clf in classifiers ]
3 exp = project.create_external_classification(experiment_name=f'churn_sklearn_{clf[
4 ↪ "name"]}',
5
6                                     dataset=train,
7                                     holdout_dataset=test,
8                                     target_column=TARGET,
9                                     external_models = external_models
10                                )
```

### 5.3.3.2 Using the R SDK

- [Getting started](#)
- [API reference](#)
- [Source code](#)

### 5.3.3.3 Using the Prevision Quantum NN SDK

Prevision-quantum-nn is a library that allows to handle automatically quantum variational circuits. Its main page is available [here](#).

## 5.3.4 Guides and Howto

### 5.3.4.1 Datascience Guide

#### How and when to do data embedding exploration

#### What is this guide about ?

Data exploration is an important step of Data modelisation and Machine Learning projects lifecycle. Even if not needed for supervised modelisation, as algorithm are now powerful enough to build the best model without human insight, Data Exploration may be useful when starting the project :



- to check data quality and integrity ( even if « no data » is still an important insight )
- to check modelisation feasibility with visual hint
- and, more important, to onboard the line of business user and formalize intuition and goal of the project !

The last point is probably the most important. In every Datascience project, first and most important step is to define clear objective that serve purpose. Exploring data with visual tools oftent allow to get insight from business expert and get them involved in the project

## What are we learning in this Guide ?

This guide is splitted in 4 sections :

- How does data exploration takes place in the Machine Learning Pipeline ?
- The principles of vector embedding anything
- How to make data embedding in Prevision Platform ?
- What to do with embedding ?

## Data exploration in the Machine Learning pipeline

### What's data exploration ?

We define data exploration as any process that take raw data and produce indicators and charts for human to analyse. Statistics is a kind of data exploration. Scatter plot and histogram are an other kind.

Sometimes, Humans can build very basic models from statistical indicators and get rules-based model like *if age > 40 then wants\_motobike = true*.

In a Machine Learning project pipeline, Data exploration may serve the 3 following purpose.

### Getting insight

Before any modelisation, the first step of any machine learning is , or at least, should be, data exploration.

Before Big Data and Machine Learning advent, most of data analysis where done visually with and data Insight were extracted by human from statistical indicator and charts.

Data Analysis is getting replaced with Artificial Intelligence and Machine Learning for understanding phenomena and building models but human mind is still great at getting insight from visual clue.

Exploring data may still help to build the target, decide modelisation type or find an innovative feature engineering. Sometimes it serves to detect underrepresented category and add some weigh to the the data.

**Avertissement :** Do not build segment for Machine Learning and AI problems ! Segmentation was a great way to build basic models but Big Data and Machine Learning tools do not need Segmentation anymore as they works on individual sample. Only use segmentation to build basic rules-based model or explore data and understand problem. But if the project need performance, use supervised learning of IA unsupervised technics.

## Checking the data quality for modeling

Even if building a model only from data exploration is not the best way to get performance, data exploration can serve as a show stopper as it can highlight two main issue from your data before going on the modelisation step :

- random or noisy data
- unbalanced data

There are specific indicators for noisy data or signal/noise ratio and this can be seen from some specific visual representation. Looking after a too low signal/noise ratio is a good way to avoid poor modelisation due to poor data quality.

About unbalanced data, it's still possible to get good models if the low rate target has some very specific features, which will probably appear in the data exploration process, but as a rule of thumbs, looking for general shape of data and some under-representation of sur-representation for planning some kind of weighting is considered as good practice.

### Talking with the Line of Business

Most important output of running data exploration, especially with visual tools, is to onboard the line of business manager into the datascience project.

Success for a datascience project often rely on building the good target, that serves a true business purpose. By running a data exploration phase with someone from the business, you can, as Data Scientist Practitioner :

- get insight to build your metrics and objectives, and thus optimize the model for R.O.I
- get the Lob manager involved and build a relationship to build the fittest model for business.

### Data exploration technic focus : Data Embedding

Data exploration often relies on the 3 following methods :

- build statistics for each feature ( average, median, minimum value, number of occurrence, mode, ... )
- build charts ( histogram, pie chart,... )
- build chart about some relation between features ( bivariate analysis, correlation matrix, .... )

In Prevision.io platform, statistics and charts are produced on the dedicated features page

Home > [my project] > [my experiment] > Features

and

Home > [my project] > [my experiment] > Features > [ my feature ]

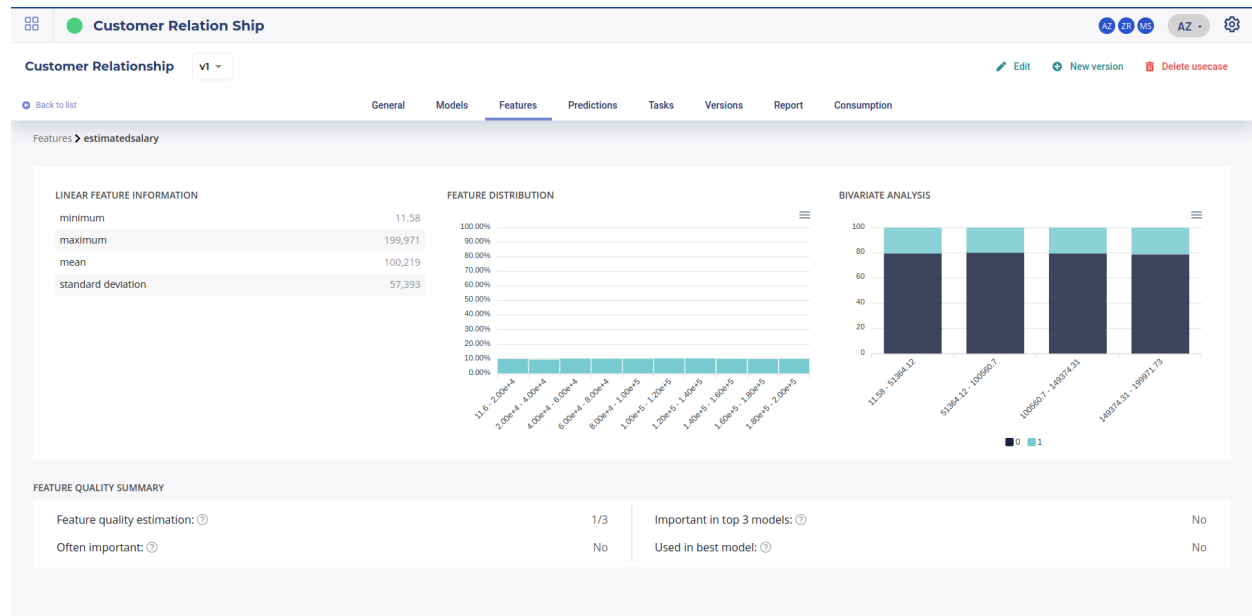


Fig. 49 – Statistics available on experiment only ( note the bivariate analysis related to the target )

or

Home > [my project] > [my dataset] > General

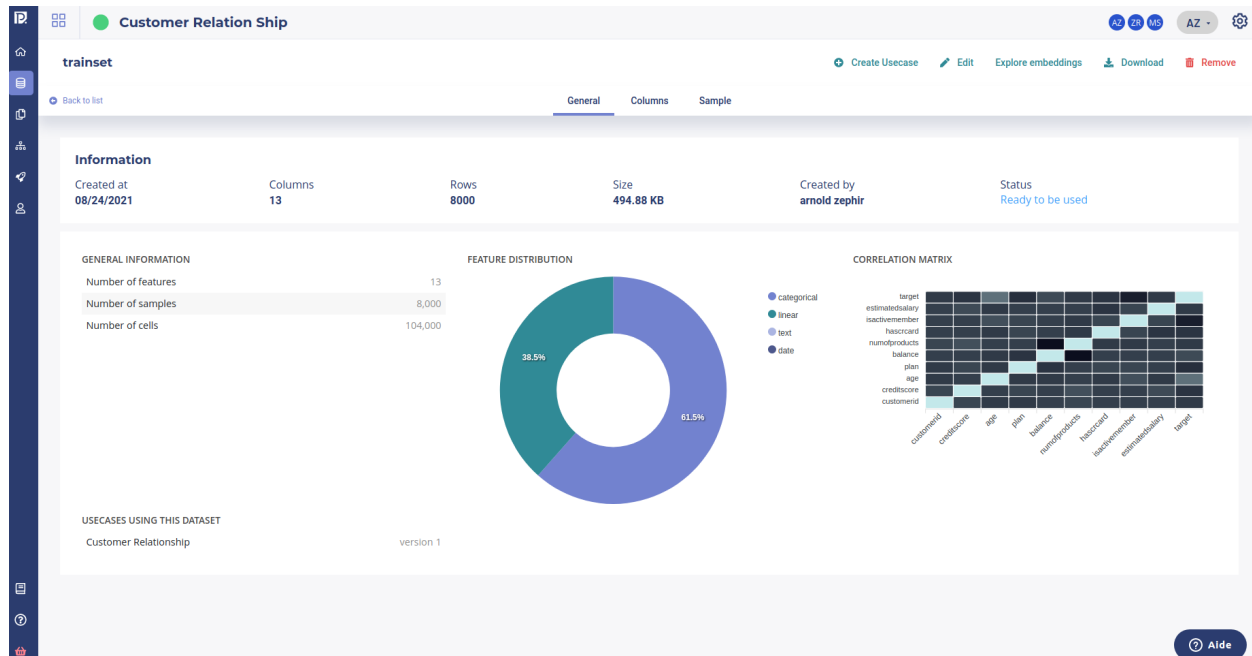


Fig. 50 – Statistics available on each new dataset ( automatic computing )

**Note :** Some statistics and chart are built only once the use case ( see [Experiments](#) ) has been defined because bivariate chart related to the target are built. Some others are available few seconds after you created new [Data](#)

Basic statistical indicators help to better understand the dataset yet more powerful technics and tools exist based on continous vector built upon the data.

Embedding technics are various way to transform data such that you go from a discrete ( categorical and such ) representation of data to a continous one with mathematical vectors. It is a very important method as it allow to run mathematical operations on all kind of data ( cosine similarity, difference, addition ) while preserving relationship between features. When embedding data, each sample of the dataset is transformed into a vector with fewer dimensions. This vector may be used to build chart, compute similarities between sample, cluster data or detect outliers.

**Note :** Four usage of Data Embedding :

- visualize cluster
- compute similarities between sample
- detect outliers
- visualize segment relative weights

There are many technics to build embedding but here are the most common

## PCA

PCA ( Principal Common Analysis ) is based on Matrix eigen vector and eigen values. When applied on a dataset, it find eigen values and eigen vector of the data and resulting vectors can be interpreted as « axes of greater variance ». It often put emphasis on feature correlation and is used as a dimension reduction algorithm

**Note :** Let's say that you got a dataset with 10 samples of 5 features

X1	X2	X3	X4	X5
fr	43	10	50	5
fr	43	4	20	3
en	13	7	35	3.5
en	12	20	100	10
en	12	34	170	17
en	13	18	90	9
fr	41	18	90	9
en	12	20	100	10
fr	43	32	160	16
fr	43	64	320	32

Even if there are 5 features, a PCA wil show that the samples are in fact variation of 2 vectors :

- a first one highly correlated with (X1,X2) features
- another one correlated with (X3,X4,X5)

Thus this 10 sample may in fact be written as two dimensionnal features :

V1	V2
0	0.1562
0	0.0625
1	0.1094
1	0.3125
1	0.5312
1	0.2813
0	0.2813
1	0.3125
0	0.5
0	1

Where :

- $X1 = \text{« fr » if } V1 == 0 \text{ else « en »}$
- $X2 = -30 * V1 + 43$

and

- $X3 = 64 * V1$
- $X4 = 320 * V1$
- $X5 = 32 * V1$

---

PCA may be used as some kind of feature importante

## VAE

Variationnal Auto Encoding is a more powerful technic trying to compress data with less features that the original dataset. For example, if a dataset has 300 features but the compression algorithm can built a dataset with 30 features that is able to reconstruct the original dataset without losing too many signal, the theory says that the analysis can be done on the 30 features without loss of meaning or signifiance.

Variationnal Auto Encoding often use a Neural Network that is tasked to generate to output the vector presented in input but with few neurons ( for example, only 4 ).

The signal in the deep layer of this Network may be interpreted as a vector representation of the sample, call embedding, and has the interesting property that you can build a distance metric such that *similar samples have a small metric distance*.

Given this property you can build your analysis on the embedding space.

This is this technology that Prevision.io platform uses for data exploration

## Data Embedding exploration in Prevision Platform

Prevision platform offers two features for data exploration :

- building the embedding
- a tool for exploring the embedding (« The Explorer »)

## Building the Embedding

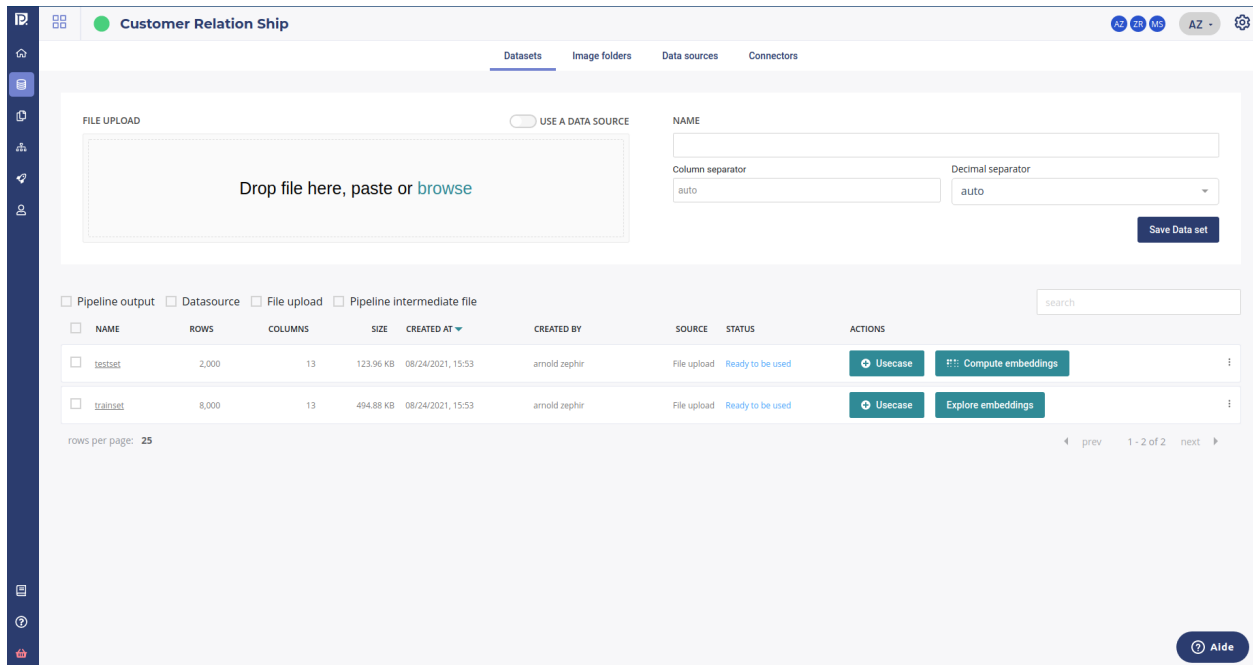


Fig. 51 – The datascreen, where to launch an embedding

The « Compute Embeddings » button is available in the *Data* section of the platform. When you Upload or import a new dataset, it becomes available when the data importation is done.

As this is a cpu intensive algorithm, user must explicitly launch it by clicking on the button. It lasts about a few minutes and once done, you can explore your data with the « Explore Embeddings » button.

