CREDIT RISK CLASSIFICATION

PROJET APPLICATION OF BIG DATA

Réalisé par :

- **AMOIN Famien Le Roi (DE1)**
- **NGUEYAP NGUIKAM Godelieve (DE2)**

INTRODUCTION

The project we have developed aims to predict the probability of loan repayment for a lending company. To do this, we used the XGBoost model and implemented a web interface using Flask to allow users to make real-time predictions.

REALISATION

PRELIMINARY

GIT for team collaboration, code & model versioning

- We use a template cookie cutter.
- We use a conda environment t for all our libraries.
- Use a documentation library Sphinx, then we updated it. it is located in the directory ***Credit_Risk_Classification\docs\build\html\***   launch ***index.html***

DATA

The data used for this project comes from the Kaggle's lending app dataset. The dataset is divided into two parts: a training part and a test part. The training part includes more than 300,000 observations and the test part includes more than 100,000 observations.

DATA PREPARATION

Before training the XGBoost model, we prepared the data using the data preparation function. This function performed the following tasks:

- Deleting rows with missing values
- Replace missing values with mean value.
- Encoding of categorical variables

FEATURE ENGINEERING:

After preparing the data, we created new features using the Feature Engineering function. This function made it possible to create new variables which improved the performance of the XGBoost model and select the most important variables thanks to One HOT Encoding

XGBOOST MODEL

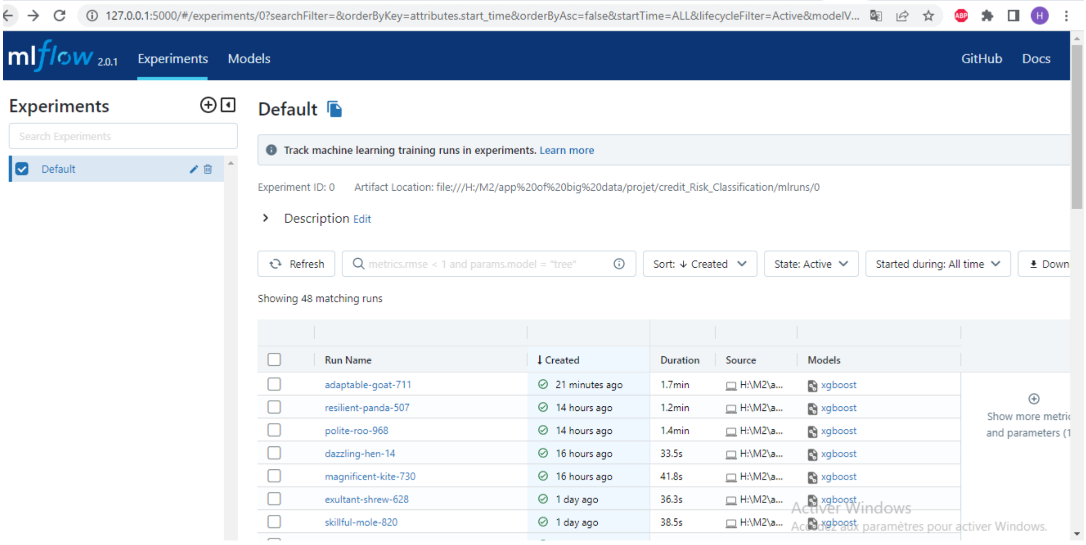
The XGBoost model used for this project was trained using the model training feature. The model was trained using the default hyperparameters such as learning rate, regularization, base value, etc. Hyperparameters can be modified by the user through the web interface.

MODEL EVALUATION

The model was evaluated using the following metrics: precision, recall, F1 score, and accuracy score. The output of prediction is located in ***credit_Risk_Classification\output\prediction\prediction.csv***

MLFLOW PROJECT

- MLProject file is located in ***credit_Risk_Classification\MLproject***
- MLflow outputs is located in ***credit_Risk_Classification\mlruns\0***
- MLflow ui



127.0.0.1:5000/#/experiments/0?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelV...

