2022 | Par : Souleymanova Adèle

Implémentez un modèle de scoring

[Parcours Data Science / Projet 7/ Openclassrooms]

Note Méthodologique

Description détaillée de :

- **La méthodologie d'entraînement du modèle**
- **La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation**
- **L'interprétabilité globale et locale du modèle**
- **Les limites et les améliorations possibles**

Méthodologie d'entrainement du modèle

1) Compréhension de la problématique et le travail demandé - compréhension du besoin métier

Dans ce projet, nous cherchons à créer un outil de « scoring crédit ». Cet outil a pour objectif de calculer la probabilité pour un client donné de rembourser le crédit qu'il a emprunté.

2) Compréhension du jeu de données

Le jeu de données est mis à disposition sous forme de 7 fichiers CSV :

- Application_train/application_test : contient des informations sur chaque demande de crédit effectuée à l'institution « Home Credit ». Chaque accord de crédit est identifié par la variable SK_ID_CURR.
  La variable TARGET indique 0 pour un crédit remboursé et 1 Crédit non remboursé.
- Bureau : sont les informations sur les demandes de crédits antérieures de ces mêmes clients qui proviennent d'autres institutions financières.
- Previous_application : ici figure les informations sur les emprunts antérieures des mêmes clients chez l'institution « Home Credit ».
  Ces clients sont identifiés par la variable SK_ID_PREV.
- POS_CASH_BALANCE : sont les données mensuelles sur le financement au point de vente que les clients ont obtenu avec « Home Credit »
- Credit_card_balance : informations mensuelles sur les cartes de crédit.
- Installments_payment : chaque ligne représente le paiement de crédit et une ligne pour chaque paiement manqué.

3) Prétraitement des données

Cette partie est réalisée en plusieurs étapes de prétraitement des données. La plupart des variables sont créées en appliquant l'agrégation (min, max, mean, sum, var).

Les valeurs nulles ont été traitées par élimination de colonnes qui continent trop de valeurs manquantes, soit par imputation avec la valeur modale.

Encodage par One-Hot Encoding a été réalisé pour les variables catégorielles.

Ensuite les différentes tables de données ont été jointes.

Le jeu de données est déséquilibré. Il contient des données de deux catégories de personnes. La classe 0, sont des personnes qui ont remboursé leur empreint de crédit et la classe 1 correspond aux personnes qui ont fait un défaut de paiement (endettés envers l'institution financière). La classe 0 représente 90% de l'échantillon et 10% de données seulement représentant la classe 1.

Ce déséquilibre est la raison d'une mauvaise prédictibilité de la classe minoritaire par le modèle de classification entrainé. La méthode SMOTE (Synthetic Minority Oversampling Technique) a été appliquée sur le jeu de données d'entrainement afin de remédier à cela.

Cette technique crée artificiellement un nouvel échantillon, en calculant des points proches de l'espace vectoriel d'éléments de la classe minoritaire, ensuite elle égalise les taux des deux classes.

Réduction du nombre de variables pour choisir les plus pertinentes : c'est une étape importante du prétraitement des données.

En effet cela permet d'améliorer les résultats de prédiction et facilite l'étape d'entraînement du modèle.

Les méthodes suivantes ont été appliquées :

→ VarianceThreshold - en supprimant les variables constantes
→ Corrélation – variables trop corrélés ont été sont éliminées

→ Information gain - mutual information : mesure si les variables sont informatives, on élimine les variables à basse de variance.

→ Facteur d'inflation de la variance : VIF évalue si les facteurs sont corrélés les uns aux autres (multi-colinéarité), ce qui pourrait influencer les autres facteurs et réduire la fiabilité du modèle

4) Modélisation :

5 Algorithmes de classification ont été testés et entrainés :

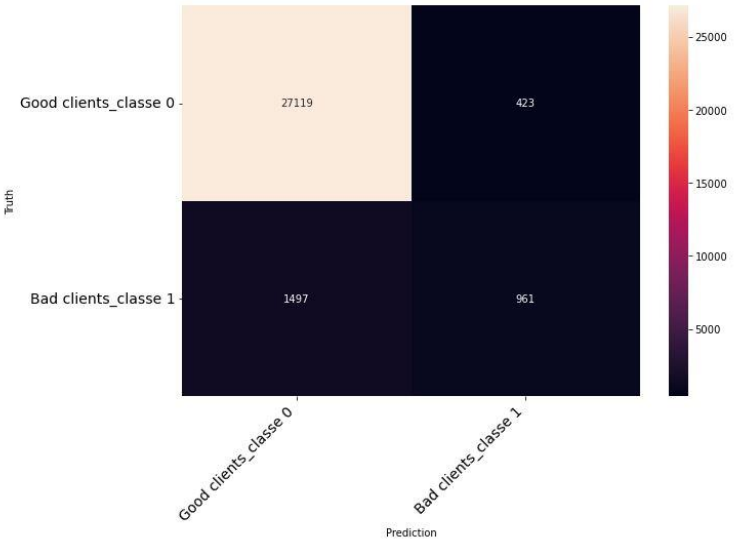
- Logistic Regression
- Random Forest
- XGBoost (Extreme Gradient Boosting for classification)
- SVC (C-Support Vector Classification)
- LightGBM(Light Gradient Boosting Machine)

Le modèle le plus performant quant aux résultats obtenus a été choisi, à savoir LightGBM.