## The Data Explorer

The data explorer is an interface to handle data rows embedding, or at least a subsample (5000) of it. It projects the data onto a 3D or 2D vector space and give some tools for exploring data :

- the search and similarity sidebar, on the left, displays nearest neighbors for 2 metrics, cosine similarities and Euclidean distance, when clicking on a point. The number of nearest neighbors is a parameters that user can change. Note that you can isolate a point and its neighbors for further investigation

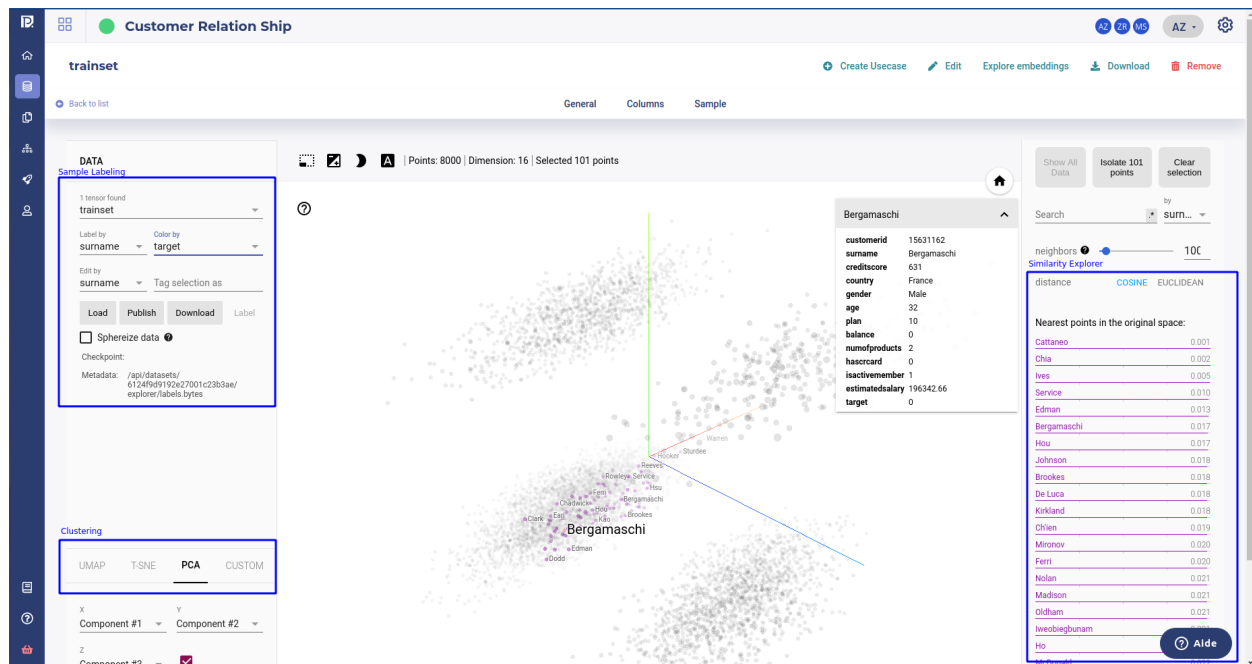


Fig. 52 – The data explorer. Draw embedding vector on a 3D charts

- the labeling box on the top left corner allows to assign labels and color to sample along some feature. You can display modalities ( or values ) of features as a label or as a color.
- the clustering box on the bottom left let user launch a clustering :
  - PCA : fastest but do not respect distance
  - TSNE : better but slow and complex to use
  - UMAP : less good than TSNE in respecting distance but fastest and a little bit easier to use

The central window displays a navigatio interface when you can pan, rotate and zoom.

## Getting and Working with the Embedding Vector

### Visualize cluster

To build cluster, and segment, you just need to launch a clustering computing by using the clustering section. There are 3 methods of “clustering” :

- PCA
- TSNE
- UMAP

Each methods has its own set of parameters that build cluster more or less constrained.

**Note :** This are not methods of clustering *per se* as the clustering algorithms assigns a class ( cluster number ) to each point of the dataset. Here are algorithm that visually bring together data rows that share similarities ( ake whom distances are small ). It allows visual inspection by a human mind, which is is often better to make generalities than algorithm.

## PCA

PCA is a very cheap method but fast to compute. It sometimes highlights very simple variance axes and give some insight but do not expect much. The interesting thing in PCA algorithm is that it computes the explained variance for each component :

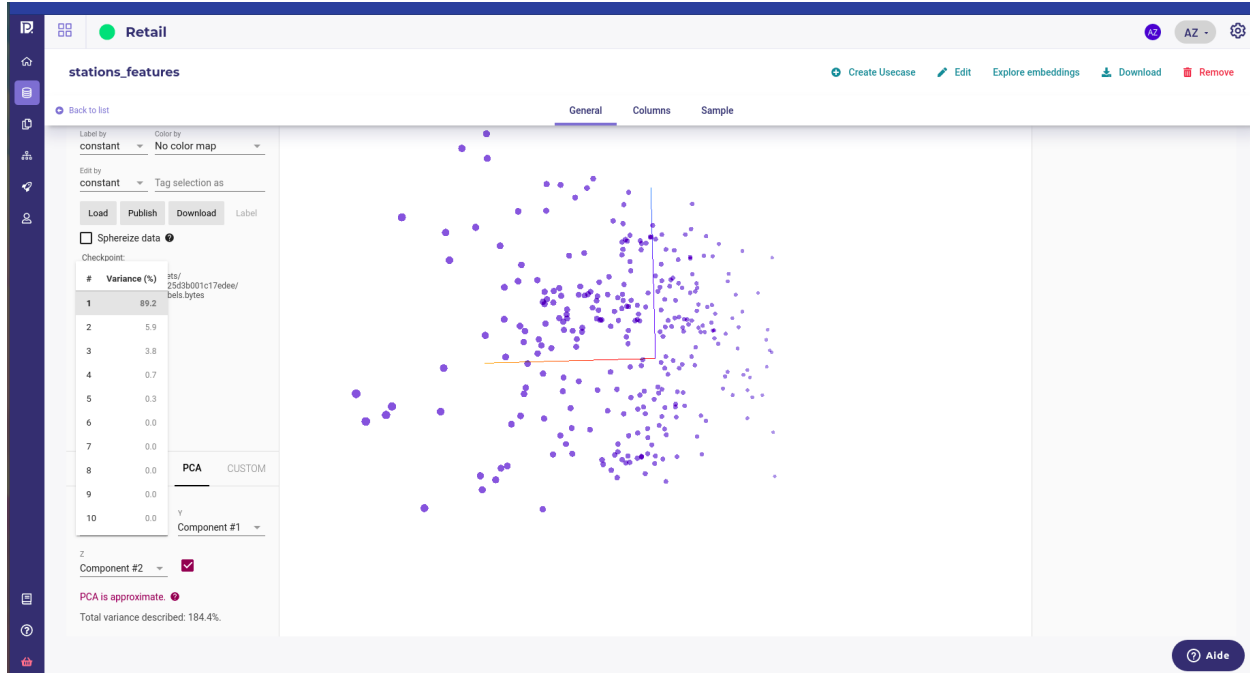


Fig. 53 – PCA Component explained variance

Thus, you can quickly see if your data has in fact a low dimensionnality.

On the example above, we see that one component hold 89% of the data variance so even if the original dataset has 428 features, there seem to share a lot of information.

## TSNE

TSNE is the most interesting clustering method as it can lead to very defined, and separated, segment when they exists. Yet it can be quite long to compute and hard to tune.

As a rule of thumb :

- the higher the perplexity, the more defined the shape but the longer to converge.
- with a smaller learning rate, convergence takes longer but point are well placed

Note that you can constraint cluster to a feature, to force the algorithm to split the cluster along this features but do it only if you want to confirm some intuition.

## UMAP

UMAP is way faster than TSNE and get good result to put most similar point together but is less able to generate well blocked shape. Yet, it's often good to use it to start exploration and then switch to tsne to validate some hypothesis.

UMAP has only one parameter, that is number of Neighbors. The more neighbors you allow, the larger and more inclusive are the structure of cluster.

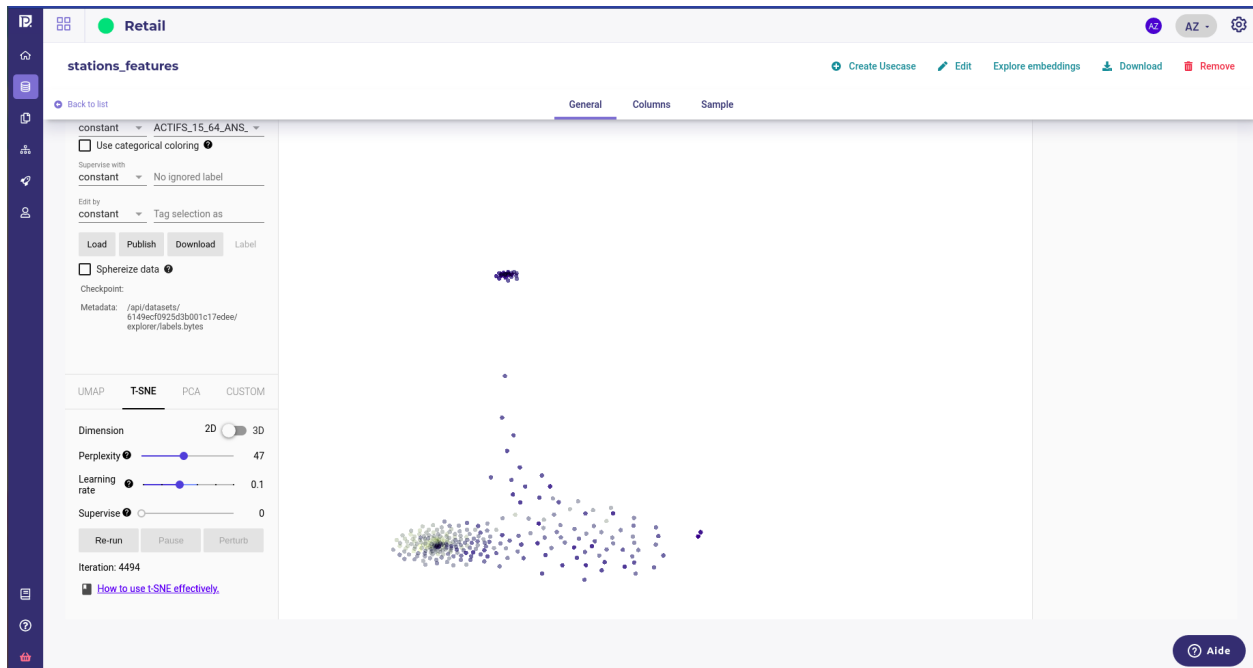


Fig. 54 – Clear cluster of Paris Subway station ( residential, job and tourism )

Once you got some visual cluster, it's time to explore points and their similarities but let's talk about similarities, distance and proximity in 3D spaces.

**Note :** As we said before, the algorithms used by Prevision build vector space where you can build distances metrics such that similar sample have a small distance. Yet, in most of case, the list of similar points displayed on the right will not always be grouped together on the central window after a clustering run because the visual explorer is only 2D or 3D. Similarities distance , cosine similarity or Euclidean distance, are computed on the whole dimension of the embedding space as UMAP and TSNE are computed so that similar points are near together in the 2D/3D space. Yet is not always possible to respect every constraints and sometimes, some points with a small distance will be far in the 3D space. When it happens, always keep the similarities distance as the truth.

## Explore sample and look after similarities between samples

Whatever the representation you used, PCA, TSNE or UMap, you can always click on a point, which represent a row of your dataset.

When you click on a point :

- its features are displayed in a small dropdown windows
- a set of similar samples are highlighted and displayed on the similarities windows on the right sidebar.

In this window you can :

- select the number of neighbors displayed
- change the distance used, cosine ( dot product ) or euclidean distance
- search and highlight for specific value on specific feature ( Note : you can use regexp for filters )
- isolate for analysis all the highlighted points
- clear the selection



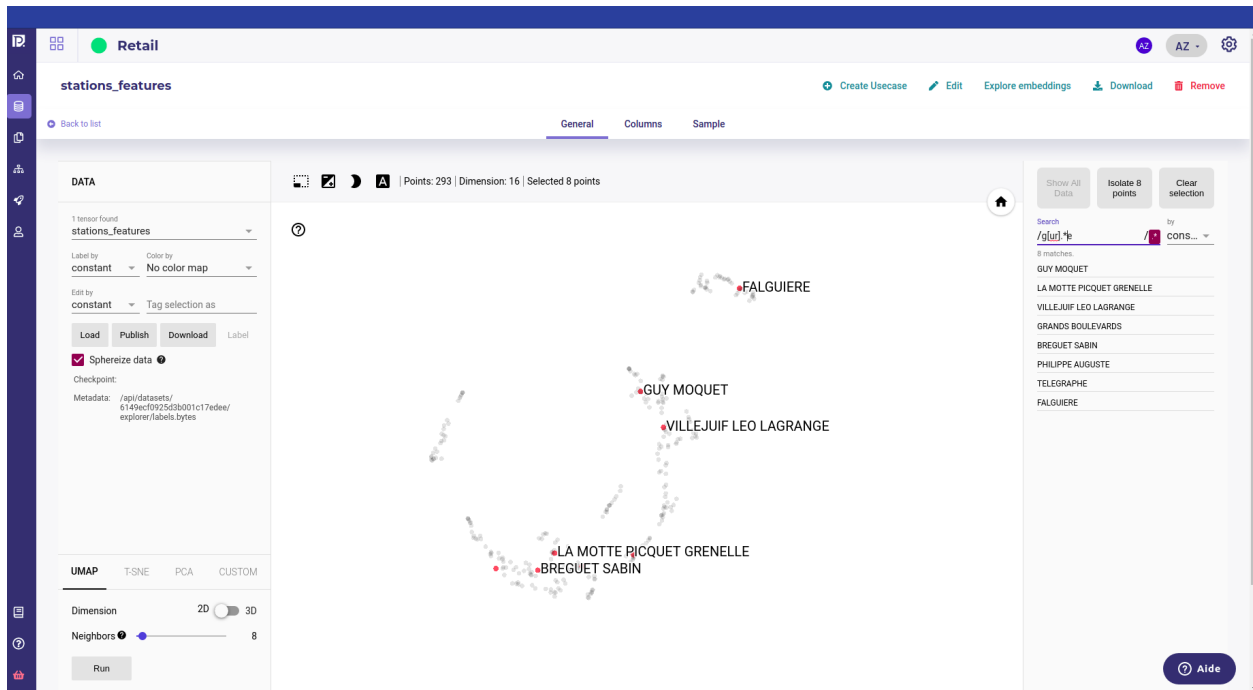


Fig. 55 – filtering all the station whom name match `g[ur].*e`

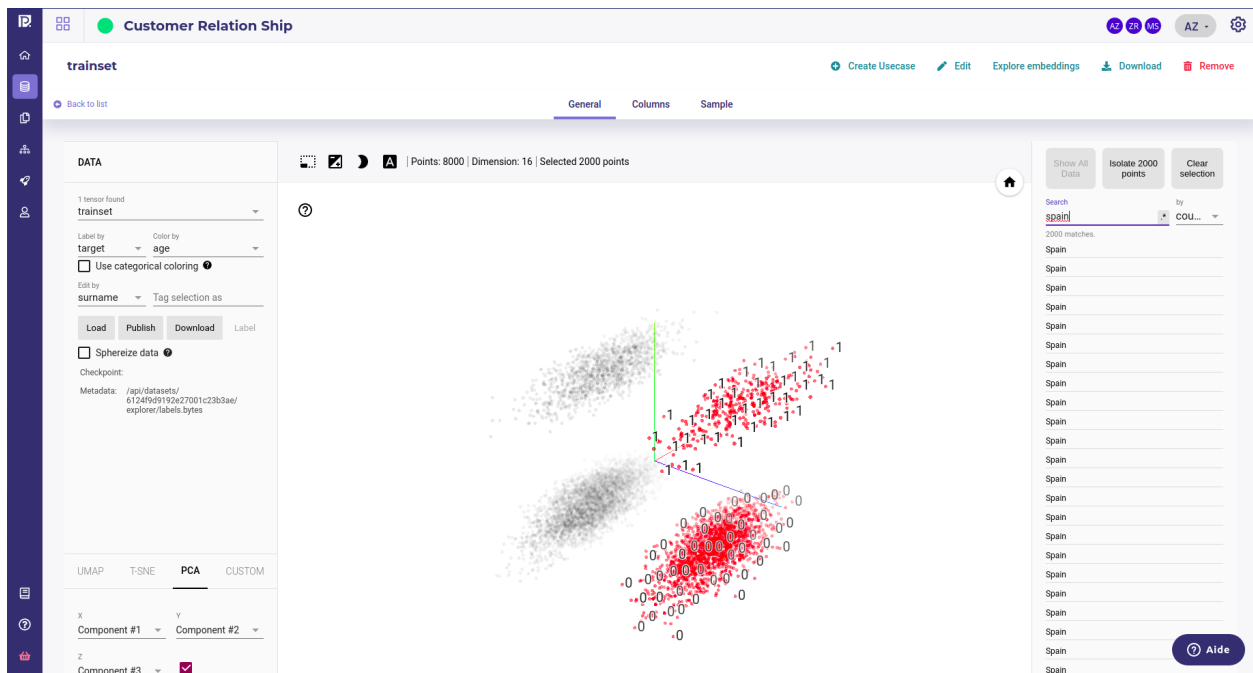


Fig. 56 – In this dataset exploration, user has highlighted all the customers from Spain and display the target.

