Note Méthodologique

Implémentez un modèle de scoring

Openclassroom Projet 7

Stéphane LUBIN

Formation Data Scientist

Sommaire

1 . Introduction

a. Mission

La société prêt à dépenser souhaite mettre en place un modèle de scoring pour prendre des décisions concernant l'attribution ou le refus de crédit . Elle veut mettre en place une classification binaire pour approuver ou refuser l'attribution d'un crédit.

De plus, pour plus de transparence avec le client est important de montrer les facteurs de décision les plus importants.

Le cahier des charges est le suivant:

- Réaliser un modèle automatique de scoring qui permettra de donner la probabilité qu'un client soit en défaut de remboursement du crédit. Pour ensuite prendre une décision sur l'attribution le refus ou l'acceptation d'un prêt.

- Réaliser un Dashboard interactif pour aider le service relation client dans sa gestion des dossiers et d'interprétation des prédictions faite par le modèle, ainsi que l'amélioration des connaissances et des informations concernant le client.

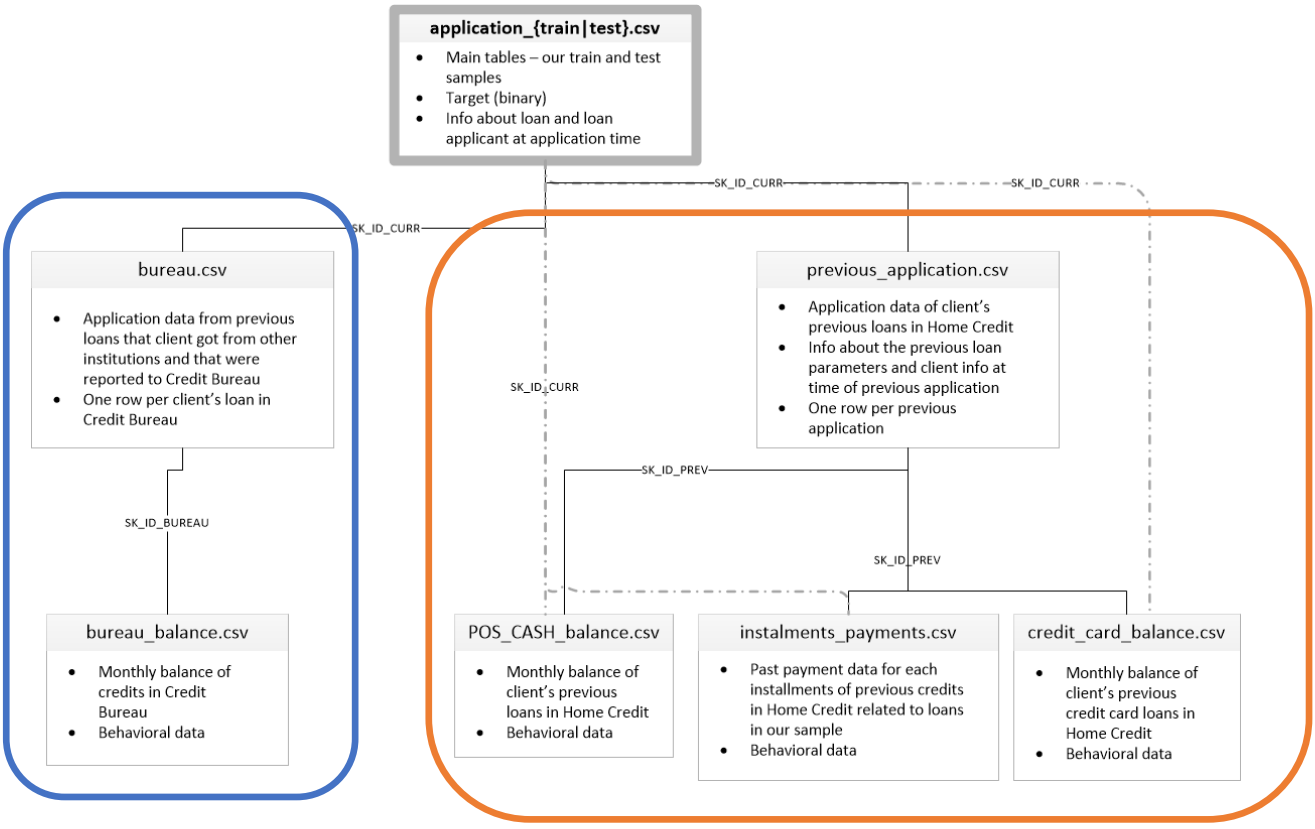
- Mettre en production le modèle de Scoring et de prédiction via une API, le Dashboard interactif fera appel à l'API pour les prédictions.