Résultats du modèle LightGBM : Les prédictions (faites sur le jeu de données test déséquilibré). Nous avons la matrice de confusion et la courbe ROC AUC.

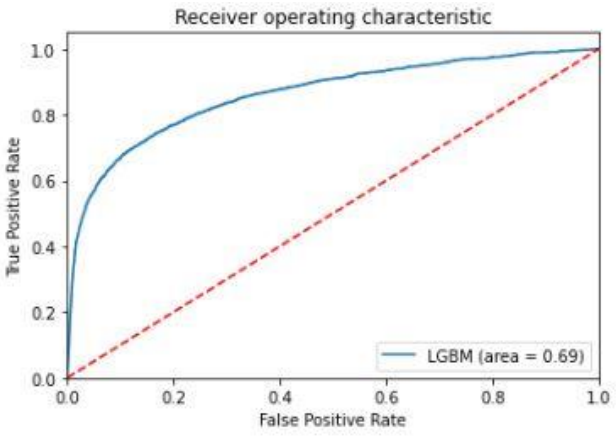
La matrice de confusion : met en évidence les prédictions pour les deux classes (0 et 1).

	Prédiction **Positive : 0**	Prédiction **Négative :1**
Vérité **Positive : 0**	**TP** les vrais payeurs	**FN** les faux non-payeurs
Vérité **Négative : 1**	**FP** les faux payeurs	**TN** les vrais non-payeurs



ROC-AUC score : La fonction d'efficacité du récepteur ROC (Receiver operating characteristic) est une courbe de probabilité.

AUC mesure le degré de séparabilité des classes. Il varie de 0 à 1. Une valeur de 0 équivaut à une mauvaise séparation des classes et, la valeur 1 correspond à une bonne séparation des classes.



GridSearchCV a été appliqué pour trouver les meilleurs hyperparamètres pour le modèle LightGBM.

Nous avons un score de ROC_AUC de 0.69 après optimisation des hyperparamètres.

Réduction du coût bancaire

1) Approche technique

Les Faux Positifs = FP, sont les clients qui ont été prédits comme des potentiels mauvais payeurs, mais qui sont en réalité de bons clients.

Les Faux négatifs = FN sont les clients approuvés par le modèle, mais qui sont en réalité en défaut de paiement. Dans notre cas, c'est cette valeur qu'il faut essayer de diminuer. Car cela représente un coût plus élevé pour la banque que d'attribuer un crédit qui ne sera pas remboursé.

C'est par une approche métier qu'on réalise l'optimisation du modèle.

2) Approche métier (intuitive)

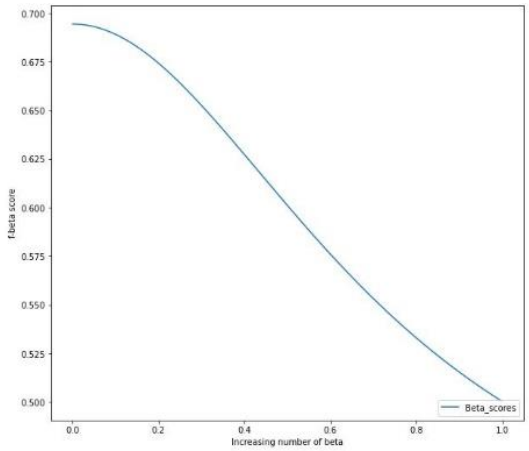
L'approche métier : on utilise les probabilités de prédictions en sortie du modèle (model.predict_proba(X_test)). Le modèle choisi automatiquement un seuil de 0.5 pour distribuer les classe (1= mauvais payeur ou 0= bon payeur) aux clients qu'on lui présente (input). Nous pouvons rechercher ce seuil en analysant les probabilités de prédictions en fonction des seuils définies au préalable.

Métriques d'évaluation du coût bancaire

1) Approche technique

F beta scores : nous avons vu plus haut que le calcul de Rappel permet de savoir à quel point notre modèle est capable de révéler les clients non-payeurs. Autrement dit, nous cherchons à minimiser les FP.

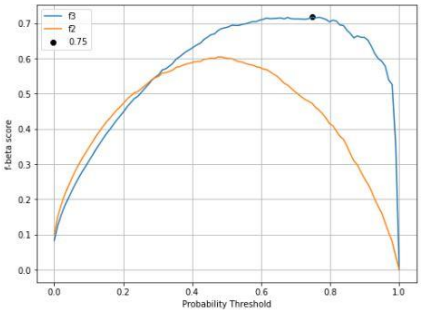
La figure ci-dessus est la variation de f beta scores en fonction de beta. Pour améliorer le score de Rappel qui nous intéresse, il faut calculer ce score en choisissant une valeur de beta petit.



