

Project of Application of Big Data

version

Vianney Gonnot, Louis Gailhac, Elodie Gueuret

novembre 22, 2021

Contents

| | |
|--|----------|
| Welcome to Application of big data's documentation! | 1 |
| Part 1 : | 1 |
| Part 2 : | 2 |
| Part 3 : | 2 |
| Indices and tables | 3 |

Welcome to Application of big data's documentation!

Application of bigdata (/Our project/) is a python project, that train us to apply tools and concepts seen in course. It pulls data from the *DataSet ofHome Credit Risk Classification* <<https://www.kaggle.com/c/home-credit-default-risk/overview>>.

To run our program correctly, you will need to run the different python scripts in the following order :

1. Cleaning_Dataset.ipynb
2. Features_Engineering.ipynb
3. Training_model.ipynb
4. Shap.ipynb

Part 1 :

In the first part, we build a machine learning project using jupyter notebook, github, a conda environnement and sphinx. We tried to separate the different workflow into different scripts, one for the data preparation, one for the data preparation, one for the feature engineering, one for the models training and a last one for the prediction.

Data preparation :

We first clean the dataset from all the NAN values.

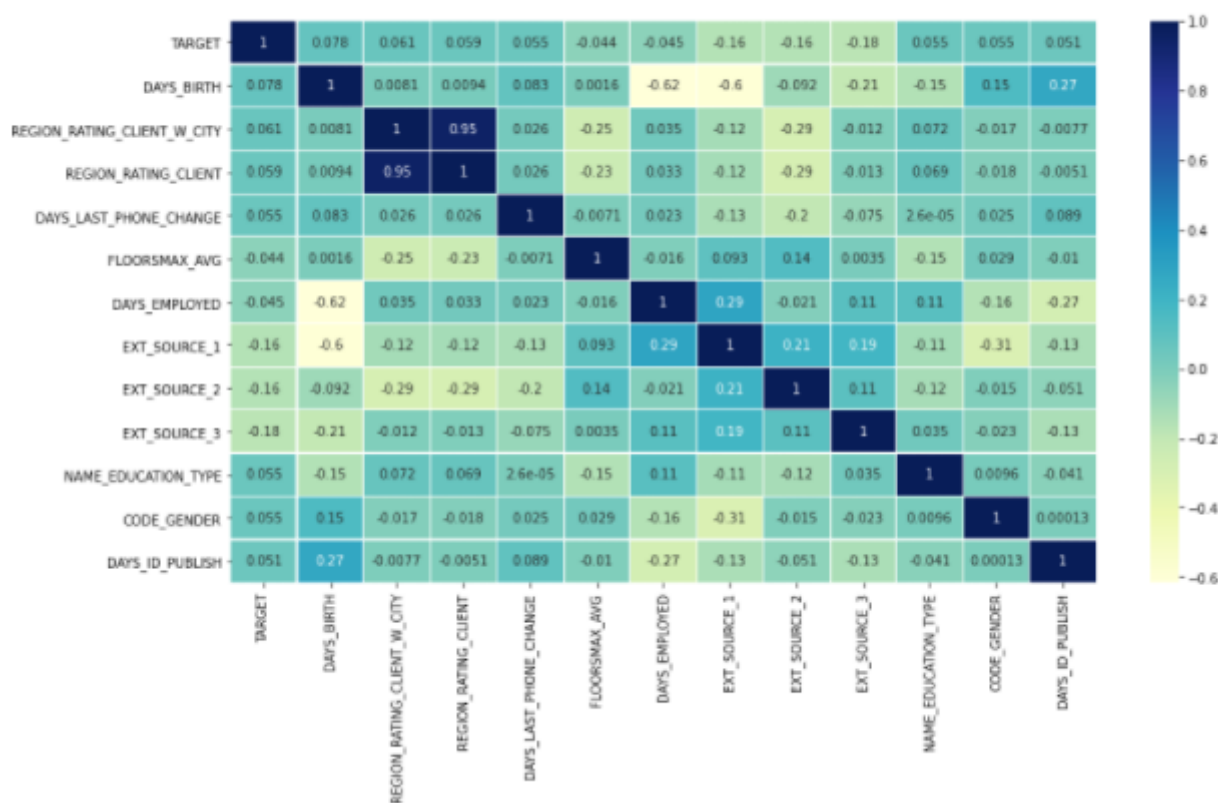
- `init()`, will return the cleaned dataset