Le projet suit le schéma suivant:

- Prétraitement des données
- Modélisation et choix des modèles
- Oversampling
- Comparaison des modèles
- Détermination de la fonction coût de Métrique d'évaluation
- Entraînement et optimisation du modèle
- Détermination du seuil
- Élaboration de l'API et du Dashboard
- Vérification data Drift via Evidently

b. Présentation des données

Le jeu de données est composé de huit fichiers il y a des fichiers qui contiennent les informations principales. Tels que sexe, l'âge, le lieu de résidence. Il y a deux fichiers qui contiennent toutes les informations des autres organismes financières. Le cadeau de fichier contient des informations concernant historique du client. Tels que les demandes de crédit immobilier antérieur, les historiques de remboursement et les soldes mensuels.



2. Méthodologie d'entrainement

a. Prétraitement des données

Pour cette première étape, j'ai réalisé le traitement en utilisant un Kernel Kaggle
https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script
Il permet de traiter toutes les valeurs aberrantes, il crée des nouvelles variables intéressantes, j'ai réalisé un One hot encodage sur les variables catégorielles, et réalise la fusion des fichiers.

Il est important de noter que la variable cible "TARGET" prend deux valeurs et présente un déséquilibre marqué (8%/92%)

0 : positif non défaillant, ce qui signifie que le client est solvable.

1 : négatif défaillant, indiquant que le client n'a pas remboursé son prêt.

Nous travaillons donc sur un modèle de classification binaire en tenant compte de ce déséquilibre, avec comme objectif de prédire la probabilité de défaut de paiement.

La création de la plupart des caractéristiques repose sur l'application des fonctions min, max, mean, sum et var aux tableaux groupés. Il y a peu de sélection de caractéristiques et cela peut entraîner un surajustement en raison du grand nombre de caractéristiques interconnectées.

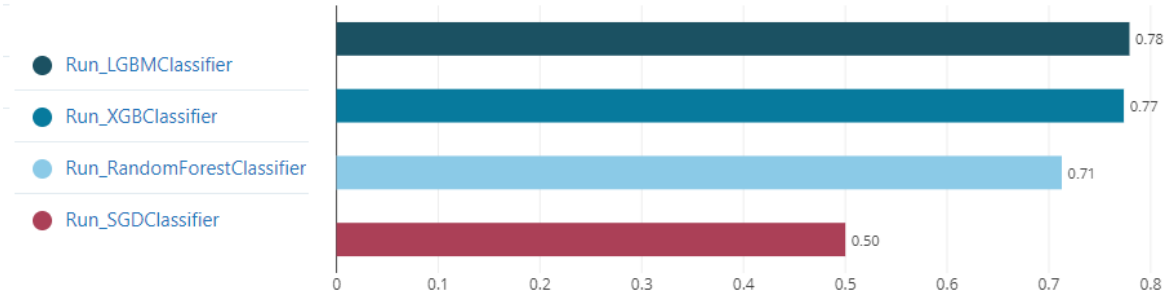
Les idées clés suivantes ont été mises en œuvre :

- Division ou soustraction de caractéristiques importantes pour obtenir des ratios, tels que le rapport entre l'annuité et le revenu.
- Pour les données provenant du Bureau, des caractéristiques spécifiques ont été créées pour les crédits actifs et les crédits fermés.
- Pour les demandes précédentes, des caractéristiques spécifiques ont été créées pour les demandes approuvées et refusées.
- Modularité : une fonction a été créée pour chaque table, à l'exception de "bureau_balance" et "application_test".
- Encodage one-hot pour les caractéristiques catégorielles.
- Toutes les tables ont été fusionnées avec le tableau "application" en utilisant la clé SK_ID_CURR, à l'exception de "bureau_balance".

b. Méthodologie

.
Il existe bon nombre de machines Learning, plus ou moins performant plus ou moins complexe. Mon choix s'est porté sur les modèles suivants.:
SGD Classifier, RandomForestClassifier, XGBoost et LightGBM

La première étape consiste à entraîner les modèles sur le jeune entraînement, avec les hyper paramètres de base et comparer les métriques d'évaluation du modèle tels que la précision, le Recall ou l'AUC. On utilise MLflow, une plateforme qui facilite la gestion et le suivi, des modèles d'apprentissage automatique, la reproductibilité et le déploiement des modèles. Il offre des fonctionnalités pour suivre les expériences, gérer les versions des modèles et intégrer différentes bibliothèques d'apprentissage automatique.