Once you have isolated some sample, the cluster then run only on this selection, allowing to target your analysis on a specific segment. Thanks to the sample labeling box on the left you can label and color your sample along a feature.

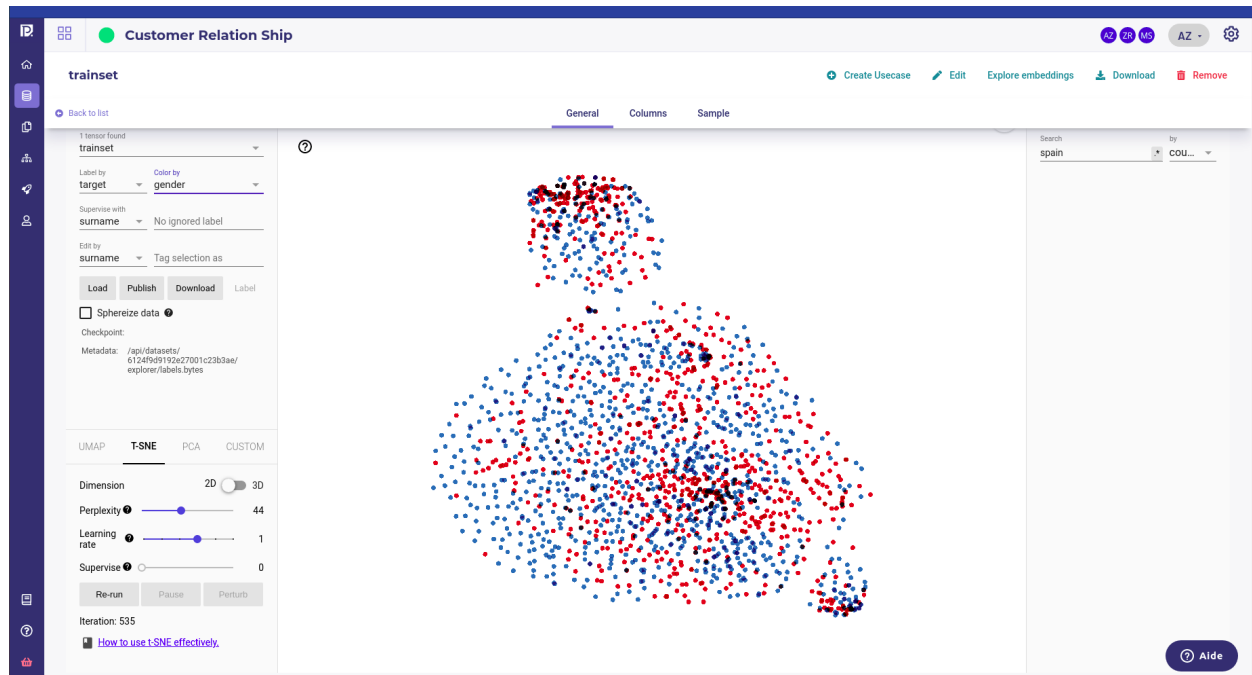


Fig. 57 – The spanish user have been isolated and a cluster ran on this segment only. The gender is used has coloring showing that gender is not a main concer in churner split

## Detect outliers

One of the positive byproduct of Embedding technics is that you can easily detect outliers visually. As the vector are build so that sample grouped by similar feature distribution and covariance, points that are visually outside any shape can be interpreted as outlier.

Using a small perplexity for tsne isolate outliers. You can then click on any point of this outlier group to select it and its neighbors, if some, and then isolate them

Once isolated, you can color or select each point in order to understand why this samples stands out.

## Going Further and use the API

Like every feature of Prevision Studio, you can get the embedding over the [API](#).

The embedding are saved as numpy32 float and is an array. From them you can do many things :

- rerun your own clustering technics
- make histogram and stats along each axes in order to understand the meaning
- anything you want

To run the Code below you need :

- your Master Token. It is available in the API Key page of your user settings :
- The Id of your dataset embedding, wich is available in the url or in the

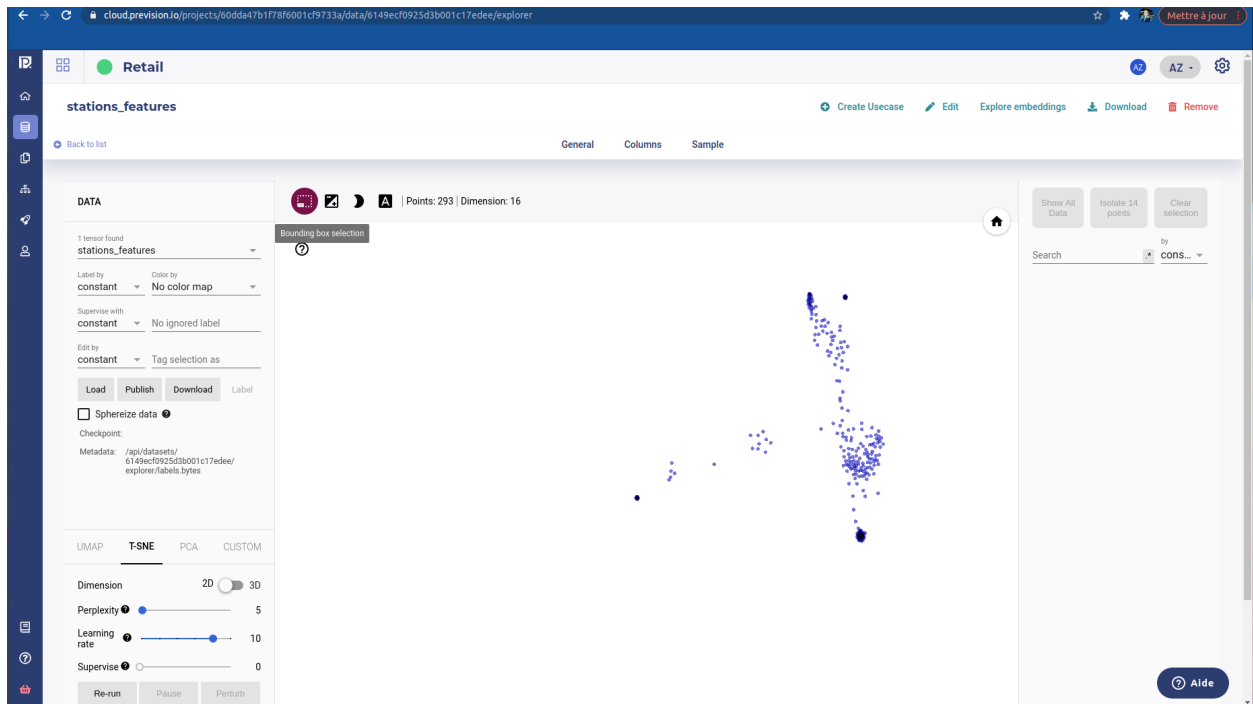
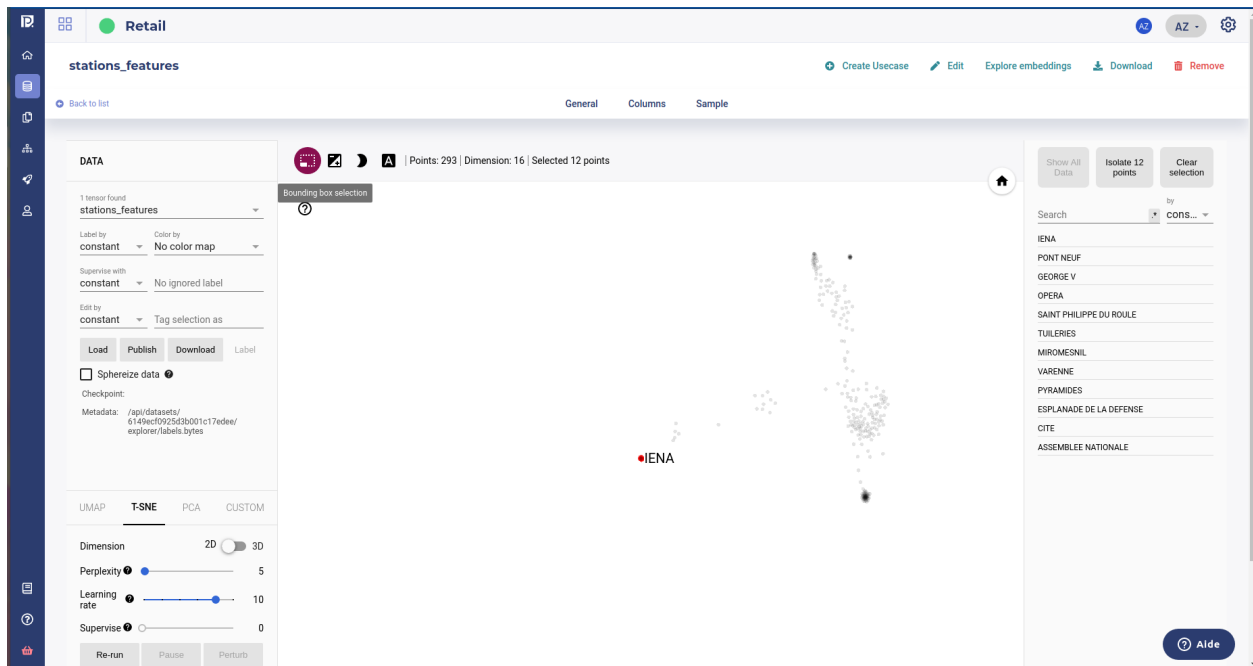
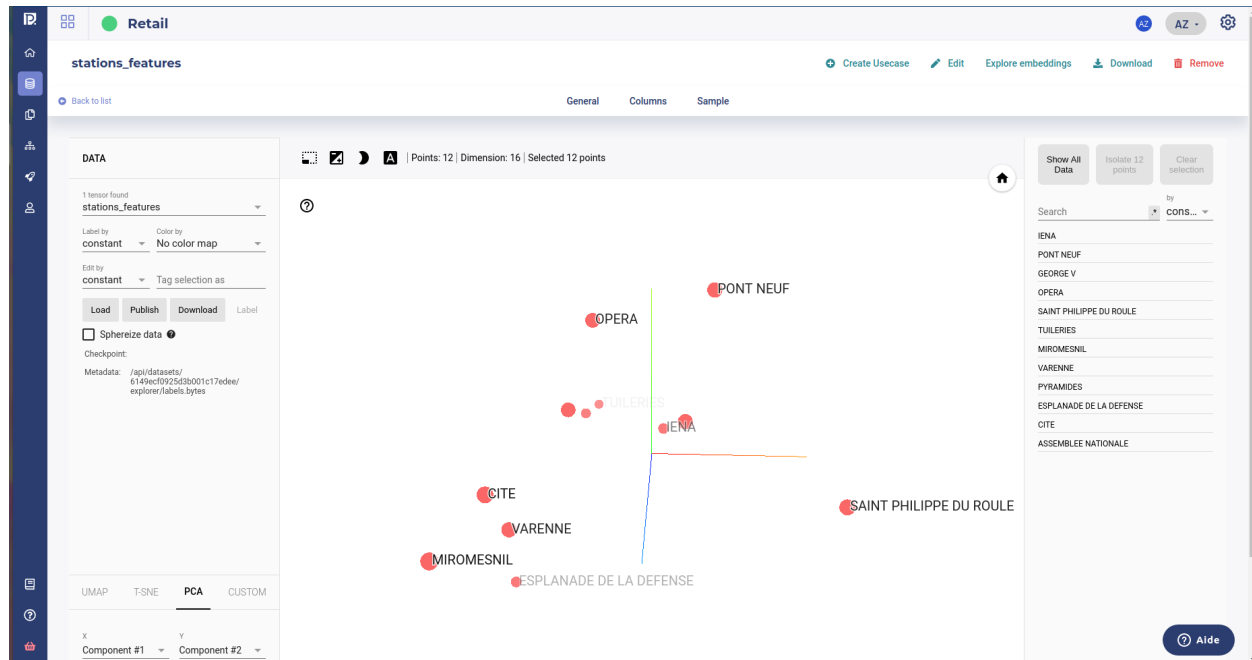


Fig. 58 – A group of isolated outliers





The screenshot displays the 'Overview' tab in the Prevision.io interface. The top section shows a table of datasets with columns: NAME, CREATED AT, ROWS, COLUMNS, SIZE, SOURCE, and STATUS. The bottom section shows a table of pipelines with columns: NAME, DESCRIPTION, NODES, CREATED AT, CREATED BY, and USED IN RUN. The bottom right corner has an 'Aide' button.

