**Credit risk scoring**

Projet 7- Data Scientist

Open classroom

Isabelle Ravot-Boutherre

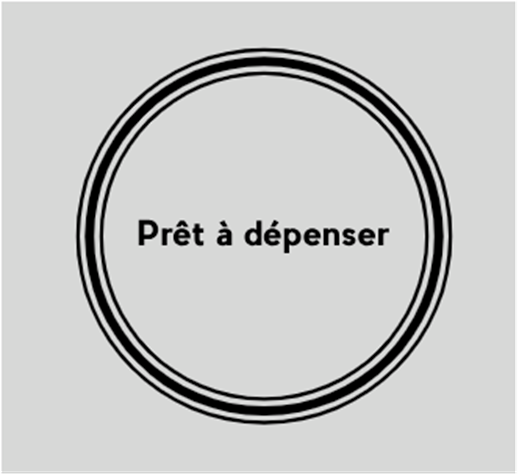


Table des matières

[1. Introduction 3](#_Toc83652657)

[2. The prediction model 4](#_Toc83652658)

[2.1. Data preparation and exploration 4](#_Toc83652659)

[2.2. Choice of the model 5](#_Toc83652660)

[2.2.1. Model output 5](#_Toc83652661)

[2.2.2. Loss function 5](#_Toc83652662)

[2.2.3. Evaluation metric 6](#_Toc83652663)

[2.2.4. Training and evaluating the model 6](#_Toc83652664)

[3. Interpretability of the model 8](#_Toc83652665)

[4. Next improvements 9](#_Toc83652666)

1. Introduction

The objective of this project is to develop a scoring model for the company “Prêt à depenser” in order to predict the probability that a customer will repay his loan or not. This probability will be used to decide whether to grant the loan to the customer or not. To explain the decision to the customer, “Prêt à depenser” wants to present him an interactive dashboard allowing:

* to visualize the score of the model and its interpretation
* a descriptive information about him
* compare the customer data with data from other customers.

To implement this model, “Prêt à dépenser” has at disposal historical loans applications data which contains the information whether the customers had payment difficulties or not.

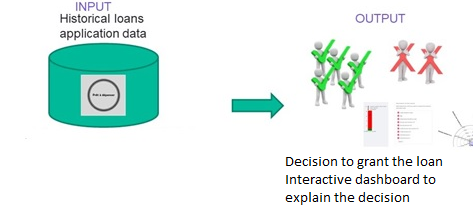


Figure 1. Credit risk project objectives

In this note, it will be explained the method used to train the model, which loss function and evaluation metric have been used, how the model can be interpreted and what are the limit and possible improvements.

1. The prediction model
   1. Data preparation and exploration

For the data exploration and preparation, as recommended , an existing kernel Kaggle has been used: <https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction>.

The data are stored in 7 tables but for the purpose of this project we only used the application\_train table which contains information about 307’511 loans applications described by 122 features. Those features provide information about the loan (for e.g credit amount of the loant, loan annuity, contract type) and about the loan applicant (for e.g. Age, education type, Family status, Housing type ). It also includes the feature TARGET which indicates whether the customer had payment difficulties.

The data exploration has allowed to identify that we are facing an imbalanced class problem. More loans were repaid without difficulties than with difficulties. This will have to be considered when training the model. Not taking this fact into account could lead to a model that will predict all loans has being repaid without difficulties.



Figure 2- Nb of loans with no payment difficulties (value=0) and with payment difficulties (value= 1)

For categorical features, one hot encoding have been applied when several values were possible and label encoding when only two values were possible.

New features have been creating from the existing data to capture important information for telling if a customer may have payment difficulties. Those features are the following ones:

* CREDIT\_INCOME\_PERCENT: the percentage of the credit amount relative to a client's income
* ANNUITY\_INCOME\_PERCENT: the percentage of the loan annuity relative to a client's income
* CREDIT\_TERM: the length of the payment in year
* DAYS\_EMPLOYED\_PERCENT: the percentage of the days employed relative to the client's age
  1. Choice of the model

We selected the Light Gadient Boosting Machine (LightGBM) algorithm because it has a faster training speed and higher efficiency than other boosting algorithm. It has also lower memory usage and a better accuracy. It is compatible with large dataset.