Default

Track machine learning training runs in experiments. Learn more

Experiment ID: 0   Artifact Location: file:///H:/M2/app%20of%20big%20data/projet/credit_Risk_Classification/mlruns/0

Sort: ↓ Created   State: Active   Started during: All time

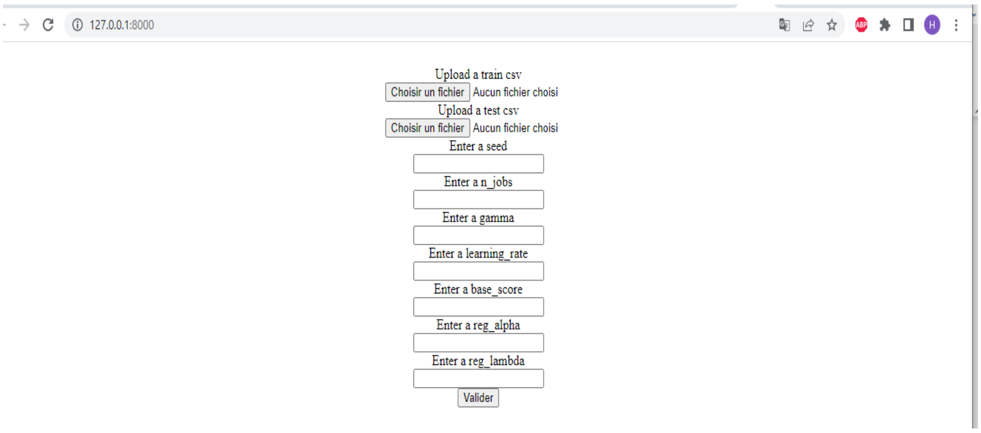
Showing 48 matching runs

□	Run Name	↓ Created	Duration	Source	Models
□	adaptable-goat-711	✓ 21 minutes ago	1.7min	H:\M2\a...	xgboost
□	resilient-panda-507	✓ 14 hours ago	1.2min	H:\M2\a...	xgboost
□	polite-roo-968	✓ 14 hours ago	1.4min	H:\M2\a...	xgboost
□	dazzling-hen-14	✓ 16 hours ago	33.5s	H:\M2\a...	xgboost
□	magnificent-kite-730	✓ 16 hours ago	41.8s	H:\M2\a...	xgboost
□	exultant-shrew-628	✓ 1 day ago	36.3s	H:\M2\a...	xgboost
□	skillful-mole-820	✓ 1 day ago	38.5s	H:\M2\a...	xgboost

REST SERVER IMPLEMENTATION

We implemented a REST Server using Flask that allows users to make real-time predictions by providing the training and test data paths along with the desired hyperparameters.

- Form



Upload a train csv

Choisir un fichier Aucun fichier choisi

Upload a test csv

Choisir un fichier Aucun fichier choisi

Enter a seed

Enter a n_jobs

Enter a gamma

Enter a learning_rate

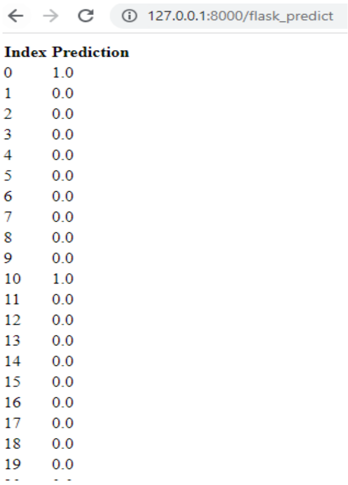
Enter a base_score

Enter a reg_alpha

Enter a reg_lambda

Valider

- route for prediction

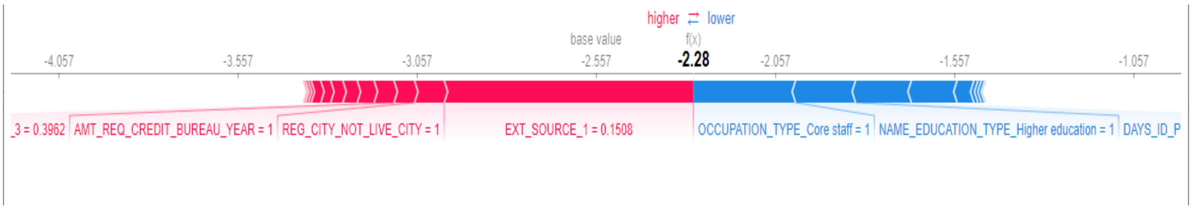


```
Index Prediction
0     1.0
1     0.0
2     0.0
3     0.0
4     0.0
5     0.0
6     0.0
7     0.0
8     0.0
9     0.0
10    1.0
11    0.0
12    0.0
13    0.0
14    0.0
15    0.0
16    0.0
17    0.0
18    0.0
19    0.0
```

