Implementing a credit scoring dashboard

JULY 2022

NAJWA MOULINE

Presentation Outline

1. Objectives
2. Dataset Preparation
3. Modeling
4. Dashboard

Objectives

Context

- French consumer credit company for people with little or no loan history
- Wish to develop a suitable scoring model that predicts the probability of customer default.

Business Problem

- Implementation of a customer scoring model and an interactive dashboard for customer relationship managers to:
  - maximise the bank's gain
  - provide transparency to customers on credit granting decisions.

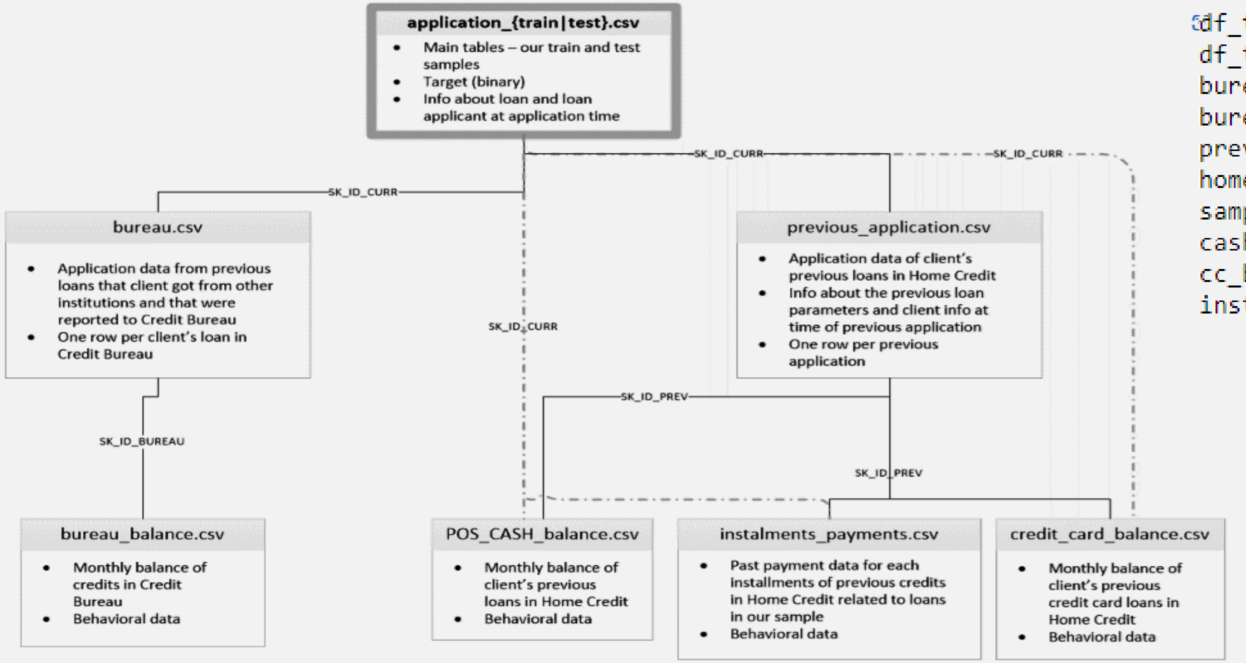
Mission

- Develop a scoring model of the customer's probability of default (with little or no loan history)
- Develop an interactive dashboard that provides the interpretations of the predictions made by the model

Dataset Preparation

Dataset

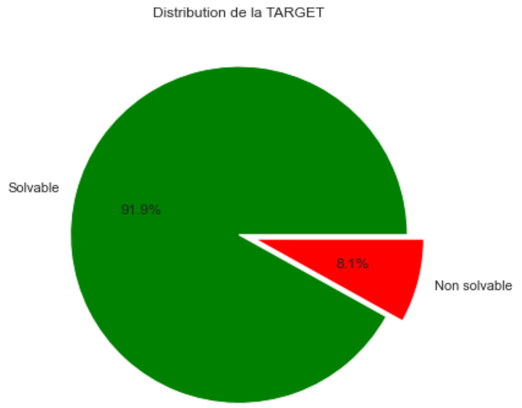
Dataset analysis




```
df_test : (48744, 121)
df_train : (307511, 122)
bureau : (1716428, 17)
bureau_balance : (27299925, 3)
previous_app : (1670214, 37)
homecredit_desc : (219, 5)
sampl_submission : (48744, 2)
cash_balance : (10001358, 8)
cc_balance : (3840312, 23)
inst_payments : (13605401, 8)
```

