24/08/2023

Keshika Dabidin Audam

Lien GITHUB :

Note méthodologique

Projet 7 OpenClassrooms

1. Introduction

L’entreprise ‘Prêt à Dépenser’ souhaite développer un modèle de Scoring de la probabilité de défaut de paiement du client afin d’assister à la décision d'accorder ou non un prêt à un client potentiel en s’appuyant sur des sources de données variées (données personnelles, données comportementales, données provenant d'autres institutions financières, etc.).

Le modèle de Scoring s’établit à partir des algorithmes de machine learning. Ces algorithmes divergent dans leurs méthodes de modélisation mais se basent sur le même principe. Le projet a été traité à l’aide de 3 algorithmes de gradient boosting (LightGBM, XGBoost et CatBoost) utilisés en raison de leur simplicité d’interprétation et aussi en raison de leur efficacité.

La démarche suivante est appliquée pour mettre en place le modèle de machine learning :

# Récupération des données

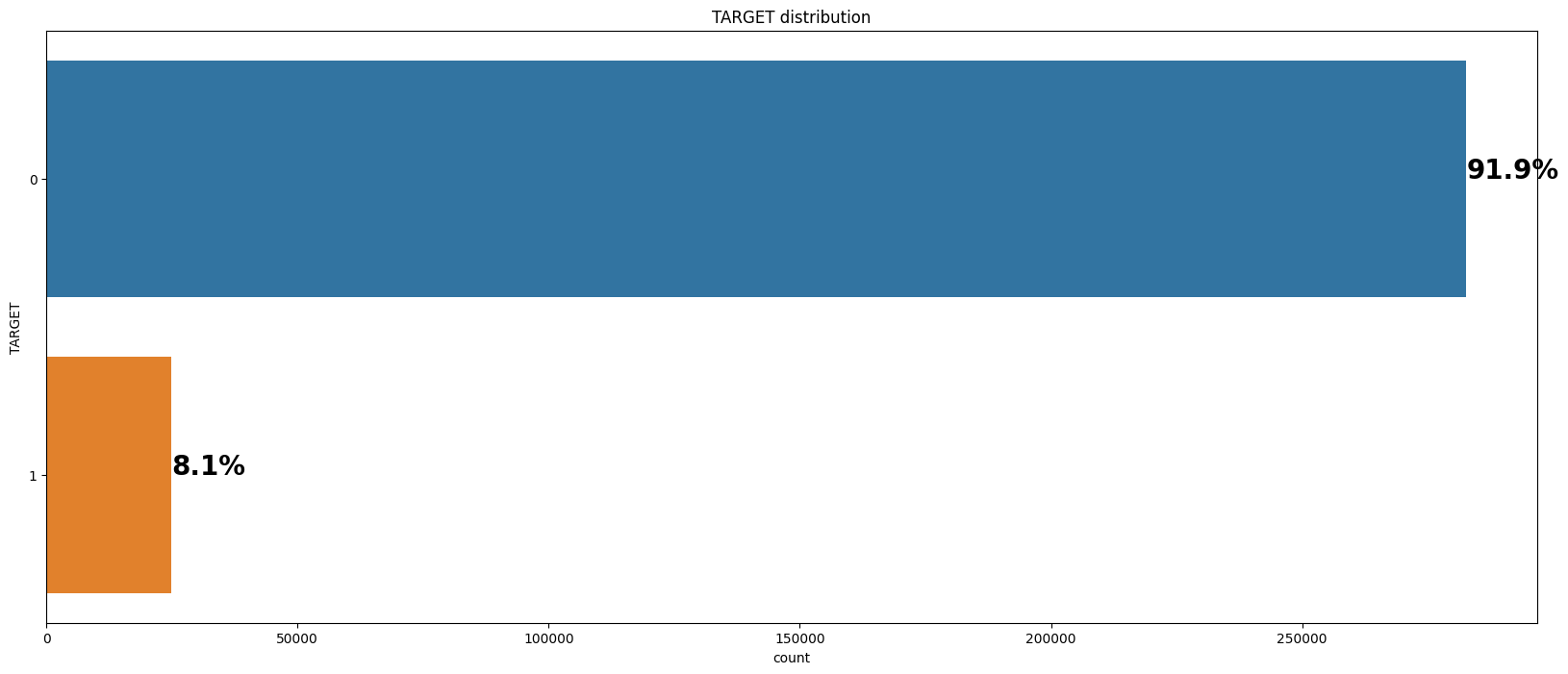
La première étape consiste à récupérer des données cohérentes afin de former la base sur laquelle l’algorithme de Machine Learning va apprendre. Des données relatives à différents clients sont récupérées sur [Kaggle](https://www.kaggle.com/c/home-credit-default-risk/data).

Un [kernel Kaggle](https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction) est utilisé pour faciliter l’exploration des données nécessaires à l’élaboration du modèle de scoring. La mise en œuvre de statistiques descriptives découle ensuite et permet de comprendre et d’avoir une vue d’ensemble des données grâce à de la visualisation graphique.

# Analyse Exploratoire des données

Il y a 307511 observations pour chaque prêt séparé et 122 features incluant le TARGET (label que l'on souhaite prédire) Le target est 1 si l'individu est éligible pour le prêt, 0 sinon.

## La variable TARGET



Le jeu de données est non-équilibré. Il y a beaucoup plus de prêts remboursés à temps que de prêts non remboursés. Une fois que nous entrons dans des modèles d'apprentissage automatique plus sophistiqués, nous pouvons pondérer les classes par leur représentation dans les données pour refléter ce déséquilibre.

1. La méthodologie d'entraînement du modèle (2 pages maximum)

Suivi via github :

* Push : git push 'remote\_name' 'branch\_name'
* Creating a new repository
* Open your Git Bash
* Create your local project in your desktop directed towards a current working directory
* Initialize the git repository
* Add the file to the new local repository
* Commit the files staged in your local repository by writing a commit message
* Copy your remote repository's URL from GitHub
* Add the URL copied, which is your remote repository to where your local content from your repository is pushed
* Push the code in your local repository to GitHub
* View files on github
* <https://www.datacamp.com/tutorial/git-push-pull>

Hello

1. Le traitement du déséquilibre des classes (1 page maximum)
2. La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation (1 page maximum)
3. Un tableau de synthèse des résultats (1 page maximum)
4. L’interprétabilité globale et locale du modèle (1 page maximum)
5. Les limites et les améliorations possibles (1 page maximum)
6. L’analyse du Data Drift (1 page maximum)

Librairie Evidently