SHAP LIBRARY FOR EXPLANATIONS

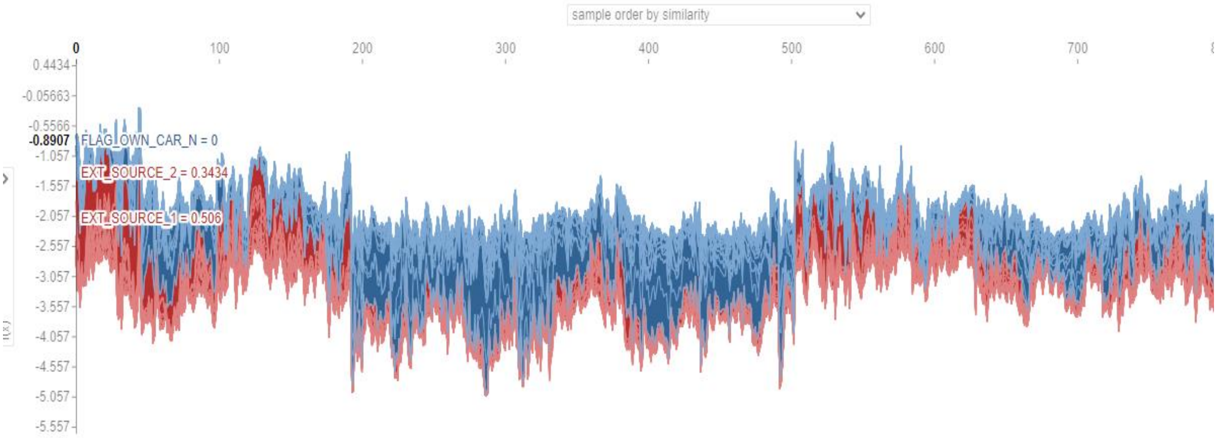
SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. the explanations are in the repository***:*** ***Credit_Risk_Classification\output\figures***

● Visualize explanations for a specific point of your data set: ***explanationSpecificPoint.html*** (the image is saved in .html because it is interactive)

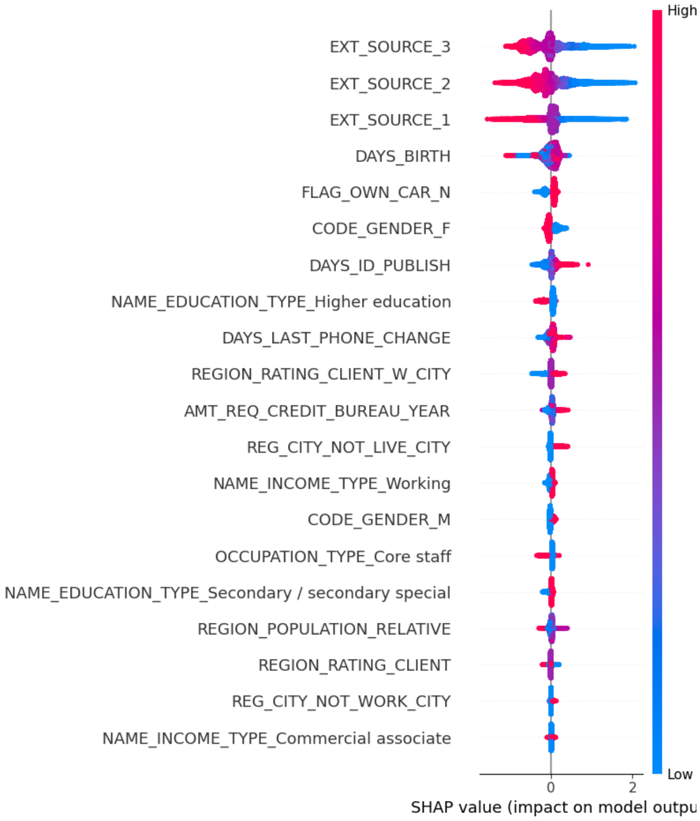


● Visualize explanations for all points of your data set at once: ***explanationAllPoint.html*** (

the image is saved in .html because it is interactive)



- Visualize a summary plot for each class on the whole dataset: ***SummaryEachClass.png***



CONCLUSION

In conclusion, we have successfully built a machine learning project using MLflow and a Flask REST API. We were able to track experiments and log parameters, code versions, and performance metrics, making it easier to reproduce our results. The Flask REST API allowed us to deploy our model and make predictions on new data. And SHAP to explain the prediction. Overall, this project highlights the importance of utilizing tools such as MLflow to streamline and simplify the machine learning development process.