Run Name	Created	Duration	AUC test	Accuracy Test	F1 score test	Précision test	Recall test
Run_LGBMClassifier	1 hour ago	37.2s	0.779	0.919	0.076	0.588	0.04
Run_XGBClassifier	1 hour ago	2.8min	0.774	0.919	0.115	0.511	0.065
Run_RandomForestClassifier	1 hour ago	10.5min	0.713	0.918	0.002	0.667	0.001
Run_SGDClassifier	1 hour ago	2.1min	0.5	0.918	0	0	0

Oversampling :

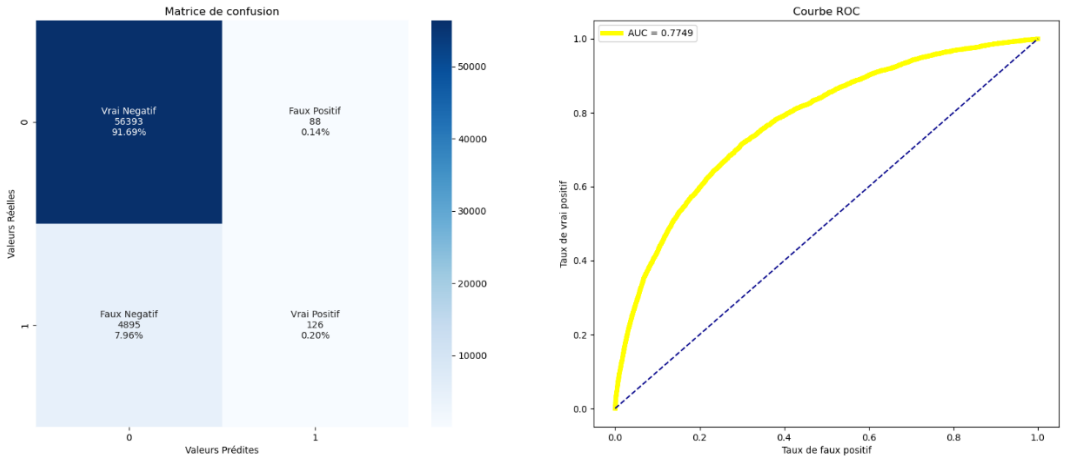
L'étape suivante consiste à gonfler artificiellement le jeu d'entraînement pour assurer un équilibre entre les classes de la variable cible avec SMOTE. SMOTE est une technique d'oversampling qui crée de nouveaux exemples synthétiques pour la classe minoritaire en utilisant des voisins proches, aidant ainsi à équilibrer les classes et à améliorer les performances des modèles d'apprentissage automatique.

Choix :

On réalise à nouveau une comparaison des métriques de chaque modèle après avoir réalisé un Oversampling

Run Name	Created	Duration	AUC test	Accuracy Test	F1 score test	Précision test	Recall test
Run_LGBMClassifier_avec_SMOTE	49 minutes ago	1.5min	0.775	0.919	0.048	0.589	0.025
Run_XGBClassifier_avec_SMOTE	57 minutes ago	8.1min	0.769	0.918	0.108	0.467	0.061
Run_RandomForestClassifier_avec_SMOTE	1 hour ago	17.0min	0.712	0.918	0.004	0.563	0.002
Run_SGDClassifier_avec_SMOTE	1 hour ago	8.6min	0.5	0.918	0	0	0

Mon choix s’est porté sur LightGBM Classifier. LightGBMClassifier est un algorithme de classification rapide et efficace basé sur le gradient boosting. Il offre de bonnes performances prédictives, une optimisation de la mémoire et une gestion pratique des données manquantes. Il est largement utilisé dans les problèmes de classification nécessitant une modélisation précise et rapide.



3. Fonction coût

Le but est de ne pas perdre de l’argent sans forcément trop pénaliser les opportunités d’en gagner. Voici les 2 cas à suivre.