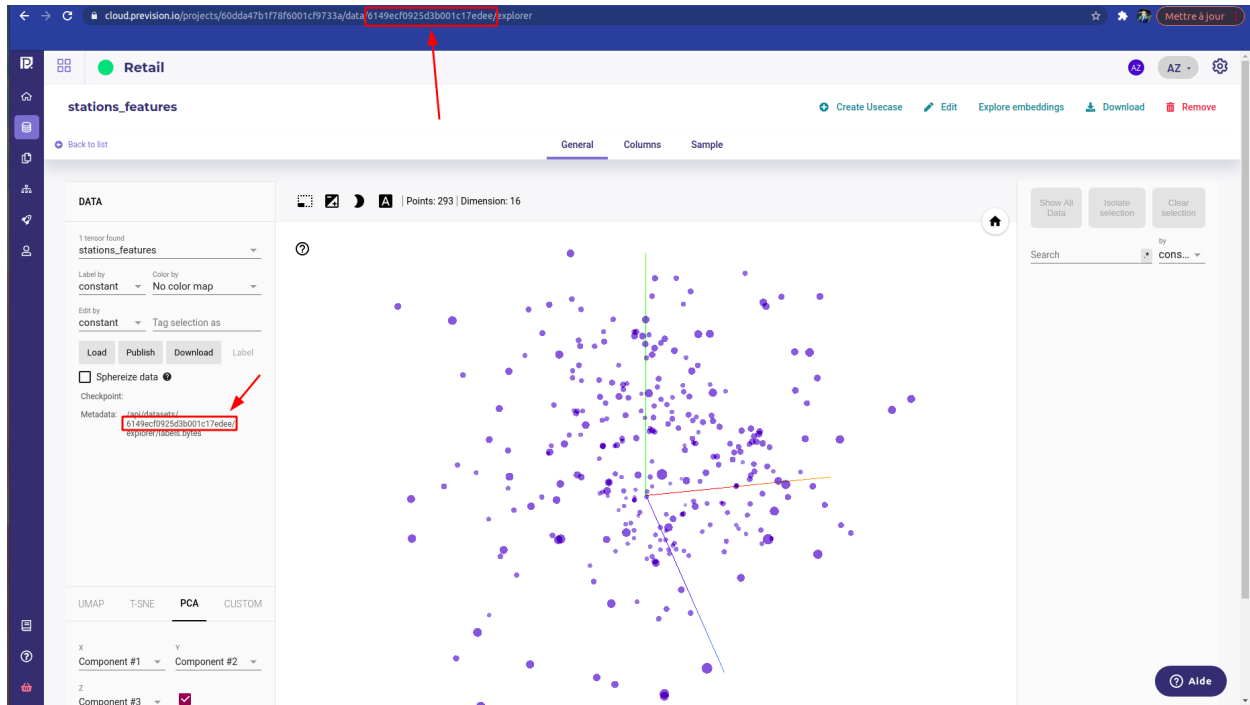
NAME	CREATED AT	ROWS	COLUMNS	SIZE	SOURCE	STATUS
station_hours_full_dataset_2017	09/21/2021, 16:32	306	6,673	11.19 MB	File upload	Ready to
stations_features	09/21/2021, 16:32	293	429	1.99 MB	File upload	Ready to

NAME	DESCRIPTION	NODES	CREATED AT	CREATED BY	USED IN RUN
retail_demo		2	08/31/2021, 15:41	arnold zephir	false

NAME	VERSION	CREATED AT	DATA TYPE	TRAINING TYPE	SCORE	MODELS	PREDICTIONS	STATUS
No data for table								



You can use the native urllib module to parse API but you need pandas and numpy to use data.

First, import native python 3 urllib.request and set up your Token and url built from dataset id ( warning : if you have an on promise server or custom dedicated domain, you need to replace the url « cloud.prevision.io » with your own )

```
1 import urllib.request
2 import numpy as np
3 from io import BytesIO
4 import pandas as pd
5 import ssl
6 import json
7
8 MASTER_TOKEN="<YOUR_MASTER_TOKEN>"
9
10 BASE_PATH ="https://cloud.prevision.io/ext/v1/datasets"
11 DATASET_ID="<dataset_id>"
12
13 meta_url    =f"{BASE_PATH}/{DATASET_ID}/explorer"
14 labels_url  =f"{meta_url}/labels.bytes"
15 dataset_url = f"{meta_url}/tensors.bytes"
```

Then, get the meta information of the embedding, especially the shape of the tensors generated

```
1 # Meta info
2 request = urllib.request.Request(meta_url)
3 request.add_header('Authorization',MASTER_TOKEN )
4 # Disable SSL check
5 meta = urllib.request.urlopen(request, context=ssl.SSLContext()).read()
6 meta = json.loads(meta)
7 tensorShape = meta["embeddings"][0]["tensorShape"]
```

If needed for readability, you can get the originals labels

```
1 # Labels
2 request = urllib.request.Request(labels_url)
3 request.add_header('Authorization', MASTER_TOKEN )
4 # Disable SSL check
5 labels = urllib.request.urlopen(request, context=ssl.SSLContext()).read()
6 labels = pd.read_csv(BytesIO(labels), sep="\t")
```

And last, get you embeddings, that are float32, and reshape them according to the meta information you got before.

```
1 # Tensors
2 request = urllib.request.Request(dataset_url)
3 request.add_header('Authorization', MASTER_TOKEN )
4 # Disable SSL check
5 vec = urllib.request.urlopen(request, context=ssl.SSLContext()).read()
6 vec = np.frombuffer(BytesIO(vec).read(), dtype="float32")
7 vec = vec.reshape(tensorShape[0], tensorShape[1])
8
9
10 df = labels.join(pd.DataFrame(vec))
```

Hence, you have a dataframe containing your original data and their embedding value, that may be use for any operation.

## Full ML pipeline : From data collection to deploying using Prevision.io

*How to release a model across all your organisation in one morning ( and stop spending 24 man-month on a model that will never go into production ) ?*

---

### What is this Guide about ?

This guide is walkthrough for delivering ( very ) quickly a complete Machine Learning Project by using the [Prevision.io platform](#)

The guide details each standard step of a Machine Learning project, from data to model usage across the organisation, and shows how to accomplish them in the platform.

We use historical sales data and intend to build a sales forecasting model.

---

### What's in this guide ?

#### Starting point

This guide assumes that :

- you got a [Prevision.io platform](#)
- the IT Teams put some historical sales data in a database and gave you access ( but if not, csv files are provided below for the sake of this guide )
- An objective has been defined by the Line of Business.

#### Steps

The steps of our guide will be :

Tableau 6 – Steps of a Machine Learning Project

Step	Name	Goal	LoB	IT	Data-lab	Output	time spent
1	Data acquisition	Get the historical data for training Machine Learning Model	No	Yes	Yes	dataset	5mn
2	Feature engineering	Prepare the dataset	No	No	Yes	Holdout and validation strategy ( fold )	20mn
3	Define the problem	Define a metrics that reflects LoB process	Yes	No	Yes	A consensus	As Much as possible
4	Experiment	Train models	No	No	Yes	~100 Models	25mn
5	Evaluate	Get the fittest model	Yes	No	Yes	A selection of 3 to 4 models that go in production	As Much as possible
6	Deploy	Share the model accross the organisation	No	No	Yes	Webapp for human, API for machine	5mn
7	Schedule	Schedule predictions	Yes	Yes	Yes	Prediction delivered each Monday a 9 :00 am in CRM software	20mn
8	Monitor	Track the model in situ	Yes	No	Yes	Dashboard	As Much as possible
TO-TAL							1h15

For each of them, the guide explains what to expect from this steps and how we accomplish it.

## Results

At the end of this guide :

- LoB will get a weekly sales forecast each Monday at 09 :00 AM
- LoB will get a simulator for testing hypothesis over the model
- Applicative team will get an API for calling the model in their own Application
- IT Team will get a dashboard to monitor model Quality of Service

## So what's the job of a datascientist anyway ?

Datascientist is not a developer, even if her or his main tools is code.

Datascientist salary ( and scarcity ) are quite high. If your datascientist spends most time coding models, you should get a developer.