Feature engineering :

We have done a correlation matrix, and from that we have kept the most correlated features and deleted the least correlated ones.

Here is the correlation matrix :

Correlation matrix :



- `matrice_corr(df_train,df_test)`, is a void function that show us the correlation matrix

Part 2 :

- **setup_train(df_train,df_test)**, will return four values (X_train, X_test, y_train and y_test)

Models training and predict :

We had to train three models: XGboost, Random Forest and Gradient Boosting. The XGboost model, is done with the optimized distributed gradient boosting library, XGboost. The Random Forest model, consists of many decision trees. The Gradient Boosting model, is an ensemble of weak prediction models(decision trees).

- **XGBC_model(X_train,X_test,y_train,y_test,learning_rate,max_depth,scale_pos_weight)**
- **RF_model(X_train,X_test,y_train,y_test)**
- **GB_model(X_train,X_test,y_train,y_test)**

Those functions train the different models.

All three model, succeed in predicting if a client could get a loan. Most had each around 0.91 of accuracy.

RF accuracy score:

0.9173535353535354

GB accuracy score:

0.9185959595959596

XGB00ST accuracy score:

0.9186363636363636

Part 2 :

In this part, we got introduced to MLFLOW. We decided to track the parameters of the XGboost model. It helped us to choose the best parameter, to have better result, with our model.

Here we can have a look at MLFlow:



Part 3 :

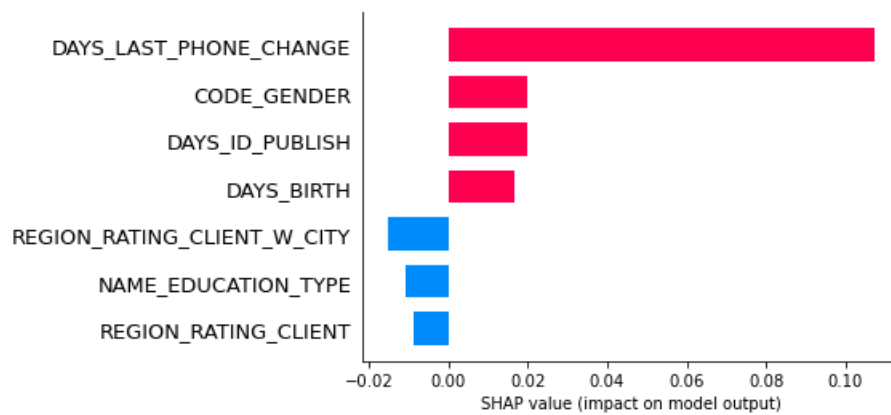
Finally, we used SHAP Library on our XGboost model to understand it. We can visualize three graph by running the following function.

The first one is to visualize the explanations for a specific value, we choose to select the 100th value, and we observed that the day of the last time the person changed his phone had a lot of influence on the result of the prediction.