Observations

- A large dataset of 9 files containing personal and financial information of customers
- No duplicates
- Primary dataset – application_train :
    - 307511 clients, 121 variables
    - 24% NaN
    - Target imbalance (TARGET) - 92% solvent (0) vs. 8% insolvent
    -



Dataset Preparation

Exploratory Analysis and Feature Engineering

Creation of Datasets to allow the modeling of the problem

Simple dataset (250518, 10)

- Source file: application_train
- Operations: Deletion of columns with more than 10% NaN, deletion of categorical variables, elimination of outliers, creation of business features (ratios) and identification of best features with VIF (variance inflation factor)

Complex dataset (356251, 798)

- Source files: all files
- Operations - entirely inspired by the Kaggle kernel available on https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script :
  - Detection of outliers / anomalies
  - Imputation of missing values
  - One-hot encoding for categorical variables
  - Tables joined by the SK_ID_CURR key
  - Creation of business features (ratios)
  - Creating features from aggregations - min, max, mean and var

Modeling

Balancing TARGET classes

Data Balancing - Approaches

Approaches

- The significant imbalance of the TARGET leads to a high "accuracy" of the Dummy model.
- To overcome this imbalance, 3 options are considered:
  - RandomUnderSampler: Removing Observations from the Majority Class
  - SMOTE: repetition of the observations of the minority class
  - Class_weight="balanced": argument that indicates the imbalance to the algorithm so that it takes it into account directly.

Impact of balancing

estimator	sampler_type	params	mean_test_score	mean_fit_time	accuracy
DummyClassifier(strategy='most_frequent')	no_sampler	{'scaler': MinMaxScaler()}	0.919514	5.682633	0.924
LogisticRegression()	no_sampler	{'scaler': MinMaxScaler()}	0.919269	16.635409	0.924
LogisticRegression(class_weight='balanced')	class_balanced	{'scaler': MinMaxScaler()}	0.755791	16.490391	0.712
LogisticRegression()	param_sampler	{'sampler': SMOTE(random_state=14), 'scaler': MinMaxScaler()}	0.754411	25.463076	0.718
LogisticRegression()	param_sampler	{'sampler': RandomUnderSampler(random_state=14), 'scaler': MinMaxScaler()}	0.749232	7.317816	0.718

Modeling

Metrics and Algorithms

Choice of metrics and algorithms

Metrics

- Considering the confusion matrix, the idea is to limit FN and FP. FN being more expensive.

		Predicted	
		Positive (1)	Negative (0)
True	Positive (1) – NS	TP	FN
	Negative (0) - S	FP	TN

Five metrics are considered:

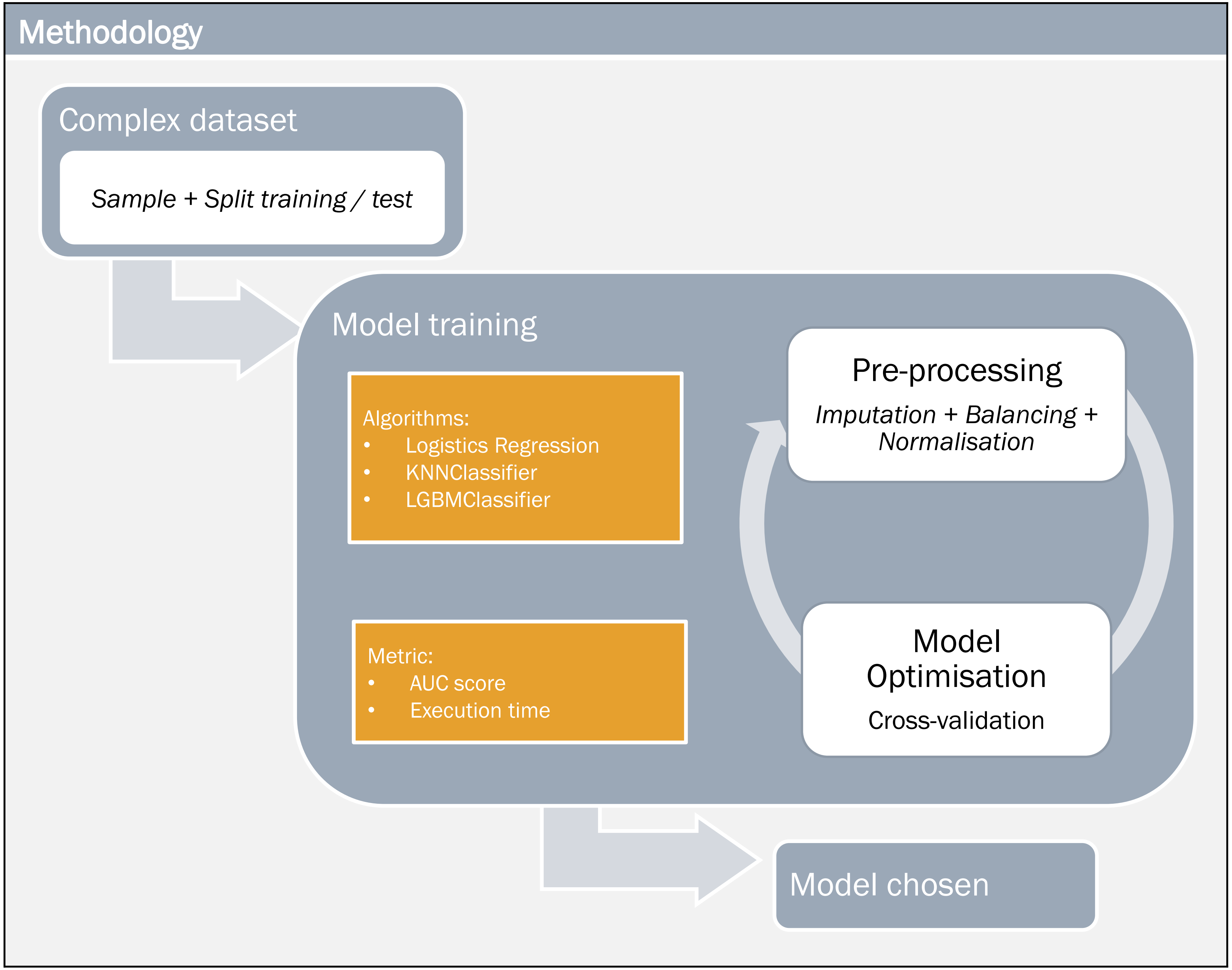
- Accuracy – excluded due to initial class imbalance
- Recall – to be maximised to minimise Type II Error (FN)
- Precision – to be maximised to minimise Type I (FP) Error
- F1 Score – considers both measures Recall and Precision but will be low if one of the measures improves at the expense of the other.
- AUC/ROC (preferred) – the more the model can predict classes, the higher the AUC.

Algorithms

- Three classification algorithms of different families are compared:
  - Logistic regression (a linear algorithm)
  - KNeighborsClassifier (an ensemblist algorithm)
  - Light Gradient Boosting Machine (a gradient boosting algorithm)

Modeling

Methodology



Modeling

Best Model

Best Model

Parameters optimisation

For each model, we will test the different parameters of the pipeline :

- Imputation of missing values
- Feature scaling – StandardScaler, RobustScaler and MinMaxScaler
- Balancing data with the 3 different methods mentioned earlier

- Results are classified in order of AUC value and execution time.