The time spent of a datascientist in an organisation should be spent with the Business owner on the 3th and 5th steps :

- DEFINE METRICS THAT REFLECTS A TRUE BUSINESS ISSUE
- CHOOSE THE MODEL THAT SOLVES THE PROBLEM THE BEST

This are the two most important points for a [project to deliver R.O.I](#) and using tools enables to focus on what matter.

You can [open a free account](#) to practice the following steps. When your account is ready, create a Project to host the assets

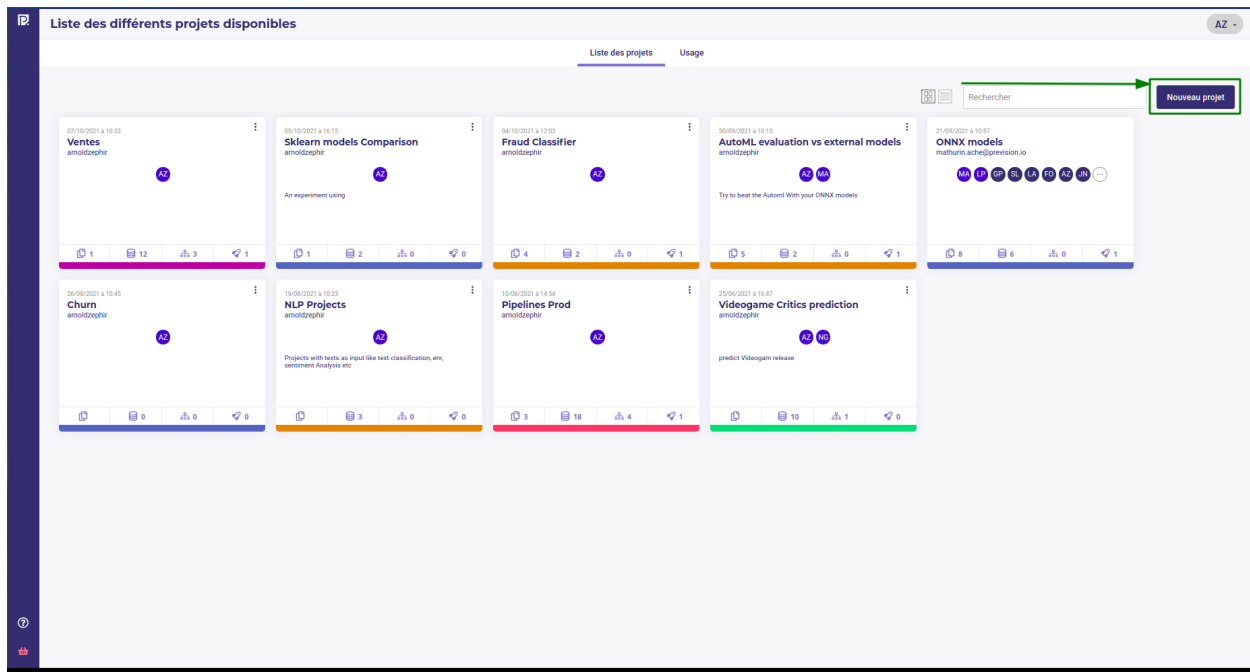


Fig. 59 – Create a new project

## Data acquisition

First step to any project is getting historical data in order to train our Algorithm. As the name implies, Machine learning is all about reading historical data and let a computer model learns to predict a target.

The datas should have been loaded into a database by the IT Team and they have generated credentials for you. Once you have created your project, and selected it :

- go to the data section ( sidebar on the left )
- create a new connector and provided the credentials
- create a new datasource from the db and table of past sales
- Import it as a dataset

If available, you could import recent sales as an holdout dataset in order to validate and check stability of your model :

Data acquisition is done, you can now start to model

Tableau 7 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn

## Download data for testing by yourself

if you don't have database credentials, you can use the following files. Just import file instead of using a datasource when importing dataset.

- the trainset.
- the holdout.



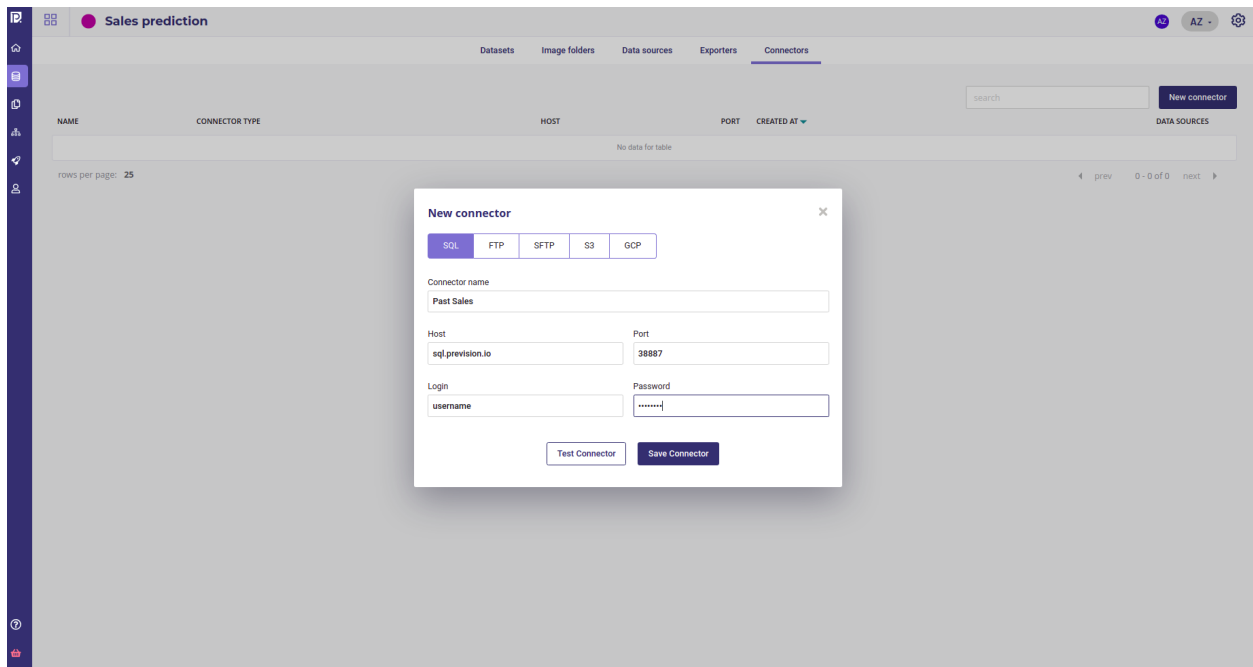


Fig. 60 – Create a new connector

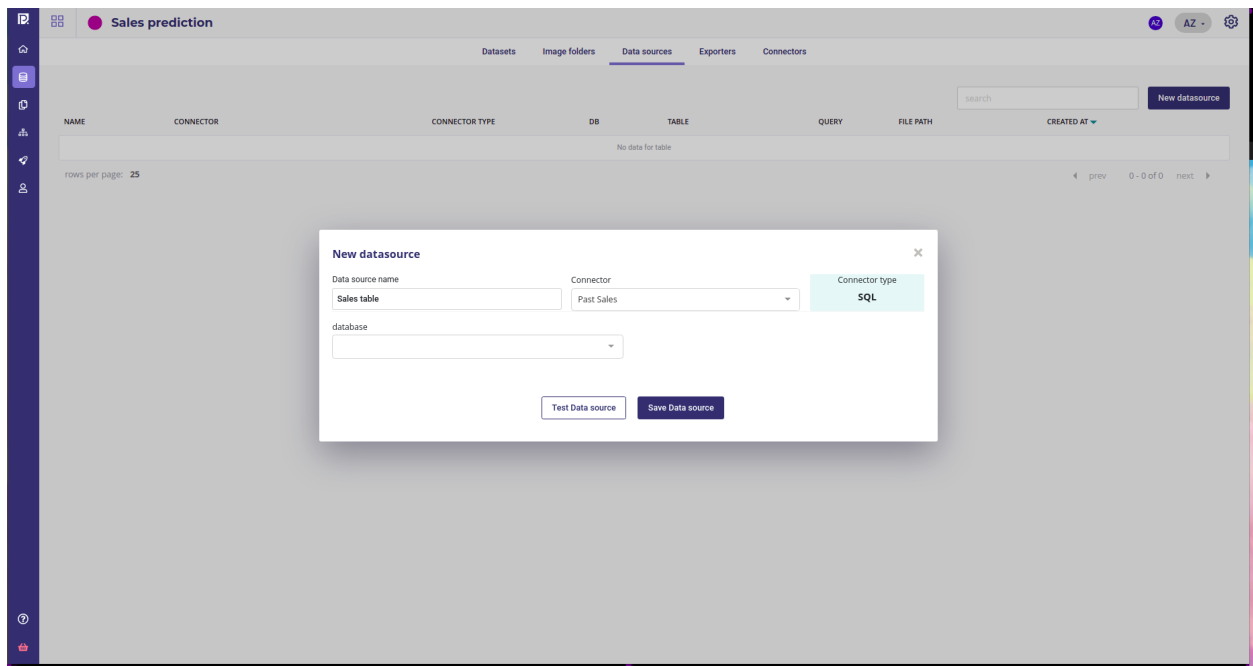


Fig. 61 – Create a new datasource

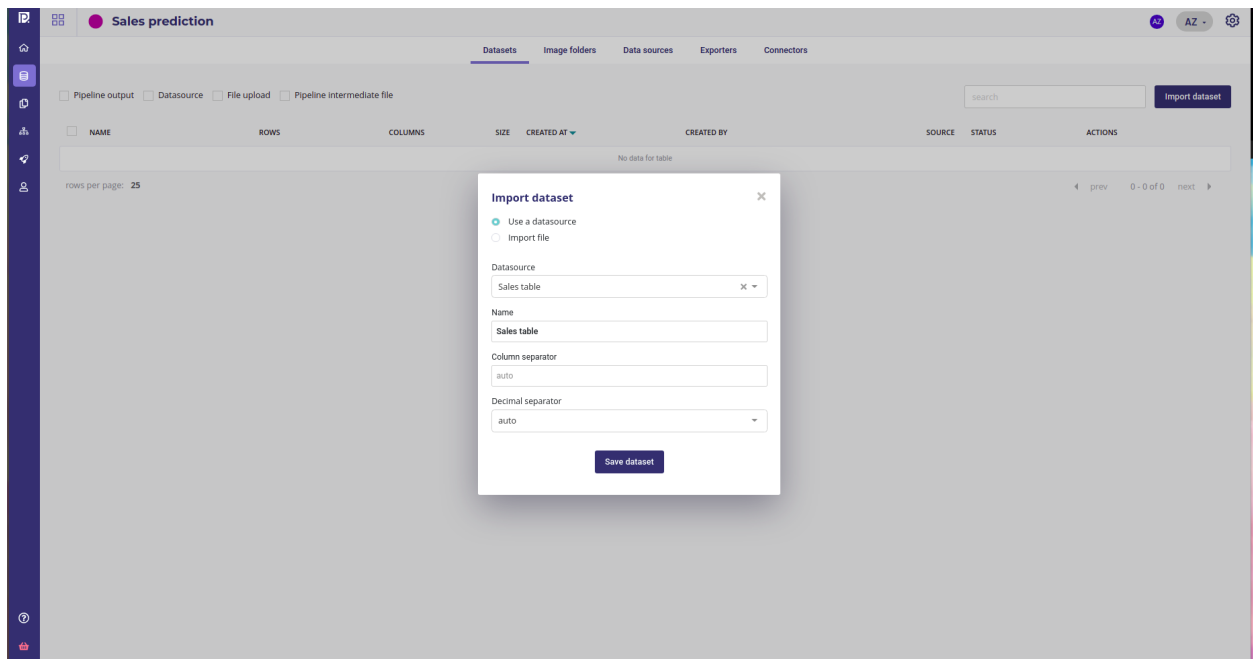


Fig. 62 – Import dataset

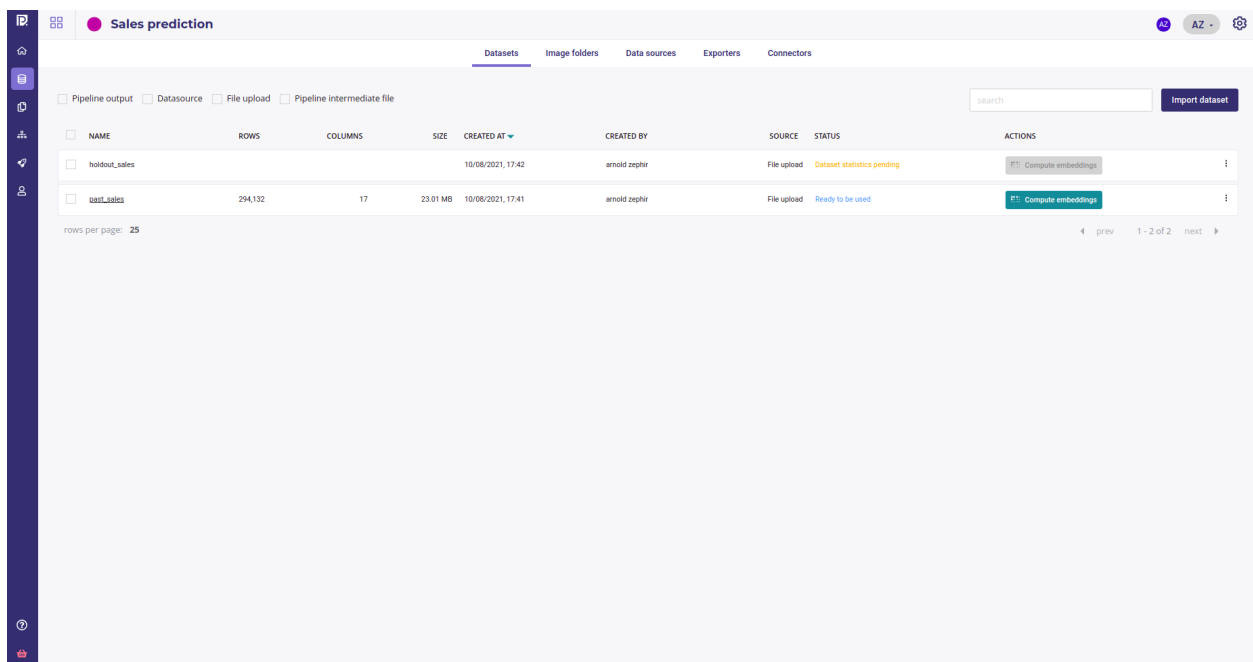


Fig. 63 – You got two datasets.

## Feature engineering

Feature engineering is the addition or transformation of one or more features to create new features from the original dataset. In Prevision Platform, and most of the modern tools, feature engineering are done with *components* and *pipelines* yet in most of case you don't need to add features as the AutoML engine makes all of the standard feature engineering by itself.

Here we are going to add a fold column on the date features in order to properly evaluate our model stability. A specific component has been developed by the datascience team *starting from the Prevision Boilerplate* and pushed on a private repo.

The component may now be integrated into the component library of the project.

Go to the pipelines section of your project and under the Pipeline Components tab, click **New pipeline Component**

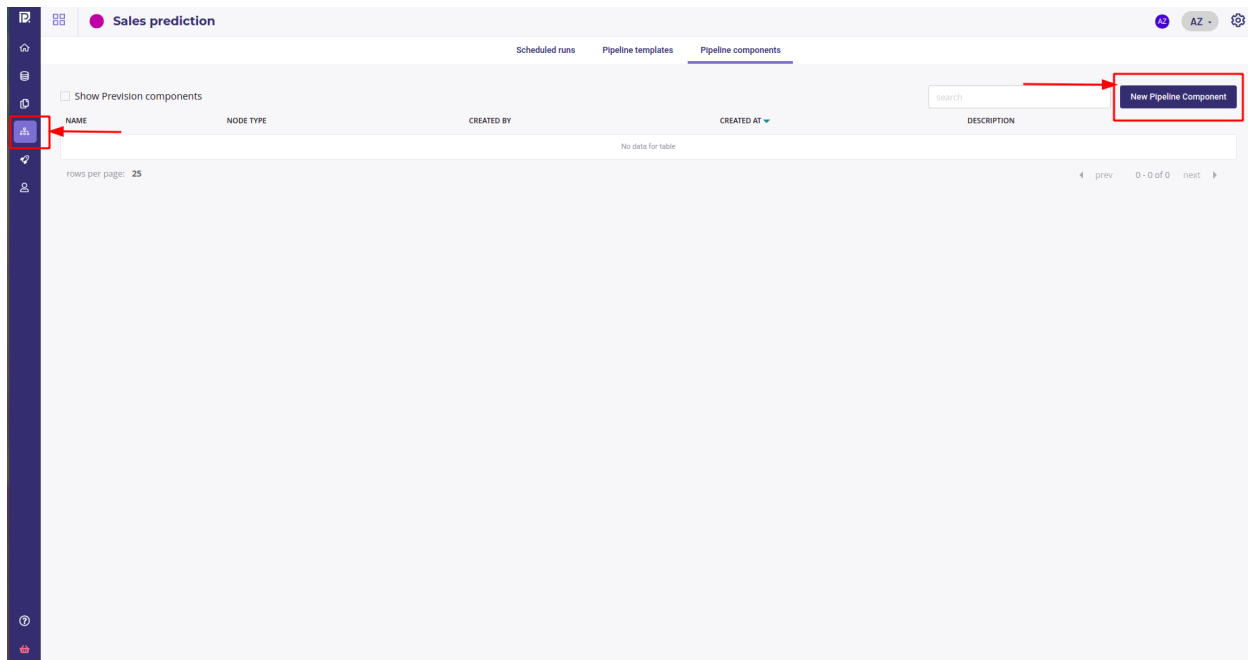


Fig. 64 – Create a new component

And select your repo and branch

Once the component is built, its status will be ok and we can use it in a *pipeline*. Create a new pipeline template with three nodes :

- an import dataset, to read the trainset
- the newly created component ( « build fold » )
- a **save dataset** node to save the feature engineered dataset into you Data

Then create a new *schedule run* that you gonna execute manually once on your trainset.

Once you did the configuration, select « Manual » as the trigger and run your Schedule run. In a few seconds, a new dataset should be available in your data section as a **pipeline output** with a new **fold** column

You now got a dataset with features for training model and an holdout to validate your models.

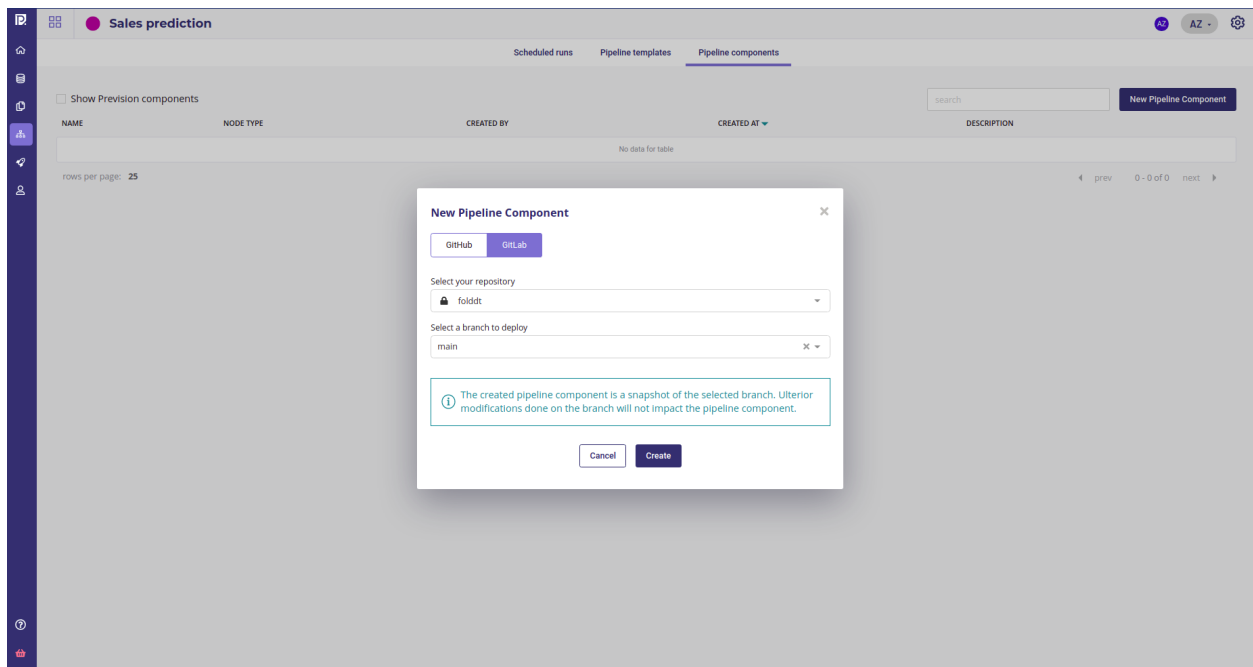


Fig. 65 – Import component from your repo

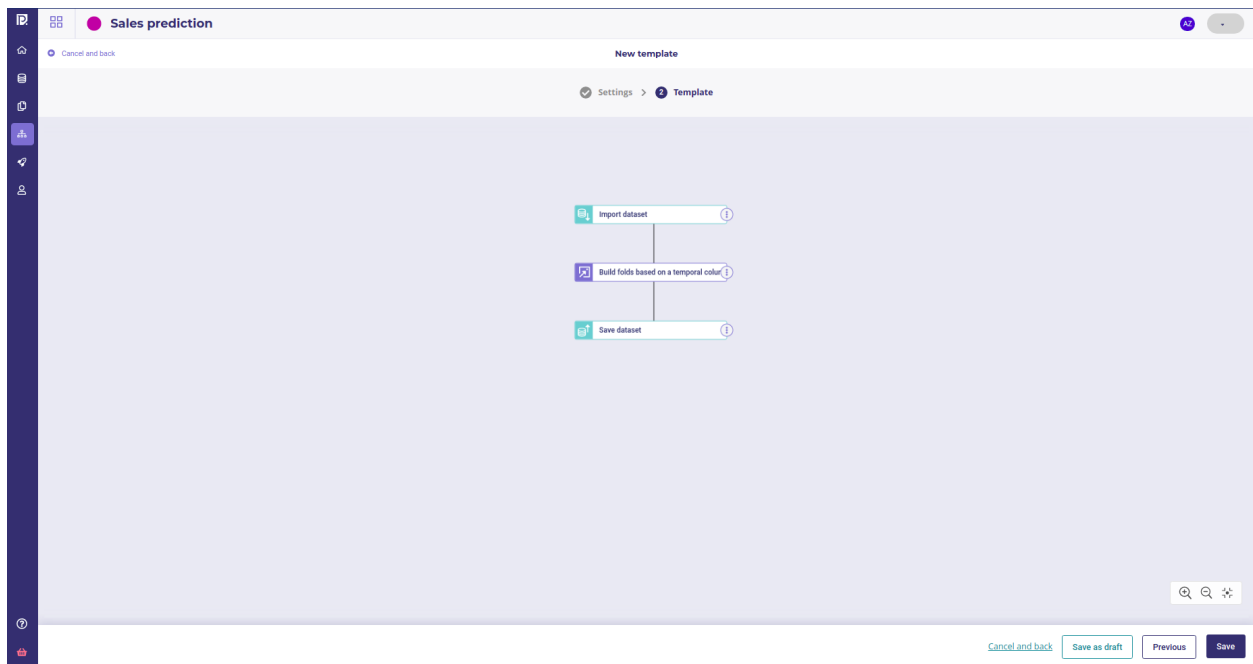


Fig. 66 – A simple feature engineering pipeline

The screenshot shows the 'New Scheduled run' form in the Prevision.io interface. The form is titled 'New Scheduled run' and has a breadcrumb trail: 'Settings > Node settings > Schedule'. The form contains three input fields: 'Name' with the value 'add fold to trainset' (20 / 40 characters), 'Description (optional)' (0 / 210 characters), and 'Template' with a dropdown menu showing 'build fold'. At the bottom right, there are three buttons: 'Cancel and back', 'Save as draft', and 'Next'.

Fig. 67 – Create a new scheduled run

The screenshot shows the 'Configuration 2/3' step in the Prevision.io interface. The form is titled 'Configuration 2/3' and has a breadcrumb trail: 'Settings > Node settings > Schedule'. The form contains a section for 'Import dataset' with the description 'This component will import a dataset, previously load into the dataset section of the project, as a pipeline input'. The 'Input dataset' dropdown menu shows 'past\_sales'. At the bottom right, there are three buttons: 'Cancel and back', 'Save as draft', and 'Previous'. The main area of the form shows a diagram with three components: 'Import dataset' (To configure), 'Build folds based on a temporal column' (Editable configuration), and 'Save dataset' (Editable configuration).