Les Faux Positifs (FP) : les cas où la prédiction est positive, mais où la valeur réelle est négative. Perte d'opportunité si le crédit client est refusé à tort, alors qu'il aurait été en mesure d'être remboursé.

Les Faux Négatifs (FN) : les cas où la prédiction est négative, mais où la valeur réelle est positive. Perte réelle si le crédit client accepté se transforme en défaut de paiement.

Pour atteindre ce but on pénalise ces 2 cas, en leur appliquant un coefficient négatif.

On calcule le taux de chaque cas : vrai et faux positif, vrai et faux négatif, on multiplie ce taux par chaque cas et on y applique un coefficient.

gain total = (taux VN*VN) + (taux VP*VP) – 2*(taux FP*FP) – 20*(taux FN*FN)

les gains max = taux VN*(VN+FN) + taux VP(VP+FP) représentent une prédiction parfaite

les pertes max = -20* taux FN*(VN+FN) - 2 *taux FP(VP+FP) représentent une mauvaise prédiction

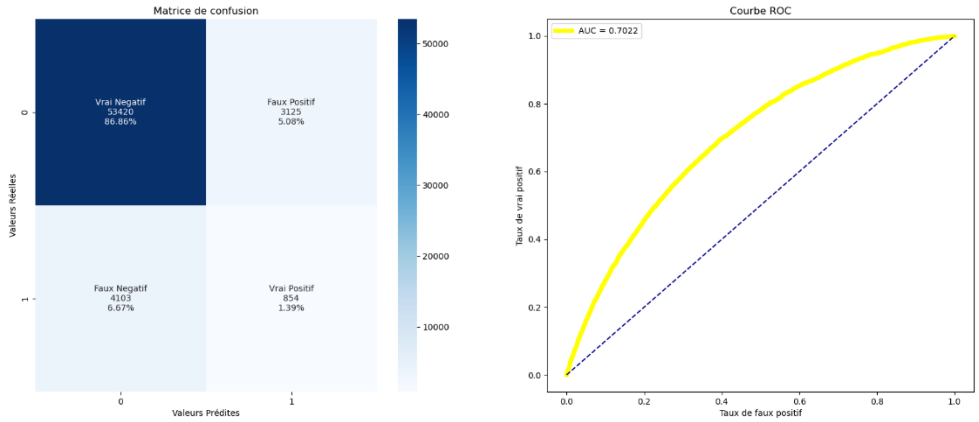
le score est (gain total – gain min)/(gain max – gain min)

4. Optimisation

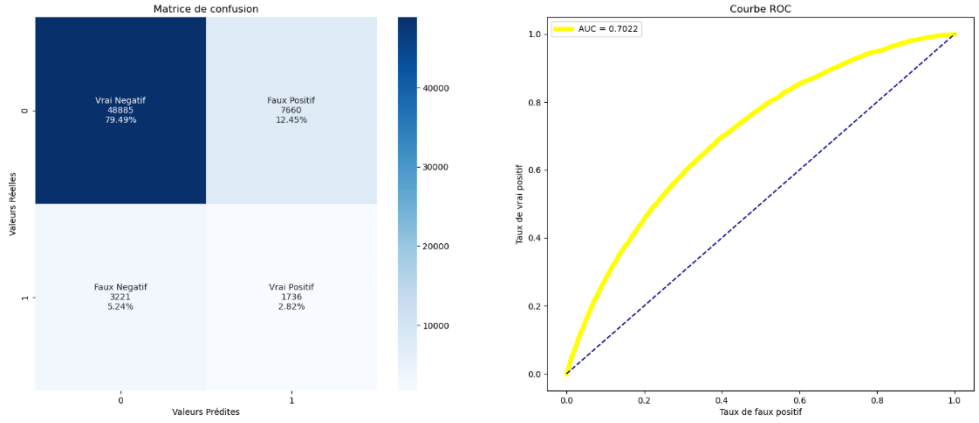
Features selection :

Une fois le modèle choisi, on réalise un choix des variables avec RFECV. RFECV est une méthode de sélection de caractéristiques récursive qui élimine itérativement les caractéristiques moins importantes d'un ensemble de données en utilisant la validation croisée pour évaluer les performances du modèle. Cela permet de trouver le sous-ensemble de caractéristiques le plus informatif pour la prédiction, améliorant ainsi l'efficacité et la précision des modèles d'apprentissage automatique.