It builds sequential decision trees. Each decision tree is built on the previous error. It uses the gradient descent method to optimize the loss function.

* + 1. Model output

The lightGBM model will provide as output the probability that the customer will have payment difficulties to repay the loan.

From this probability, the loan application can be classified as “Payment difficulties” or “No payment difficulties”.

Probdifficulties <Threshold → 0 (No payment difficulties- the loan can be granted)

Probdifficulties > Threshold →1 (Payment difficulties- the loan should not be granted)

As part of the training of the model, we must identify the optimal threshold for the payment difficulties probability above which the loan should not be granted. This threshold is an hyperparameter.

* + 1. Loss function

The loss function is the function the model will minimize over the training. In this case, we use the default one, the binary log loss. The binary log loss, measures the performance of a classification model whose output is a probability value between 0 and 1.  The log loss increases as the predicted probability diverges from the true value.

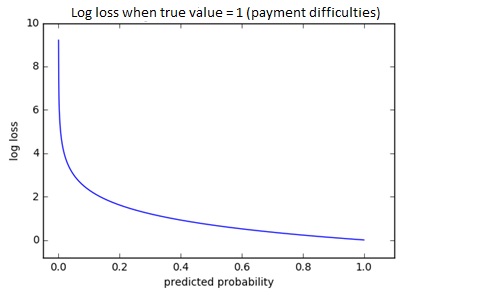


Figure 3.Log loss when the true value is one (Payment difficulties)

Logloss highly penalizes when the model makes a wrong prediction with confidence (for example when the model predicts with a high probability that the customer didn’t have payment difficulties while in reality he had).

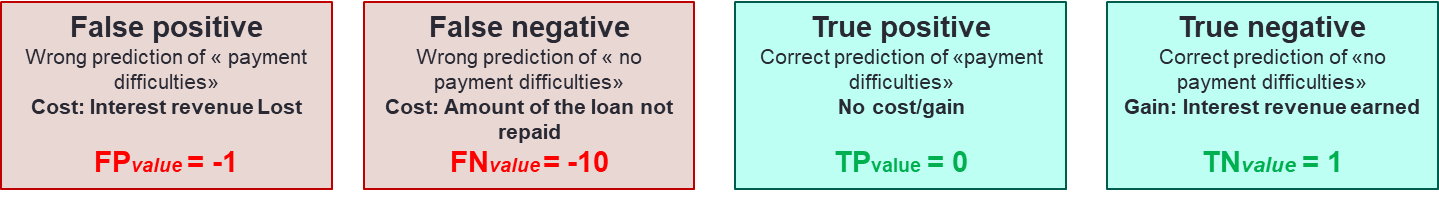
* + 1. Evaluation metric

The evaluation metric is the metric we want the model to optimize. It should reflect the business need.

Granting a loan to a customer that will default is more costly than not granted it to a customer that will repay it. The evaluation metric should reflect this fact. We define our own evaluation metric. We associate to each prediction a value reflecting the gain or cost for the company “Prêt à dépenser” depending if the prediction was correct or not.

* Granting a loan to a customer that will have payment difficulties has a cost corresponding to the amount of the loan not repaid (False negative prediction).
* Not granting a loan to a customer that will not have payment difficulties has a cost corresponding to the interest revenue lost (False positive prediction).
* Not granting a loan to a customer that will have payment difficulties has neither cost nor gain (True positive prediction).
* Granting a loan to a customer that will not have payment difficulties has a gain corresponding to the interest revenue earned (True negative prediction).

We consider that a FN prediction is 10 times more costly than a FP one. A TN predication has a gain equal to the opposite of the cost of a FP one.