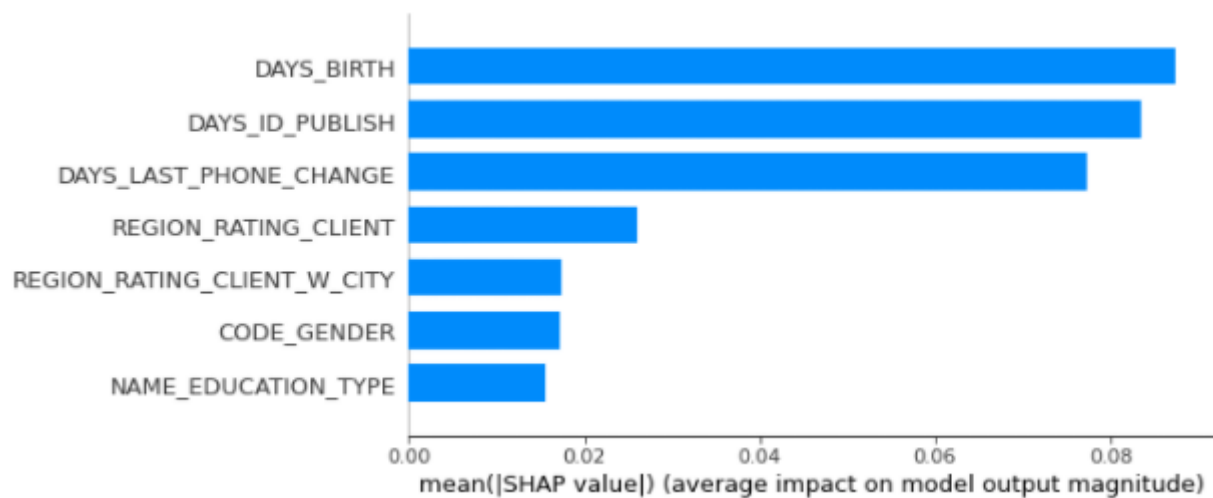
The second, is a the same as the one just seen but for all the values of the dataset. We can see the day of birth is the most influent between all the features.

The last one, is a summary plot for each class of the dataset.

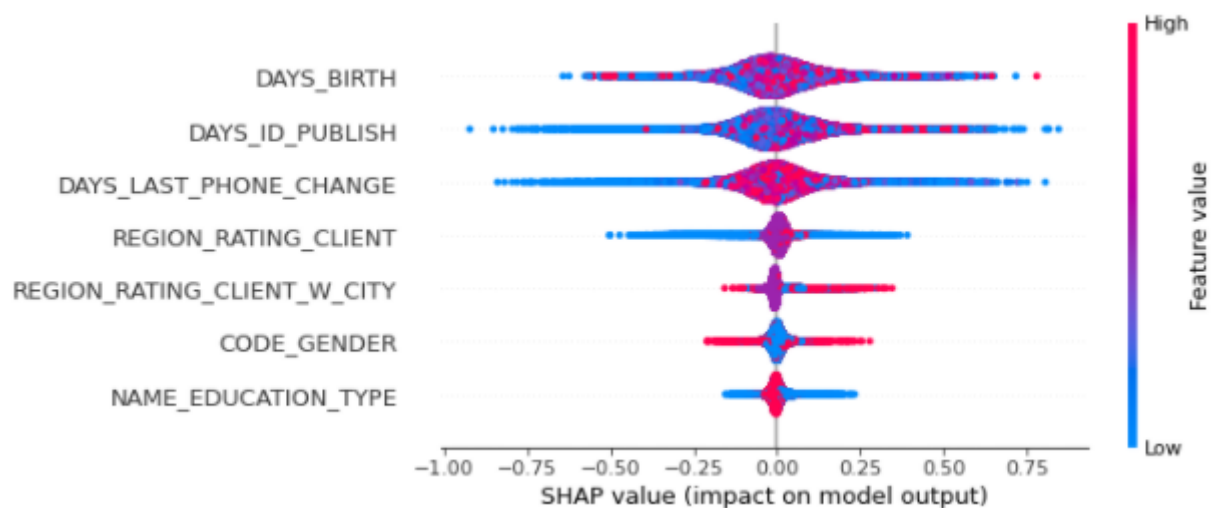
Here is the graph for a specific value :



Here is the graph for all values :



Here a summary plot for each class on the whole dataset :



- `get_explainer(xg_clf,X_train_test)`, is a void function that print the three graphs

Indices and tables

- `search`