Je réalise une validation croisée via Hyperopt. En utilisant comme métrique le score métier. Une fois, le modèle entraîner je vérifie les métriques d'évaluation du modèle.

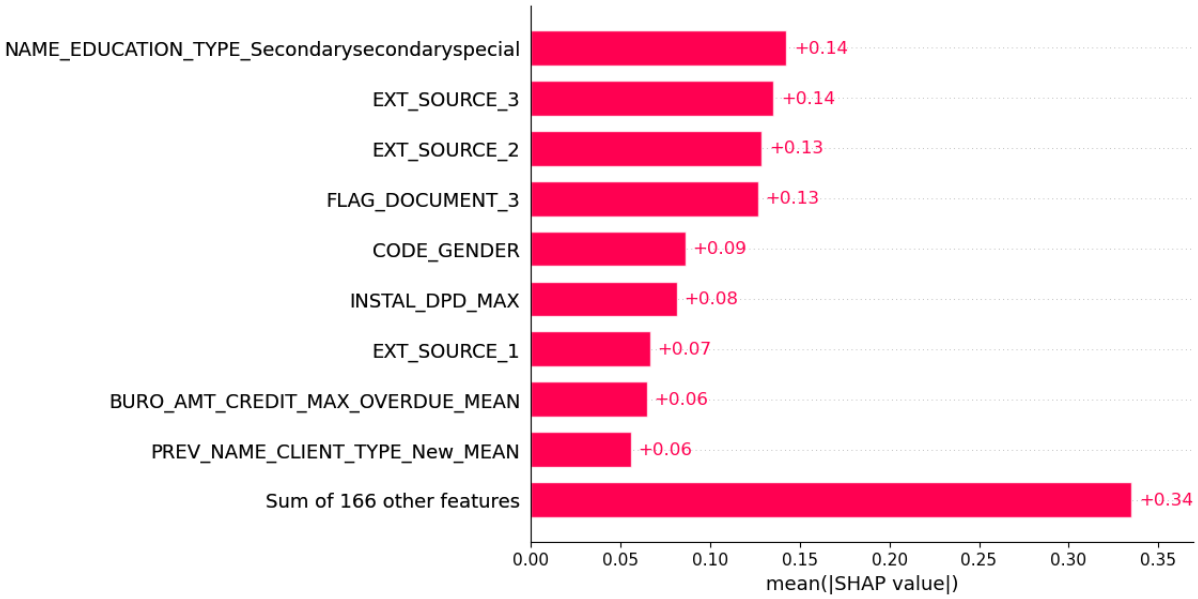


Une fois cette étape fini, nous terminons le seuil de probabilité, qui permet d'optimiser le score métier. On compare ensuite le résultat avec et sans le seuil.

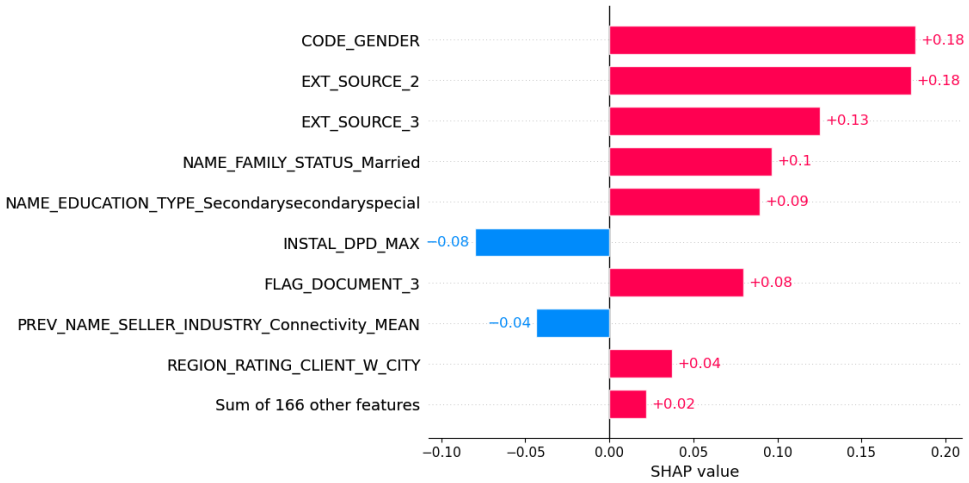


Run Name	Created	Duration	AUC test	Accuracy Test	F1 score test	Précision test	Recall test
LightGBM_optimise_avec_seuil	22 hours ago	3.5s	0.702	0.805	0.242	0.176	0.386
LightGBM_optimise_sans_seuil	22 hours ago	1.9min	0.702	0.882	0.191	0.215	0.172