Fig. 68 – Set your trainset as the input dataset

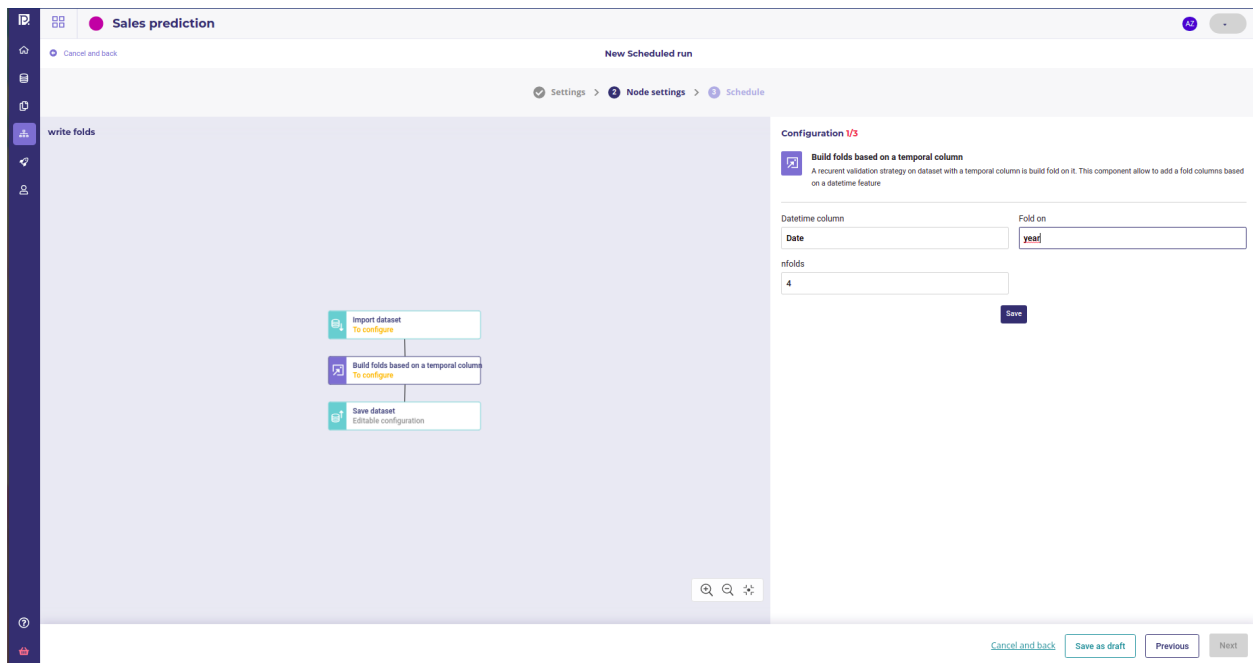


Fig. 69 – configure your fold component parameters

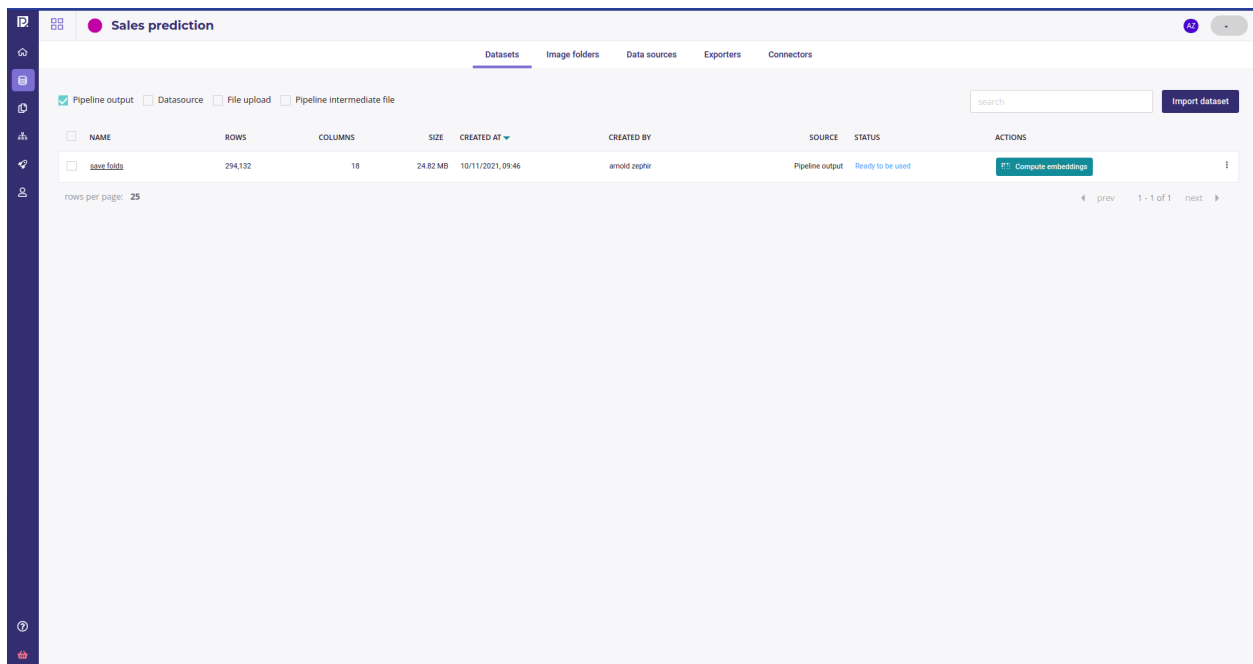


Fig. 70 – Pipeline output dataset

The screenshot shows the 'Columns' tab of a 'Sales prediction' pipeline. The table displays the following data:

Column name	Role	% of missing value
fold	Categorical	0.00%
Weekly_Sales	Linear	0.00%
Unemployment	Linear	0.00%
Type	Categorical	0.00%
Temperature	Linear	0.00%
Store	Linear	0.00%
Size	Categorical	0.00%
MarkDown5	Linear	91.84%
MarkDown4	Linear	92.85%
MarkDown3	Linear	92.14%
MarkDown2	Linear	93.15%
MarkDown1	Linear	91.94%

Fig. 71 – Pipeline output dataset

Tableau 8 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn

## Build your own

For the sake of this guide, we built a very basic feature engineering pipeline but you can add as many transformation as you want and build very complex pipeline.

Here we only have one component that adds a fold column, which is the year modulo 4. You can make the feature engineering on your local machine with the following code. Yet, if you want to build your own component you can [follow this guide](#) or some others

```

1 def addfold(df: pd.DataFrame, dtcol: str="dt", foldon:str="year", nfolds:int=3) -> pd.
  ↳ DataFrame:
2     if nfolds <=0 :
3         nfolds=3
4
5     df[dtcol] = pd.to_datetime(df[dtcol])
6     df["fold"] = df[dtcol].dt.month % nfolds
7
8     if foldon=="year" :
9         df["fold"] = df[dtcol].dt.year % nfolds

```

(suite sur la page suivante)

(suite de la page précédente)

```

10 if foldon=="day" :
11     df["fold"] = df[dtcol].dt.day % nfolds
12 if foldon=="hour" :
13     df["fold"] = df[dtcol].dt.hour % nfolds
14 return df

```

## Define the problem

This is the most important part and the one that should be allocated the most time.

In this step, you're going to define with the Line of business how to qualify the project as a success and you, as a datascientist, are gonna translate this as datascience metrics.

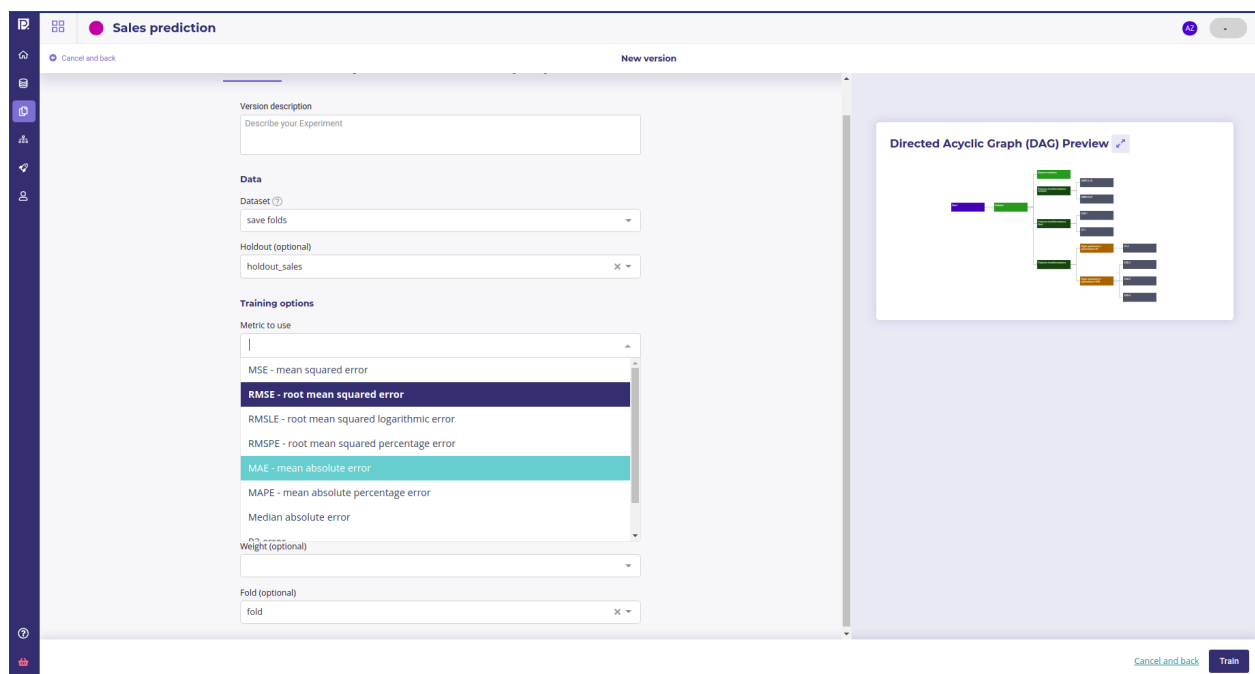


Fig. 72 – Regression metrics

Choosing the best metrics is out of the scope of this document but you must spend time with your business teams and ask this kind of question :

- Imagine that I have the perfect model, does it make me gain something ?
- how many money do I lost if I forecast 110 sales instead of 100 ?
- how many money do I lost if I forecast 90 sales instead of 100 ?
- are all the predicted product equal ?
- Should I forecast the total number of item sold, the total amount of sales (in € ), the total weight of my items or the total volume ?
- How much time before should I forecast ?
- etc ...

As a datascientist, by using an AutoML platform, your role is not to code in python or make dockerfile but to transcribe business problems to Machine Learning parameters.

However, in Prevision Platform, you can build what is called **experiment** to help refine your objectives



Tableau 9 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week

## Experiment

An experiment is a set of Model Building with slightly different parameters across version and a common Target on each version. On each experiment, many models will be automatically built evaluated in cross-validation and on the holdout dataset if you provide some.

In our case, the models will be trained on our engineered dataset **with a fold column** and evaluated on an holdout provided by IT Team.

It is very important to have a good validation strategy to guarantee that the model built in the experiment phase will stay stable on production. Here we choose to :

- build a fold column on the modulo of the year number so that we stay confident that the model learned some trends that stay stable over the year
- Validate on an holdout with sales from a year that was not in the trainset

Hence, if the holdout score is near to the cross validation score, we know that our model is going to stay good when launched in production and shared across the company

For creating a new experiment, go to the **experiments** section of your project and click **new experiment**. You could choose to import some *external models* if you have some but here we are using the AutoML Prevision Engine. As we want to forecast sales, choose « Tabular » and « Regression ». Give a name to your experiment and click « Create experiment »

When you create a new experiment, there is no version of the experiment existing so you will be prompted to create a new version. The next screen is where you setup all your experiment parameters :

- The train dataset : use the output of the Schedule run from the step 2 with engineered features
- The holdout dataset : use a dataset with same target than trainset but with data that **are not** in the trainset
- The metric : use the fittest metrics that solves the business objectives defined in the step 3. You can change it on each version of your experiment so run as much version as you need if you are not sure
- set your target ( here we choose « Weekly Sales » )
- and set the fold column up, using the column built during the feature engineering phase.

Note that you may go the models and feature engineering tabs to change some automl configuration but in most of case the default configuration is fine.

Once done, click on **train** to launch the training. The platform will immediately start to build and select models with the best hyper parameters. The models will stacks in the « models » tabs of your experiment :

Note that you can launch another version of your experiment as soon as you want, for testing other metrics for example, by using the new version button in the top right corner.

If you have several version , the experiment dashboard will always display the last version but you can change to another version with the version dropdown menu or the **versions list** tabs

**Sales prediction**

**New experiment**

**Name**

Experiment name

**Data type**

Tabular Timeseries Images

**Training type**

Regression Classification Multi-classification Text similarity

**AUTOML**

The Prevision.io AutoML engine allows you to quickly benchmark and optimize a range of open source algorithms to get highly performant models.

**What do I need?**

You just need data, imported into Prevision.io as a Dataset. Your dataset just needs to contain a target column (and a temporal one, if you are working with time series), and you are good to go.

**What's next?**

Once the experiment and the models are created, you can analyze, version, and deploy them to create a prediction API endpoint, or use it with pipelines to schedule batch predictions.