Précision : % des instances réellement positives parmi toutes les prédictions classifiées positives TP/TP+FP	Rappel : % des prédictions positives parmi toutes les instances positives TP/(TP+FN)	Fbeta-score : compromis entre précision et rappelle 1+beta^2*(Rappel * Précision) / (Rappel + Précision)

Le choix du seuil de probabilité optimal de distribution des classes (0 et 1), est fait en analysant l'évolution des f beta scores en fonction des différents seuils (voir graphique ci-dessus).

Le graphique ci-dessous représente l'évolution du score f beta pour deux valeurs de beta (f3 = 0.1 et f2 =0.5). Le meilleur score est de 0.717 pour un seuil de probabilité = 0.75.



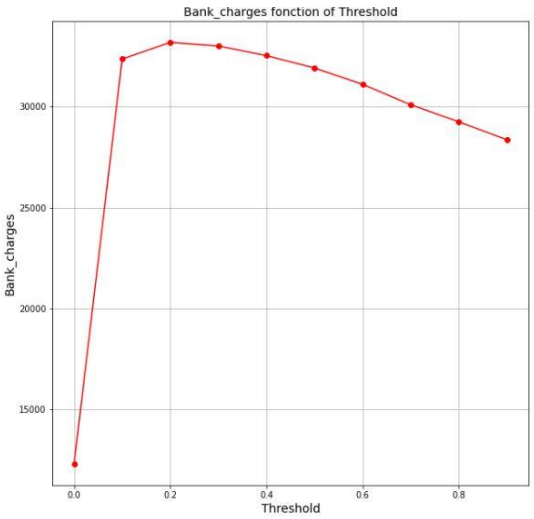
2) Approche métier

Evaluation métier : calcul de coût bancaire.

Le but ici est de montrer l'importance de réduire les valeurs de FP. En augmentant de 5 fois les valeurs des FP nous avons un aperçu des coûts que les FP trop élevés peuvent représenter pour la banque.

La figure ci- dessus est une courbe qui représente l'augmentation des coûts bancaires en fonction du seuil de classement (pour classe 0 ou 1).

Nous pouvons constater (comme démontré précédemment avec les scores f beta) que le seuil de probabilité optimal s'approche de la valeur 0.75.



L'interprétabilité du modèle

Il est difficile d'expliquer explicitement les prédictions sans rentrer dans les détails techniques et les calculs. Expliquer simplement comment les résultats ont été obtenus.

Pour rendre le modèle plus compréhensible aux personnels d'une institution telle qu'une banque, nous devons interpréter les modèles autrement.

1) Interprétabilité globale avec l'algorithme Permutation Importance de Sklearn

C'est l'explication des causes de la décision d'une prédiction du modèle, qui va dans le sens de la transparence recherchée par les institutions de crédit.

Le but est de mesurer l'influence de chaque variable (feature) dans un modèle. Elle est basée sur l'augmentation de l'erreur de prédiction du modèle après perturbation (permutation) des valeurs d'une variable. Une variable est importante si après sa perturbation l'erreur de prédiction augmente significativement. Et si une variable n'est pas importante sa permutation n'induit pas de changement remarquable à la prédiction.

2) Interprétabilité locale avec LIME de Sklearn

Nous appliquons la technique Lime (Local Interpretable Model agnostic Explanations) qui est une librairie Python. Elle prend en entrée un modèle et génère des explications concernant la contribution des variables sur les résultats de prédiction du l'instance (client) concerné.

Limites et améliorations possibles

L'interprétabilité du modèle reste une étape sensible, l'explication fournie par l'approche Lime est instable car dépend des échantillons de données. Elle peut fournir une explication correcte pour un certain nombre de données alors que pour d'autres l'interprétation peut être fausse.

Pourquoi ? Parce que lime est sensible aux perturbations qui peuvent affecter les données.

Pour ce projet nous avons choisi d'entrainer des modèles basés sur des algorithmes de classification traditionnel de Machine Learning, et au cours du processus d'entrainement nous avons vu que cette approche est couteuse en temps. Quant à la performance de la prédiction, elle dépend de plusieurs facteurs : la qualité du traitement des données, les choix des variables, la connaissance du domaine d'application.

Le Deep Learning permet à la fois de simplifier les premières étapes de prétraitement des données, mais aussi de tenir compte d'un nombre de variable beaucoup plus important pour la construction du modèle. L'utilisation Réseaux de Neurones pourrais également diminuer le temps d'entrainement du modèle.

Mais cette approche vient avec ses limites, celle de l'interprétabilité, qui est plus difficile, voire impossibles à réaliser.