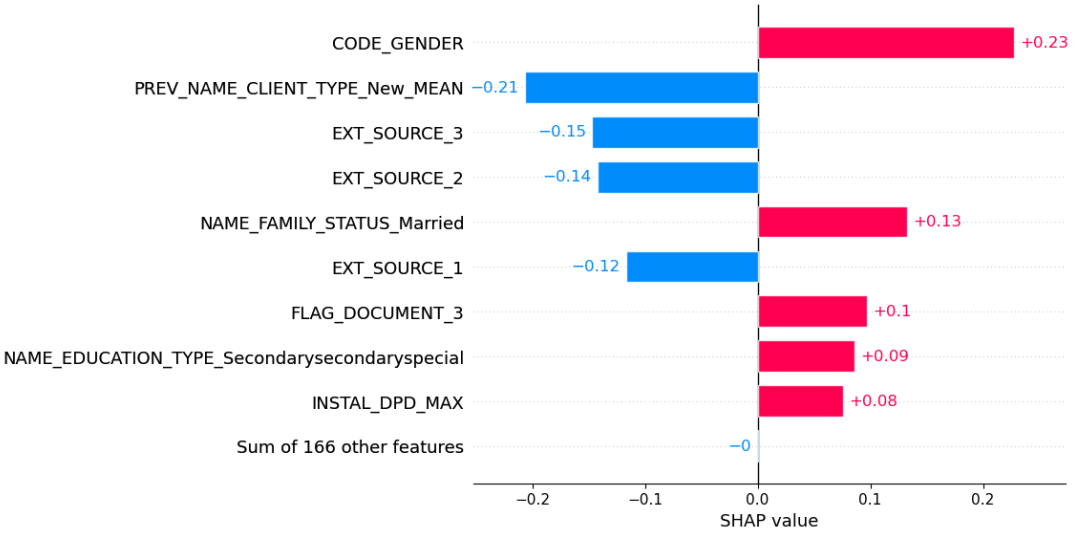
Maintenant, je porte mon attention sur l'importance des variables, dans la décision des probabilités. On commence par regarder des variables globalement