We define the evaluation metric as follow

We normalized it to have a value between 0 and 1

Where

(All loans applications are classified correctly)

(All loans application are classified as no payment difficulties)

Defining such an evaluation metric, which reflects the business needs, solves the issue of the imbalanced data.

* + 1. Training and evaluating the model

We used the train-test split procedure to train and evaluate the performance of our model. We use 80% of the data to train the model and keep 20% to test how well the model generalizes on unknown data.

First, we must tune/optimize the hyperparameters. LightGBM has several hyperparameters that can be tuned (n\_estimators, learning\_rate, max\_depth…). In our case, we must also tune the threshold for the payment difficulties probability above which the loan will not be granted.

To make this tuning, we use the technique of K-fold cross validation. The data selected to train the model is divided in K folds. K-1 of this fold is used to train the model with a selected set of values for the hyperparameters we want to tune. The remaining fold is used to validate the performance through the computation of the evaluation metric. This process is repeated until all folds is used once for validation. The performance of the model is evaluated by calculated the evaluation metric as the means of the evaluation metrics of each of the K validation folds. We repeat the technique with different values for the set of hyperparameters.

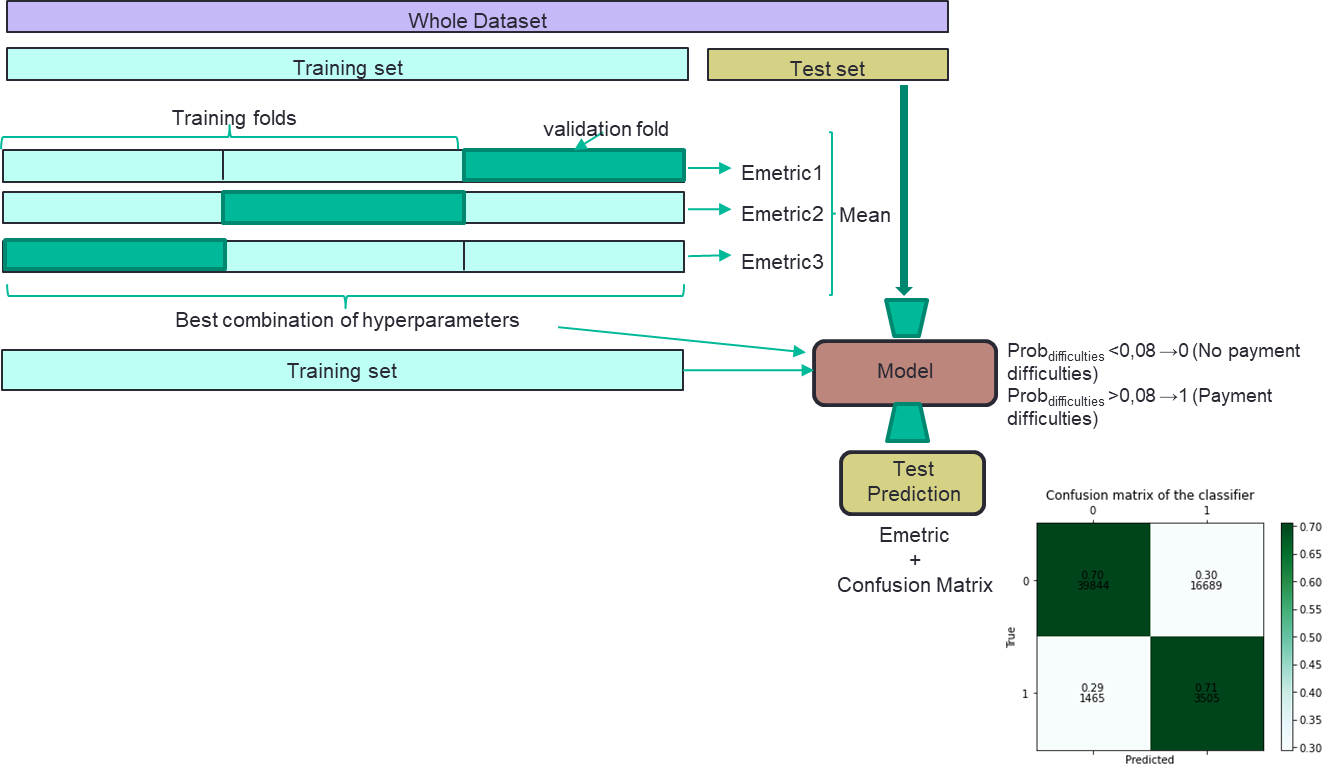


Figure 4 Training and evaluation of the model

Once the best values for hyperparameters are identified, the model can be trained using the whole training set. Then we can evaluate the performance of the model using the test set and by calculating the evaluation metric. To have a more understandable vision of the performance, we also calculate the confusion matrix.

1. Interpretability of the model

One of the objective of this project is also to allow sales representatives to explain why a loan has been granted or not. The lightGBM algorithm provides an output called “feature importances”. For each feature, a score is calculated that represents how relevant is the features towards the prediction.