Result of the first 5 models

estimator	sampler_type	params	mean_test_score	mean_fit_time	accuracy	precision	recall	f1score	auc
LGBMClassifier()	param_sampler	{'sampler': RandomUnderSampler(random_state=14), 'scaler': RobustScaler()}	0.752936	14.343164	0.923	0.413	0.041	0.075	0.742
LGBMClassifier()	param_sampler	{'sampler': RandomUnderSampler(random_state=14), 'scaler': MinMaxScaler()}	0.752340	14.495963	0.923	0.413	0.041	0.075	0.742
LGBMClassifier()	param_sampler	{'sampler': RandomUnderSampler(random_state=14), 'scaler': StandardScaler()}	0.750664	15.301300	0.923	0.413	0.041	0.075	0.742
LGBMClassifier()	param_sampler	{'sampler': SMOTE(random_state=14), 'scaler': RobustScaler()}	0.762773	32.593179	0.923	0.413	0.041	0.075	0.742
LGBMClassifier()	param_sampler	{'sampler': SMOTE(random_state=14), 'scaler': MinMaxScaler()}	0.762897	44.197593	0.923	0.413	0.041	0.075	0.742

Modeling

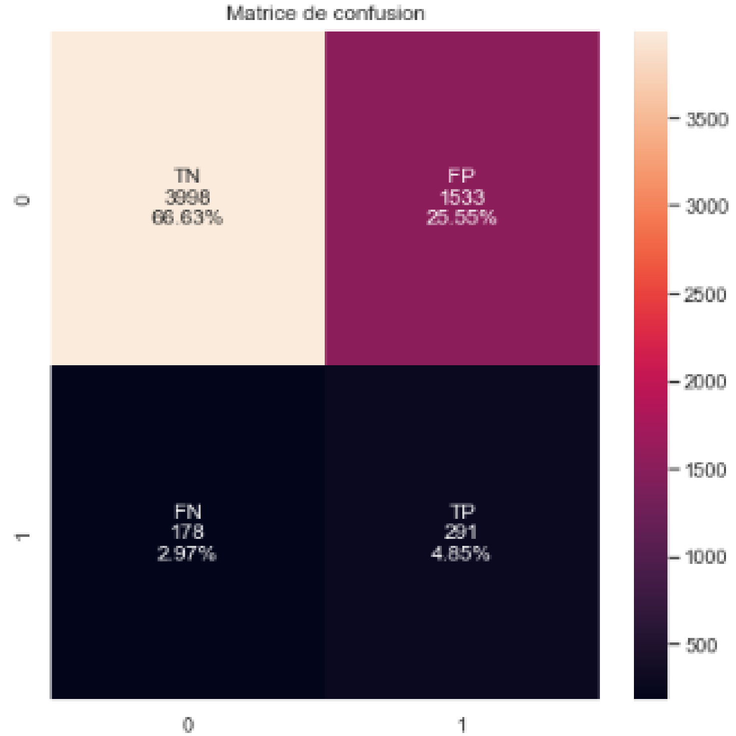
Hyperparameters Model Optimisation

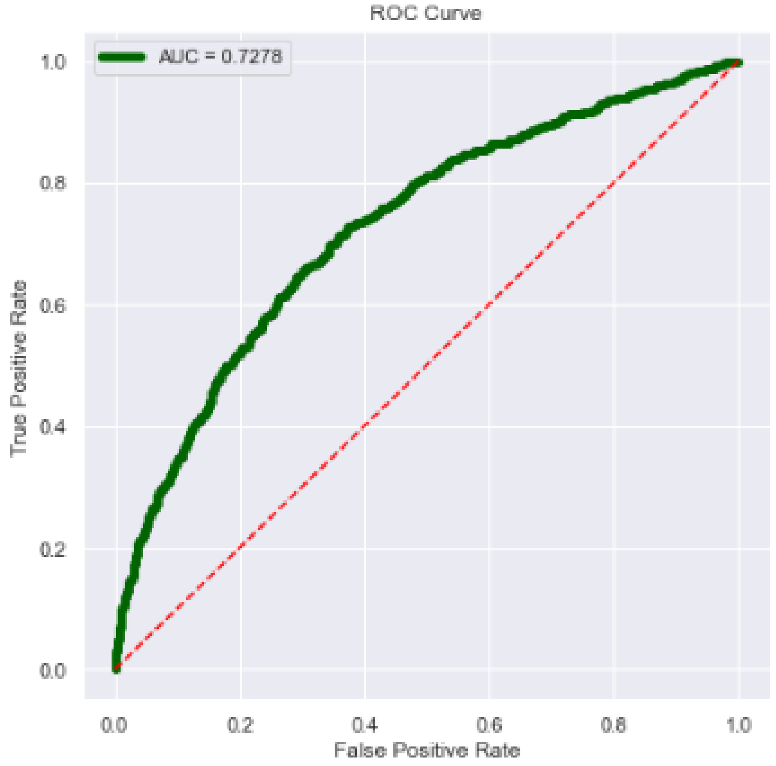
Hyperparameters Model Optimisation

Best Model

The best model was obtained with :

- Standardisation - RobustScaler
- Balancing - RandomUnderSampling
- Algorithm - Light Gradient Boosting Machine Classifier (an optimization of the hyperparameters of this algorithm had little impact on the AUC)





Modeling

Gain optimisation

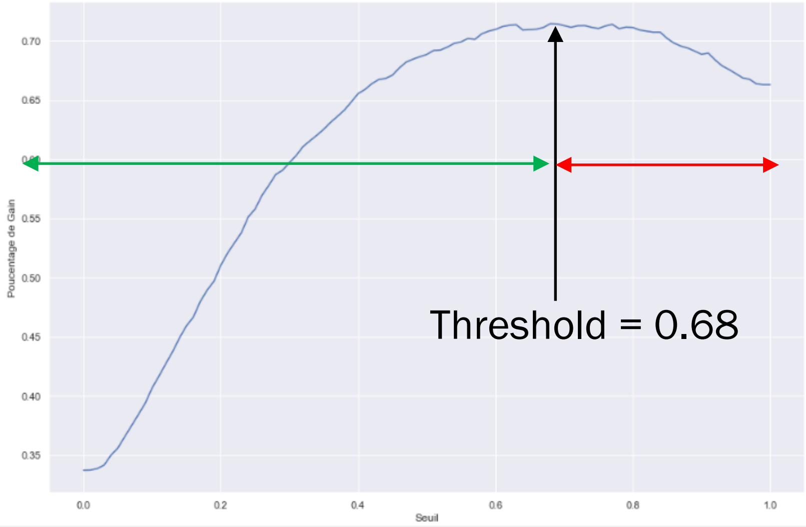
Gain Optimisation

Gain function

- The company seeks to maximize its gain.
- The gain function is : f(g) = (gain -gmin) / (gmax – gmin) où
    - gain = FN * FNcost + TN * TNprofit (cost/profit = 0% when predicted positive)
    - gmin = (TP + FN) * FNcost (when true positive)
    - gmax = (FP + TN) * TNprofit (when true negative)

		Prédicted	
		Positive (1)	Negative (0)
True	Positive (1) – NS	TP 0%	FN -60%
	Negative (0) - S	FP 0%	TN 10%

Optimum gain threshold for the bank



- By default, modeling uses a 50% classification threshold.
- To optimise the gain for the bank, we look for the threshold between 0 and 1 which maximizes the gain function.
- The optimum threshold is 68%

Modeling

Model interpretability

Model interpretability with SHAP library

Global importance of variables

SHAP makes it possible to identify the global importance of variables.

All features contribute to the result of the model.

The higher and "more red" the value of the score source is, the more it contributes to a negative prediction (thus to the loan being accepted).

Local importance of variables

Local features indicate the influence of variables on the prediction of whether or not to lend to a customer.

For this customer, the score source 1 variable is red, so most favorable for a negative prediction (therefore for the loan to be accepted).