[Cancel and back](#) [Create Experiment](#)

Fig. 73 – Setting the experiment up

**Sales prediction**

**New version**

**Version description**

Describe your Experiment

**Data**

Dataset save folds

Holdout (optional) holdout\_sales

**Training options**

Metric to use RMSE - root mean squared error

Performances ☒ QUICK ☐ NORMAL ☐ ADVANCED

**Fields configuration**

Target column Weekly\_Sales

ID column (optional)

Weight (optional)

Fold (optional) fold

**Directed Acyclic Graph (DAG) Preview**

[Cancel and back](#) [Train](#)

Fig. 74 – Experiment parameters

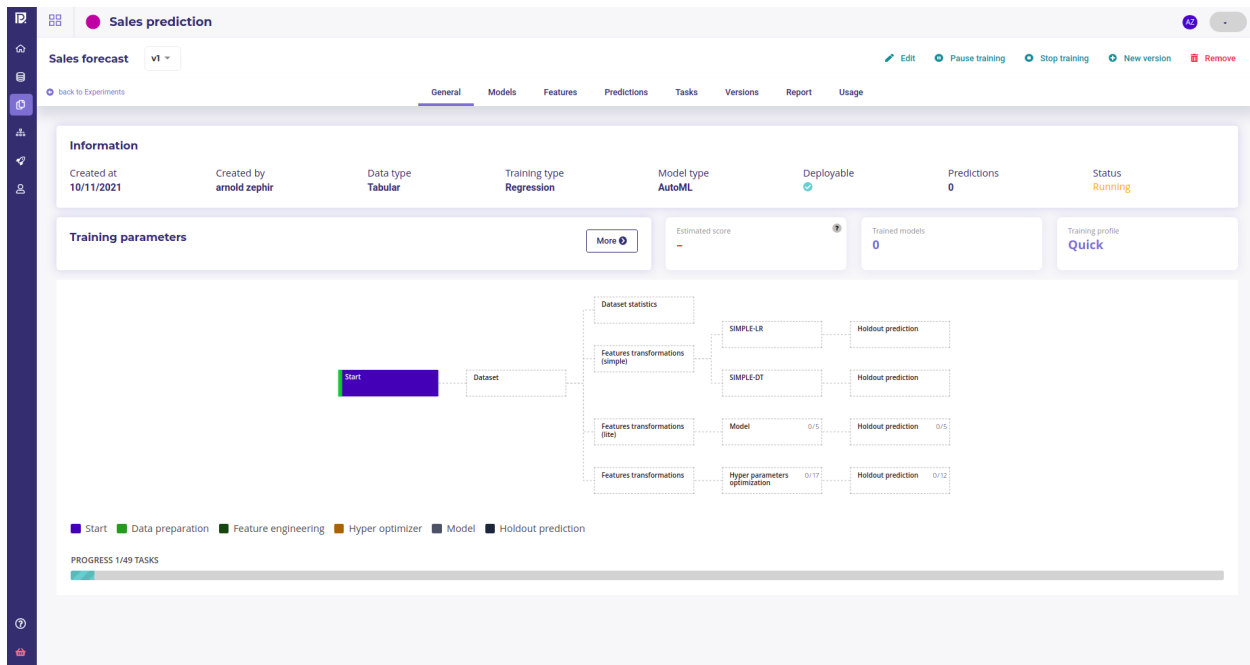


Fig. 75 – The experiment dashboard

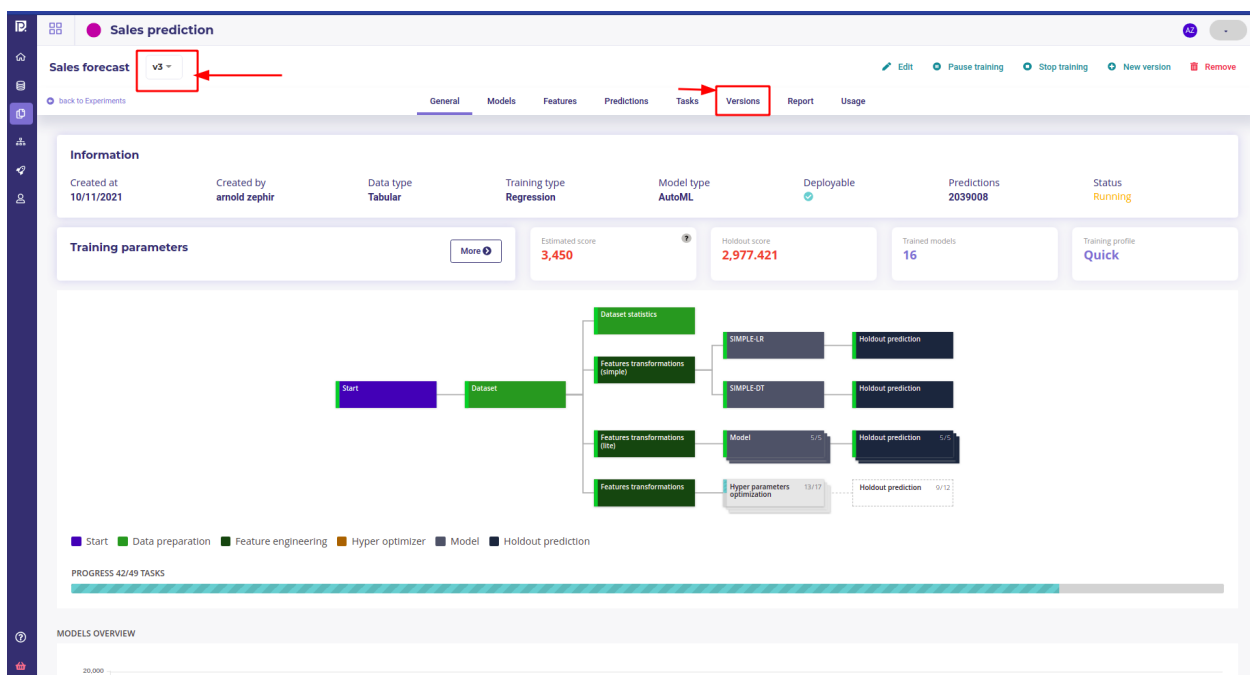


Fig. 76 – The experiment dashboard

You can launch as much version as you want and they will run in parallel. You can now grab a coffee and wait till models are built ! Depending on the size of your dataset and the plan you subscribed, expect to wait from 10mn to 2 hours before having enough models to evaluate your experiment. In our case, we got our model in ~20mn

Tableau 10 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week
Experiment	Done	~100 models	~20mn

## Evaluate

After a few minutes, you should have between 15 and 40 models for each version, depending on your option

VERSION	DESCRIPTION	CREATED AT	CREATED BY	SCORE	MODELS	PREDICTIONS	STATUS
V3		10/11/2021, 10:12	arnold zegher	2,450 (mae) ★★☆☆	17	2,166,446	🟢
V2		10/11/2021, 10:12	arnold zegher	429.8 (mape) ☆☆☆	17	2,166,446	🟢
V1		10/11/2021, 10:03	arnold zegher	8,943 (mse) ★☆☆	19	2,421,322	🟢

rows per page: 25

Fig. 77 – List of experiment versions.

This step is all about evaluating all the models produced and select 2 to 4 models to deploy for testing model in real conditions.

First, have a quick look at the list of versions below ( tab **versions** of your experiment ). There is a small 3-star evaluation that gives you informations about versions quality. In that case, the Version 3, that has been trained on Mean Absolute Error, looks the most promising. Click on the version to enter the version dashboard for deepest analysis.

On the Version dashboard, you got several indicator but the most important is the models comparator :

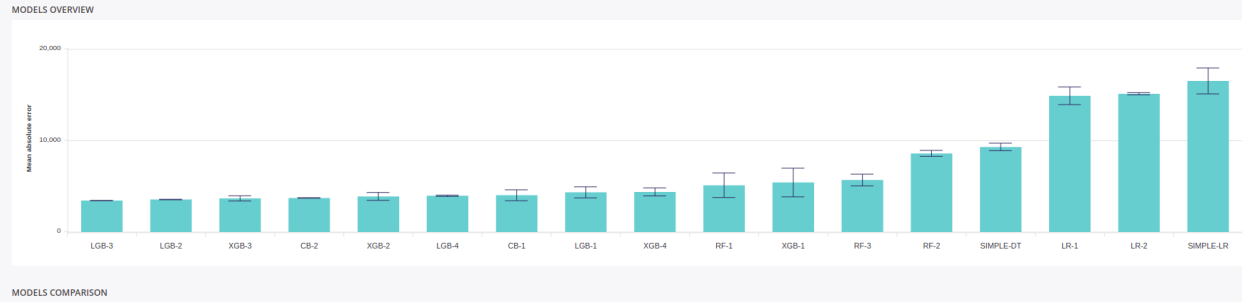


Fig. 78 – List of models of a version

You can quickly see :

- performance of each model done , evaluated on the metrics you choose for this version.
- stability of each model ( represented with a small error bar ) computed on a cross validation of the trainset **using the fold column provided**

### Dumb models

Prevision Platform always produces what we sometimes call « dumb models », a linear regression and a Random Forest of only 5 depth, called simple-LR and simple DT. It is always a good idea to watch performance of this models against the most complexe one and ask yourself if using them could be good enough for your problem.

Indeed, as they are very simple :

- they can be implement in sql ( auto-generated code is even provided on the model analysis page )
- they often are more explainable and are more accepted from the Business teams, are they are easier to understand and use.

As a datascientist, accepting to use a simple if-else instead of complex Blend of Gradient Boosting if it solves the issue is your responsibility too !

On the experiment above, the :

- LGB-3
- XGB-4
- CB-2

Looks promising so we are going to have a closer look. Click on the model barplot to enter the detailed model analysis, CB-2 for example.

Here you got more detail about the models you select, like various metrics and the actual vs predicted Scatterplot

You can download the Cross validation file if you want to run your own evaluation. The CB2 is quite good but if we look at the Scatterplot, we see that performance fall in the range from 40k to 80k. If we go to the LGB-3 page, we see a more stable performance

Evaluating a model is out of the scope of this guide but be aware that it is another step where you **MUST** involve your business team and explain each metrics and chart to them so you choose together the model that solves their problem the best.

The model analysis page is full of metrics to parse and you can run as much experiment as you want in order to find the model that fit the business problem the best.

After discussions with the LoB , we decided to keep the LGB-3 and the XGB-4, one because it performs well and the others because its performance are stable when evaluated on the holdout.

In order to refine this, we are now going to deploy both models and see how they perform in real world.

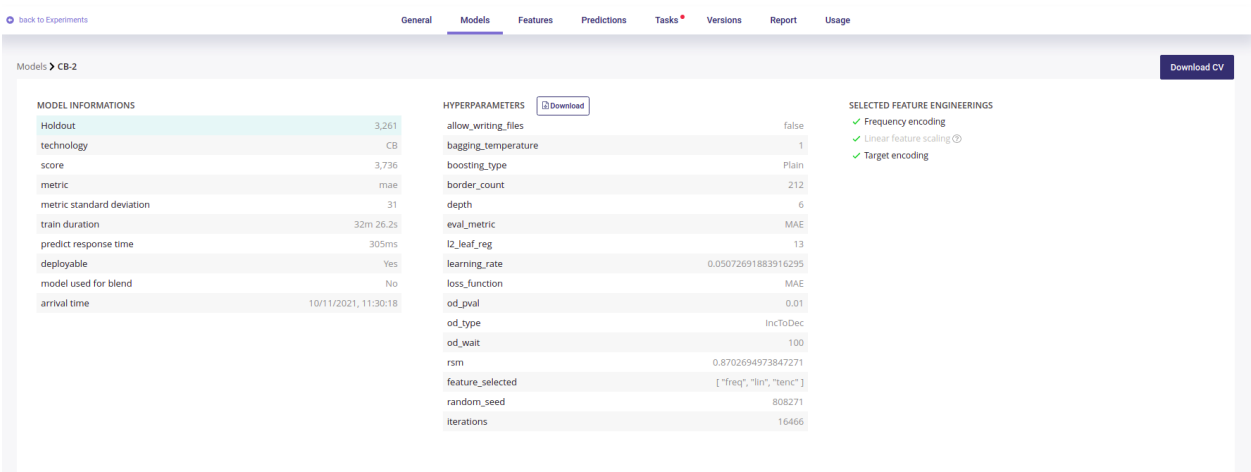


Fig. 79 – All the metrics of the model

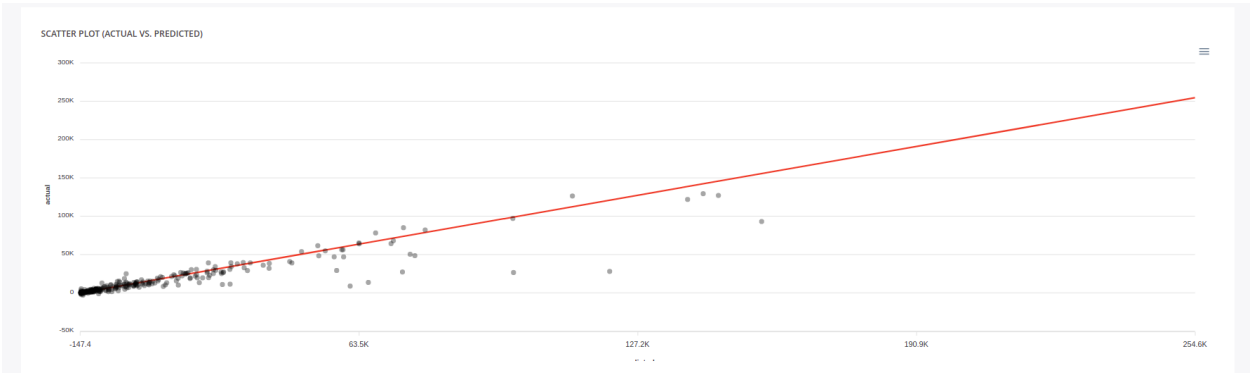


Fig. 80 – Predicted vs actual

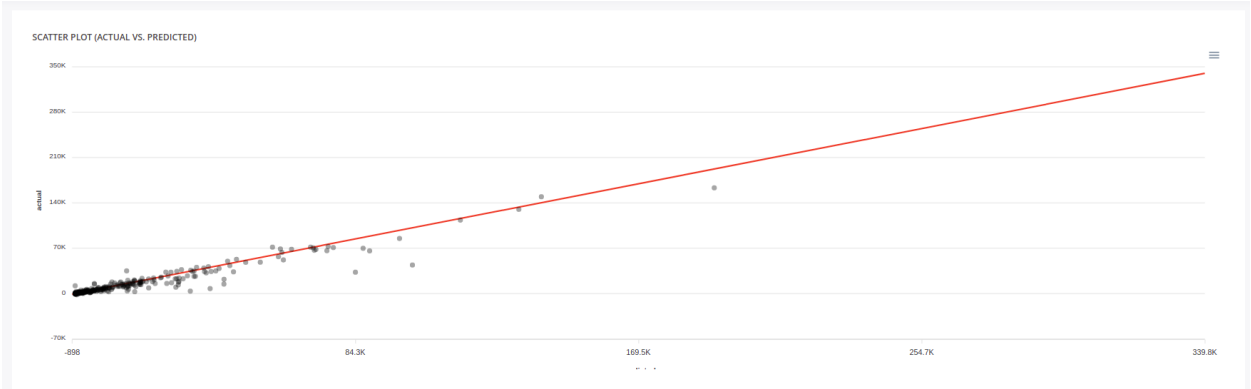


Fig. 81 – Predicted vs actual ( LGB-3 )

Tableau 11 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week
Experiment	Done	~100 models	~20mn
Evaluate	Done	2 models selected with business team	A week

## Deploy

In this step two models will be deployed in order to test them on real data and usage. While deployed, their performance will be closely monitored for deciding if they are good for production grade utilisation.

Go to the « Deployment » section of your project and click on **deploy a new experiment**. Select LGB-3 as main model and XGB-4 as a challenger in order to see which one performs best on real data.

The screenshot shows the 'Deploy a new experiment' form in the Prevision.io interface. The form is titled 'Sales prediction' and has tabs for 'Deployments experiments' and 'Deployments applications'. The 'Deploy a new experiment' section includes a 'Deployment name' field with the value 'Sales forecasting run', a 'Description (optional)' field, and a URL field with the value 'https://sales-forecasting-run.int.prevision.io/'. Below these are two rows of model selection. The first row has 'Select an experiment' set to 'Sales forecast', 'Select a version' set to '3', and 'Select a tagged model to deploy' set to 'LGB-3'. The second row has 'Select an experiment' set to 'Sales forecast', 'Select a version' set to '3', and 'Select a tagged model to deploy' set to 'XGB-4'. At the bottom, there is an 'ACCESS RIGHTS' section with three radio buttons: 'Public', 'Instance collaborators' (which is selected), and 'Project collaborators'.

Fig. 82 – Set your main and challenger

The Main model will be used for prediction but each time you call it, a prediction will be done with the challenger model too and chart will be generated so you can compare them.

Wait a few minutes to get :

- a standalone webapp for human user to test ( « Application link » url )
- a batch predictor available for scheduling prediction
- a REST API for calling the model from others software ( « Documentation API » link )

**Deploy a new experiment**

Deployment name: Sales forecasting run

Description (optional):

<https://sales-forecasting-run.int.prevision.io/>

Select an experiment: Sales forecast X

Select a version: 3 X

Select a tagged model to deploy: LGB-3 X

Select a challenger model (Optional):

Select an experiment: Sales forecast X

Select a version: 3 X

Select a tagged model to deploy: XGB-4 X

**ACCESS RIGHTS**

☐ Public ☒ Instance collaborators ☐ Project collaborators

**Deploy**

Fig. 83 – Set your main and challenger

That's all. Your model can now be called from any client of your company and all its requests will be logged for further monitoring. Yet, in order to send prediction each week to the sales team, you need to schedule them.

Tableau 12 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week
Experiment	Done	~100 models	~20mn
Evaluate	Done	2 models selected with busines team	A week
Deploy	Done	Model available accross the organisation	5mn

## Schedule

Once any model is deployed, it can be used to schedule prediction. First step is to insert it into a pipeline template and then create a new Schedule using this template.

Note that you need helps from your IT team in this step, in order to define the name of the table where you will read the features from each week. You can use the same table that will be overwritten each week, for example « sales to predict » to read and « Sales predicted » to write, or a more complexe naming scheme.