Cas refusé



Cas accepté

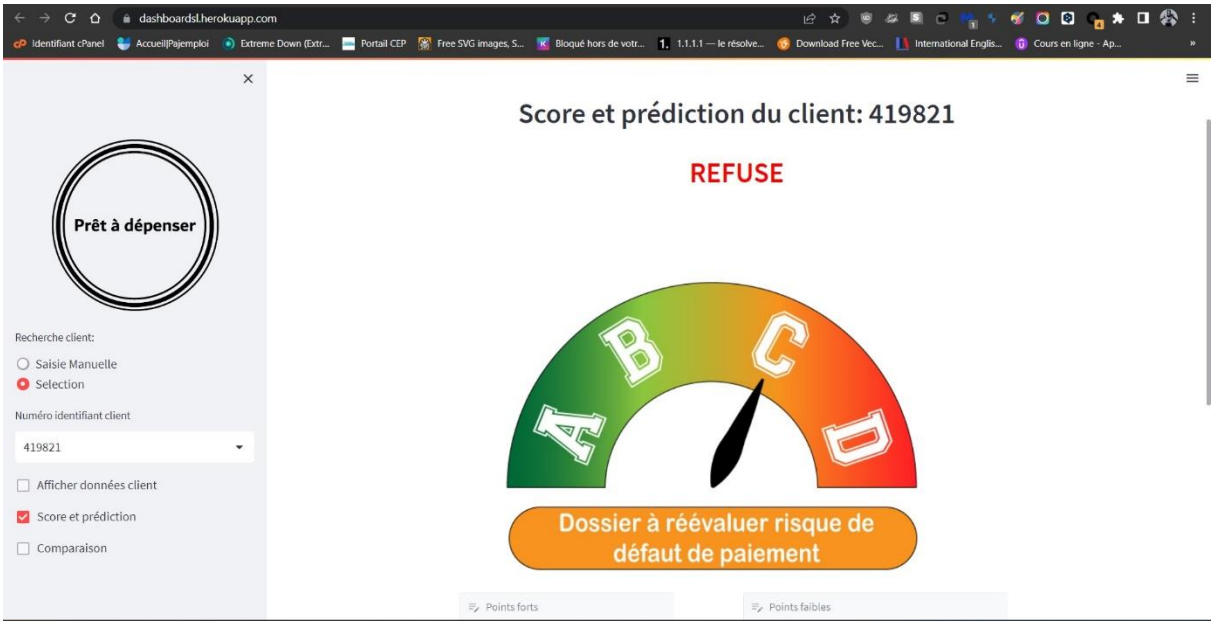


5. Dashboard et Datadrift

Dashboard :

Je réalise une API via le Framework FastAPI, le modèle optimisé et est chargé dans API, celle-ci réalisera les prédictions, il fera une extraction de l'importance des variables dans la décision, des prédictions, de façon locale et globale.

Une fois l'API réalisée, c'est au tour du Dashboard interactif réalisé via Streamlit, le Dashboard interactif permet d'avoir les informations concernant le client, visualiser le score établi par le modèle et la décision d'attribution de refus du prêt, mais aussi voir les points forts et les points faibles du dossier client, et donc avoir un retour sur les principales causes de refus ou d'acceptation, il permet également de comparer les clients par rapport à la clientèle, via des visuels.

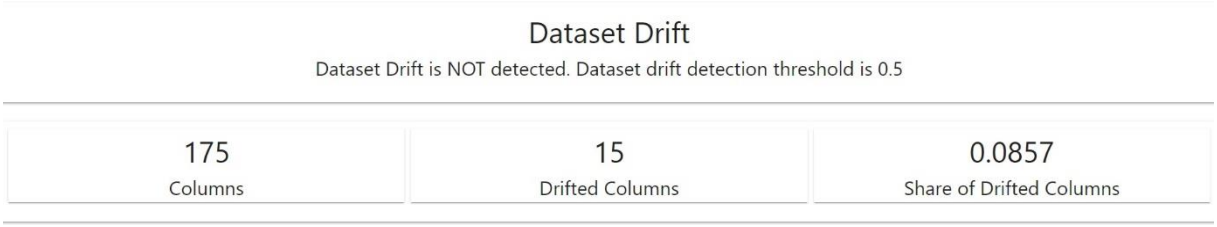


Liens de l'API : https://prediction-api.herokuapp.com/

Liens du Dashboard : https://dashboardsl.herokuapp.com/

DATA DRIFT

Le Data Drift fait référence aux changements dans les caractéristiques statistiques des données au fil du temps. Il peut affecter la performance des modèles d'apprentissage automatique et nécessite une surveillance continue et des mesures d'adaptation pour maintenir la précision des prédictions.



6. Limites et Améliorations

Pour résoudre le problème de classification binaire à partir des 8 fichiers fournis par Prêt à dépenser, nous avons dû utiliser plusieurs techniques d'apprentissage automatique. Cela inclut le rééquilibrage des classes, l'ingénierie des caractéristiques, la sélection des variables, le choix de métriques adaptées à notre domaine d'activité, ainsi que la réflexion sur le compromis entre le taux de faux négatifs et le taux de faux positifs, et sur le réglage du seuil de décision.

Cependant, pour affiner les hyperparamètres du modèle LightGBM Classifier et trouver le seuil optimal pour atteindre nos objectifs, il serait bénéfique d'avoir une expertise métier sur le taux de faux négatifs/positifs. Cette expertise nous aiderait à ajuster les paramètres du modèle de manière plus précise et à trouver le bon équilibre pour répondre aux exigences spécifiques de notre problématique.

Mais beaucoup d’axes peuvent être améliorer ou tester tels que la technique d’équilibrage des classes cibles. Les experts métiers pourraient nous aider à créer une métrique bancaire plus efficace et adaptée et pourraient nous donner leur avis sur l’intérêt des nouvelles variables créées et pourquoi pas nous indiquer un seuil.

L’aspect visuel du Dashboard interactif pourrait également être améliorer, notamment les graphique de comparaison, ou bien la création d’une fonction de pré-simulation avec peu de données client en entrée.