Dashboard

Specifications

Specifications

Interactive dashboard

Demande

Réponse

Front-end *(other name: Dashboard)* **allows to visualize :**

- descriptive information about a customer (via filter)
- the comparison of a customer's descriptive information to all others
- the score and interpretation of this score for each client

Demande

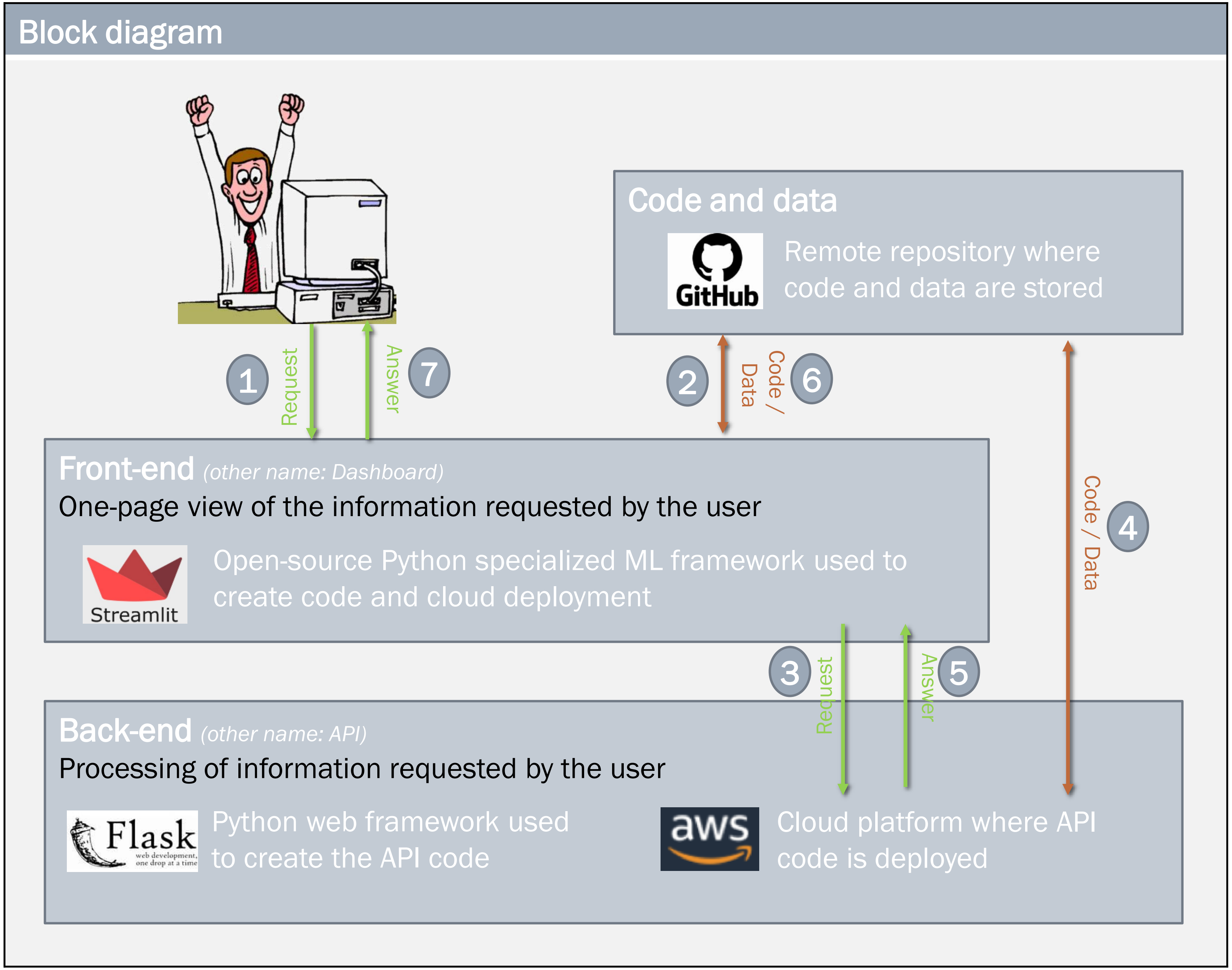
Réponse

Back-end *(other name: API)* **provides:**

- the list of customer identifiers
- descriptive information of a customer
- descriptive information of a variable
- a customer's scoring
- interpretation of the client's scoring