First you need two create two new assets :



- a new datasource that gonna link to the Table were the IT team is going to put the features for prediction each week
- a new exporter to push the result

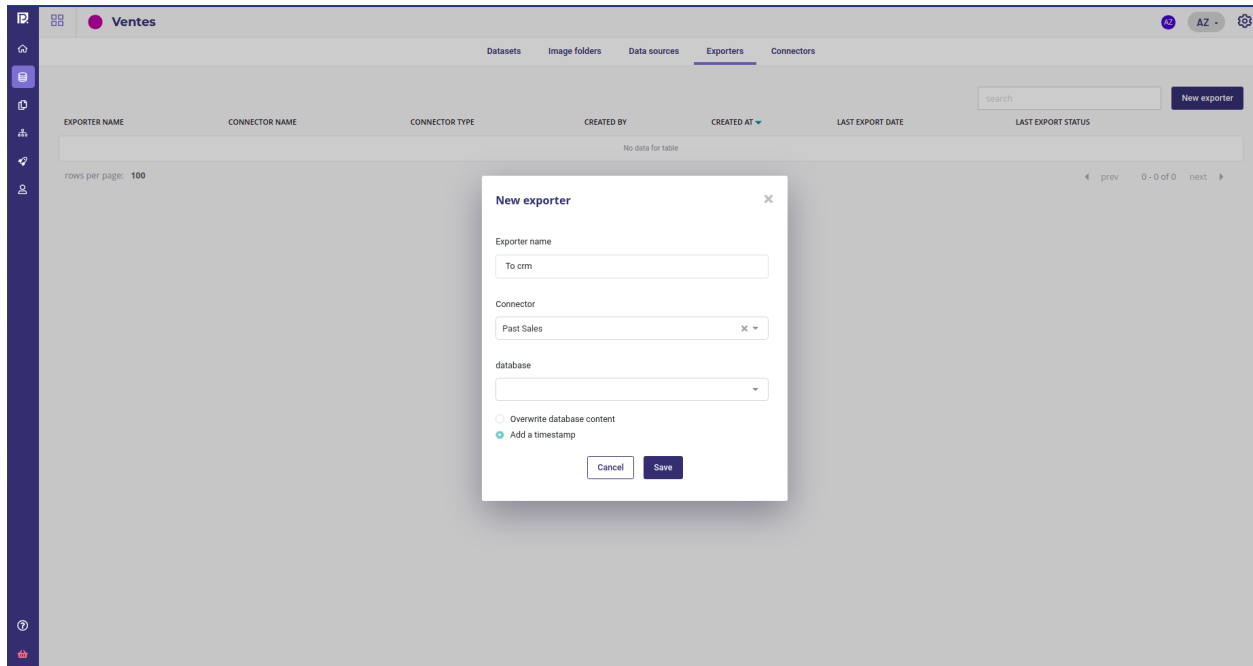


Fig. 84 – Create an exporter to push data to your crm

So you can use them in a new pipeline template with 3 nodes again :

- Import from the datasource, where the datasource is the table with all the weekly features
- a deployment predict regression node
- an export dataset node, that use the exporter above

Once you got your template, create a new Schedule based on it

And choose the Name of your deployment as the experiment deployment Id

And then, instead of the manual Trigger, use a periodic one , putting the configuration that fits your need the best ( here, a weekly prediction each Monday at 7 :00 AM )

Click run and wait few seconds. Your Prediction is now scheduled to run every Monday , from the table of « sales to predict » to the « Sales predicted » table of your databases.

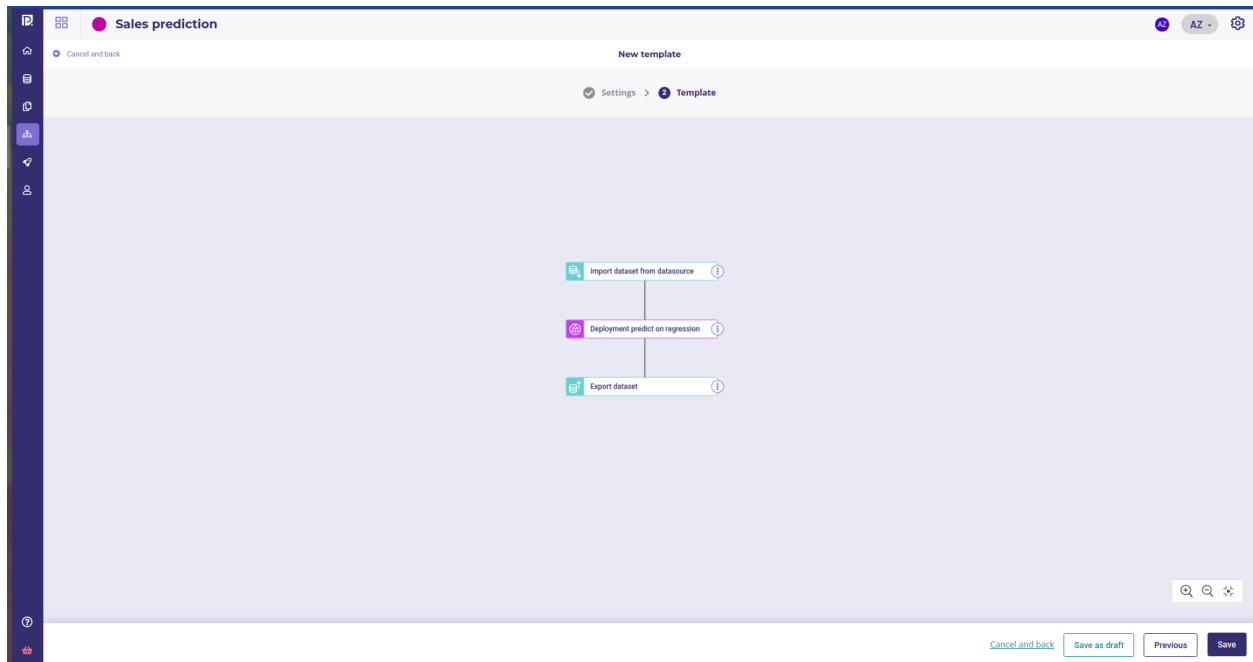


Fig. 85 – Template

Tableau 13 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week
Experiment	Done	~100 models	~20mn
Evaluate	Done	2 models selected with busines team	A week
Deploy	Done	Model available accross the organisation	5mn
Schedule	Done	Prediction in CRM each Monday at 09 :00	20mn

## Monitoring

Once a model is deployed, each call to it will be logged, being unit one or scheduled batch. You can track your model into the Deployments section of your project by clicking on a deployed experiment name in the list of experiments to access the deployment dashboard :

You can watch the features distribution of the trainset compared to the feature distribution seen in production and check the drift. Target distribution of the Main Model and Challenger model are shown side-by-side with those of the production in order to evaluate performance in a real application.

Under the monitoring/usage tab sit some SLA statistics about number of call average response time and errors.

By tracking all this indicators for a month or more, you can evaluate how does your model lives in production and

---

New Scheduled run

1 Settings > 2 Node settings > 3 Schedule

Name

0 / 40 characters

Description (optional)

0 / 210 characters

Template

write folds

build fold

Weekly Sales Forecast

Fig. 86 – Use your template in a schedule run

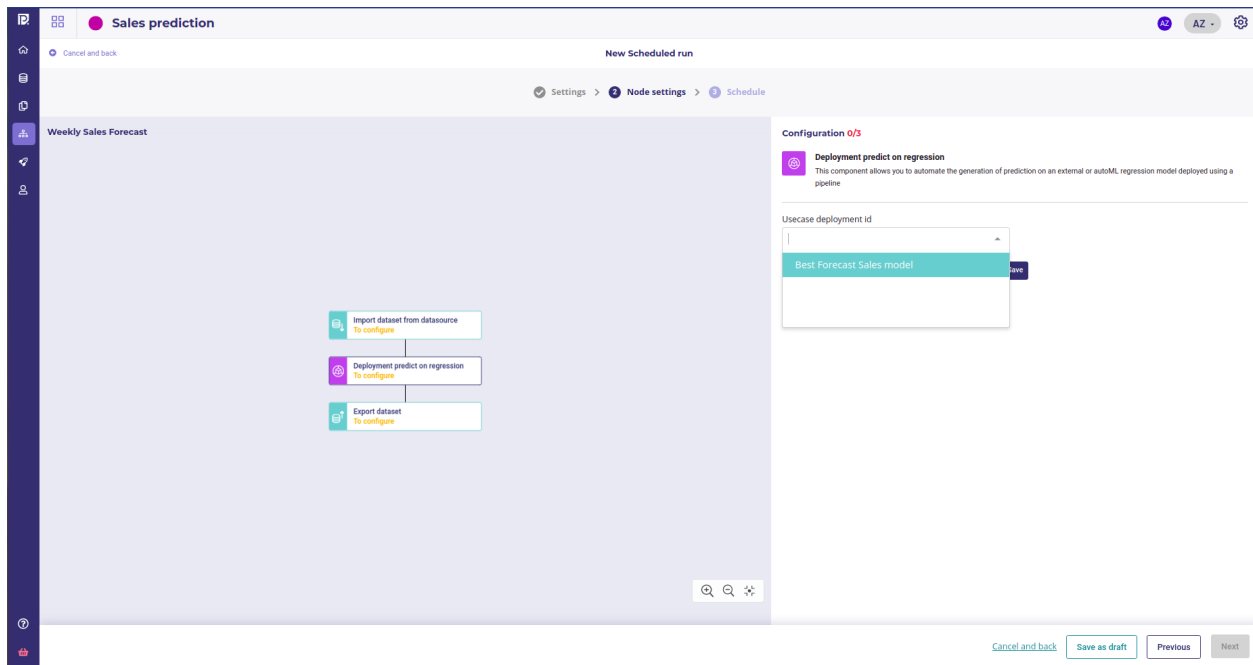


Fig. 87 – Use your template in a schedule run

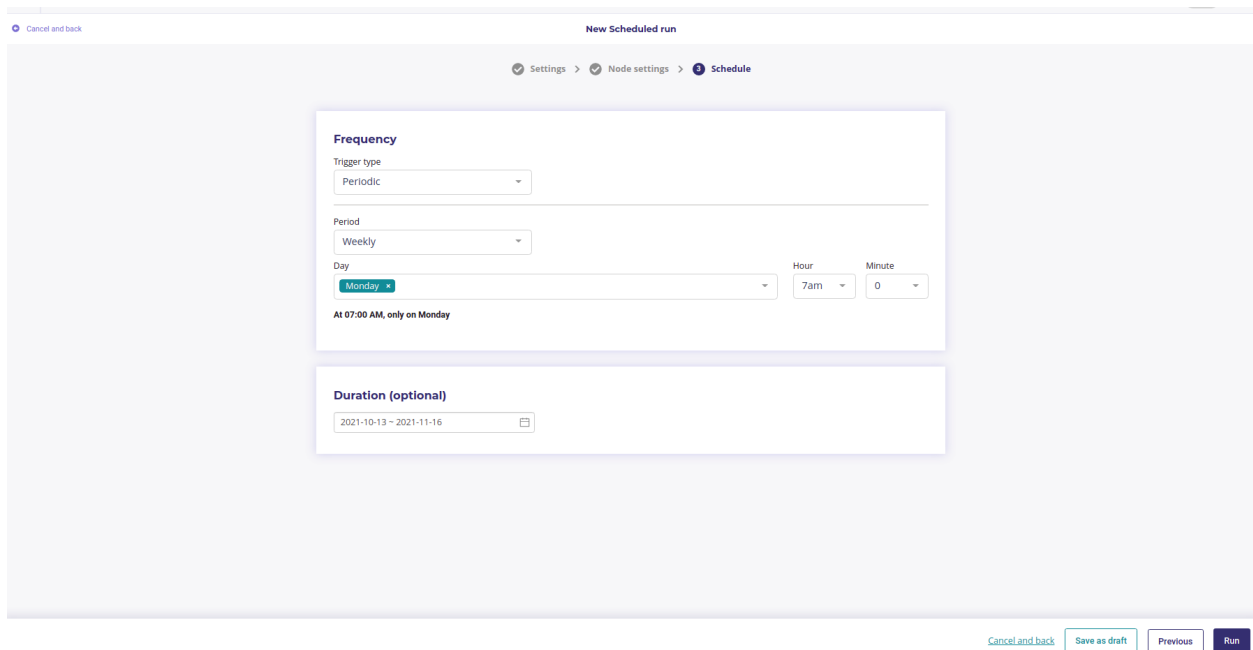


Fig. 88 – Scheduling a prediction each monday Morning

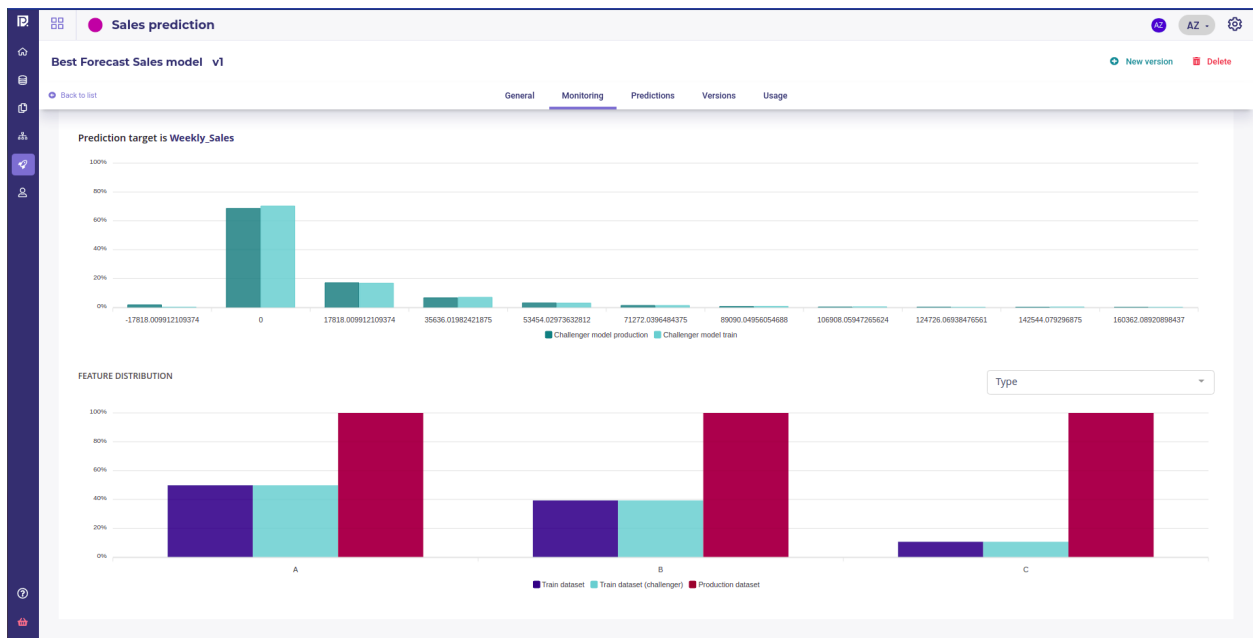


Fig. 89 – Train and production distribution

check that it behaves the way you expected while evaluating it in the experiment step.

Tableau 14 – Status

Task	Status	Output	Time spent
Data acquisition	Done	one trainset, one holdout	5mn
Feature engineering	Done	one engineered dataset with features, one holdout	20mn
Define the problem	Done	a metric to validate the models	A week
Experiment	Done	~100 models	~20mn
Evaluate	Done	2 models selected with busines team	A week
Deploy	Done	Model available accross the organisation	5mn
Schedule	Done	Prediction in CRM each Monday at 09 :00	20mn
Monitor	Done	Prediction in CRM each Monday at 09 :00	A month

## Conclusion

In this guide, you saw how to complete the whole datascience process in less than a morning and went from data to fully deployed model, shared accross the company with full monitoring.

Using a tool to solve the technical issue of the datascience, like finding the best model, deploy a model or import the data, allow to spend more time on what trully matters : talk with the Line of Business team to translate their problem to datascience configuration and metrics.