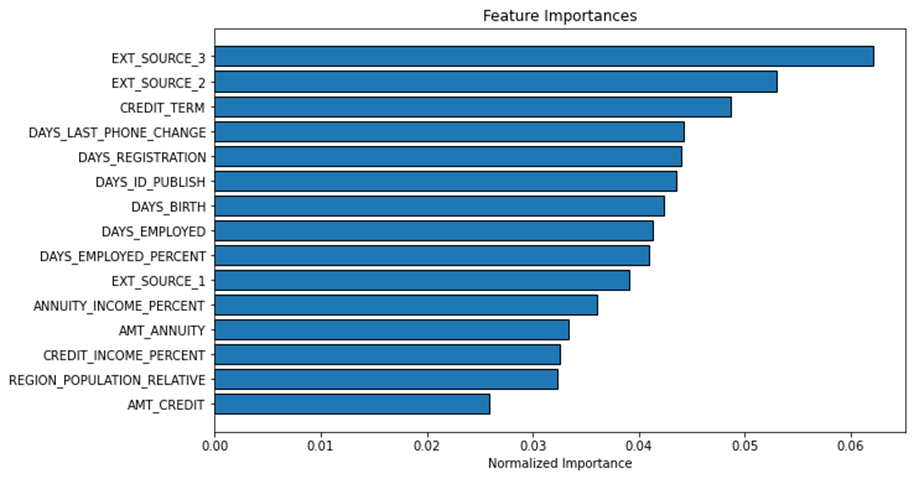


Figure 5. Features importance

We can see that the four features we have created are among the 15 more important features. The two most important features EXT\_SOURCE\_3 and EST\_SOURCE\_2 are opaque features. Finding information about how they have been calculated would be important.

Once, we have the information about the most important features we could design a dashboard for sales representative to compare the values of those features for a particular loan application against others loans applications in order to explain why the loan has been granted or not.

The dashboard can be visualized at the following address: [P7\_3\_Dashboard · Streamlit (creditdashboard.herokuapp.com)](https://creditdashboard.herokuapp.com/)

1. Next improvements

The two objectives of this project has been reached. We built a model able to predict whether the loan should be granted or not. This model takes into account the business need: It is more costly to grant a loan to a customer that will default than not granting it to customer that would have repay it. We provide a dashboard allowing sales representative to explain the decision to grant or not a loan.

However, some possible improvements can be implemented.

Some features, used as input of the model, are opaque and some of them are among the most important features (for e.g. Ext\_sources). To better explain the decision to grant a loan or not, it would be important to obtain more information to what corresponds those features.

We define an evaluation metric based on some hypothesis: Granting a loan to a customer that will default is 10 times more costy that granting a loan to a customers that would have repay it. For those hypotheses to be as close as possible to reality, it would be required to work with the internal team of the company “Pret à dépenser”

For the purpose of this project, we have as recommended based the data exploration and preparation on an existing Kaggle which was using only using part of the data (one table among the seven available). It would be interesting to explore others available features to identify if some of them could improve the performance of the model.