Dashboard

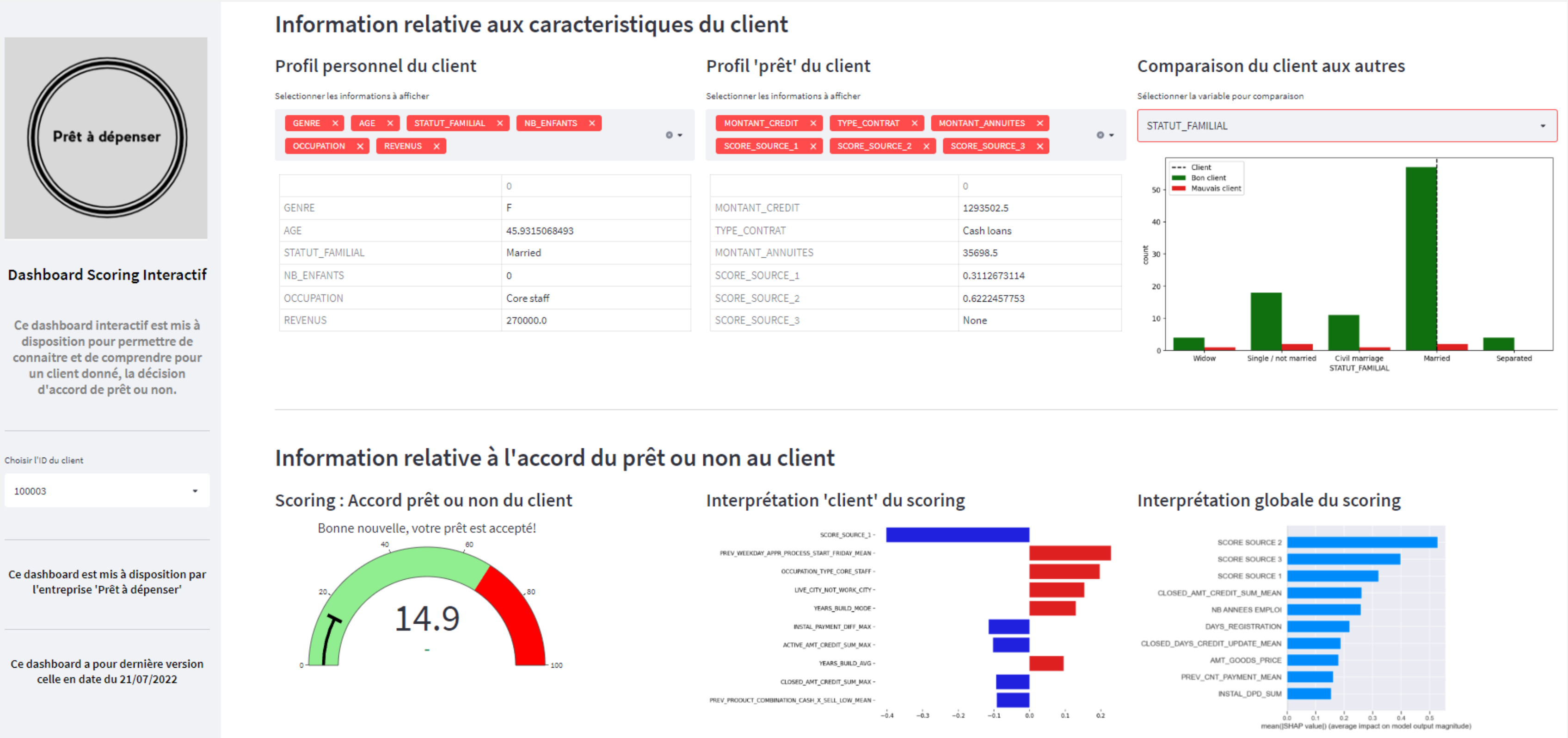
Operation



Dashboard

Visualisation

Dashboard Visualisation



Conclusion

Areas for improvement

Areas for improvement

- Discussion with business teams to improve:
  - feature engineering to a further level
  - gain maximisation function from confirmed business assumptions
  - dashboard – more "user friendly" and able to see the impact of a change of customer variable on his scoring
- From a 'Datascientist' point of view:
  - testing with other algorithms, standardization, ...
  - Re-training the model with a more optimised threshold
- Modeling and interpreting the dataset using more powerful resources (higher processing capacity).

Thank you for your attention!