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 aura pour objectif de calculer la probabilité pour un client donné de rembourser le crédit qu'il a emprunté. If faut donc construire un modèle de classification.

2) Compréhension du jeu de données

Le jeu de données est accessible 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

Dans cette partie nous avons des étapes de prétraitement des données. La plupart des variables sont créés 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 pour les variables catégorielles**.**

Les différents datasets ont été sont mergés.

Pour pallier le déficit de classes des clients ayant fait un défaut de paiement qui représentent moins de 10% de l'ensemble de la donnée,
Les classes ont été équilibrés avec la librairie Imblearn par la méthode Oversampling. Cette dernière, consiste à dupliquer les données de la classe minoritaire et de les redistribuer aléatoirement, ensuite elle égalise les taux des deux classes.

Pour réduire le nombre de variables et en avoir celles les plus intéressantes, nous avons effectué une sélection de variables :
→ 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

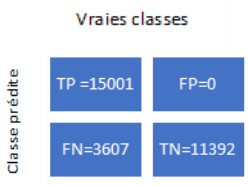
→ Importance de la caractéristique de permutation

4) Modélisation : Entrainement de 4 modèles avec

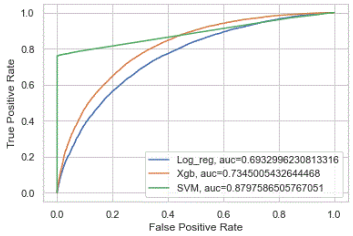
- Logistic Regression
- Random Forest
- XGBoost (Extreme Gradient Boosting for classification)
- SVC (C-Support Vector Classification)

Le modèle retenu est le SVC avec les résultats suivants :

La matrice de confusion



La courbe ROC_AUC (AUC_score=88%)



Réduction du coût bancaire

1) Approche technique

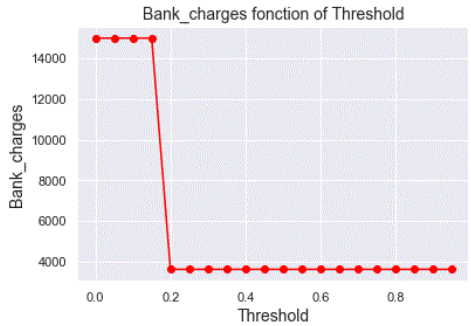
L'approche technique : consiste à recourir à des algorithmes d'optimisation d'hyperparamètres des modèles. RandomizedSearchCV en est un, il recherche les meilleurs hyperparamètres pour un modèle ML et augment sa performance quant à la prédiction.

Les Faux positives = FP (dans notre cas =0) est le nombre de clients qui ont été prédits comme des potentiels mauvais payeurs, mais qui sont en réalité de bons clients.

Les Faux négatives = 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. C'est par une approche métier qu'on réalise l'optimisation du modèle.

2) Approche métier

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 payer ou 0= bon payeurs) 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. Dans notre cas le nouveau seuil calculé = 0.2.



Métriques d'évaluation du coût bancaire

1) Approche technique

Evaluation technique : par la fonction d'efficacité du récepteur ROC (Receiver operating characteristic) est une courbe de probabilité et AUC mesure le degré de séparabilité des classes. Il varie de 0 à 1. Une valeur de 0 = mauvaise séparation des classes et 1 = bonne séparation des classes.

2) Approche métier

Evaluation métier : calcul de coût bancaire. Le but ici est de montrer l'importance de réduire les valeurs de faux négatifs. En maximisant les valeurs des faut négatifs nous avons un aperçu des coûts que les FN trop élevés peuvent représenter pour la banque.

L'interprétabilité du modèle

Expliquer les mécanismes derrière un modèle ML et les prédictions aux professionnels qui ne sont pas du domaine de ML n'est pas envisageable avec les métriques tels que l'Accuracy ou ROC_AUC etc…

Pour rendre le modèle plus compréhensible aux personnels d'une institution telle qu'une banque qui ne sont pas des professionnels de ML, 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 modèle 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 de 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 dans certains cas impossibles à réaliser.