Projet 7 - Note methodologique

Mitul VYAS

Août 2022

1 Introduction

Dans la cadre du projet 7 de la formation Data Scientist d'Open Classroom, il est demandé d'implémenter un modèle de scoring pour la société financière "prêt à dépenser". Ce modèle de scoring est appliqué à des clients ayant peu ou pas du tout d'historique de prêt. L'enjeu du modèle est d'attribuer à chaque client de la société une probabilité de défaut. Nous déterminerons un seuil métier grâce auquel nous pourrons savoir si le crédit est accordé ou non. Afin de présenter notre analyse, un dashboard interactif sera présenté au manager Michael. Ce dashboard donnera les renseignements pour un client et essayera de répondre à la question : "quels sont les facteurs ayant permis ou pas l'octroi du crédit ?"; "quels sont les renseignements pertinents au client ?".

Le fichier mis à disposition pour mettre en place notre modèle est composé de 307 000 clients et 121 variables. (sexe, nombre d'enfant, ses revenus, le montant du crédit).

2 Démarche de modélisation

Pour mettre en place notre modèle, nous nous sommes appuyé sur un kernel kaggle. Ce kernel permet d'effectuer le travail de preprocessing de dataset. Comme tout modèle de machine learning, nous devons séparer notre jeux de données en deux parties.

- 75% de notre jeu de données a été utilisé comme données d'entrainement. Ces individus ont été splitté en plusieurs folds (5) de sorte à optimiser les hyperparamètres et éviter le risque d'overfitting.
- 25% de jeu de test, afin de tester les performances de notre modèle.

Nous traitons dans ce projet un problème de classification binaire supervisé. Toutefois, la variable cible que nous souhaitons prédire est composé de 92% de valeur 0 (le crédit est accordé) contre 8% de valeur 1 (le crédit n'est pas accordé). Afin de pallier ce défaut de la donnée cible. Nous allons utiliser la technique SMOTE (Synthetic Minority Oversampling Technique), qui va équilibrer notre jeu de données cible en y ajoutant des variables synthétiques de la classe de

donnée minoritaire. Ces variables synthétiques étant basé sur les observations existantes.

Nous pouvions également utiliser le paramètre *class_weight* présent dans les modèles afin d'ajouter plus de poids à la classe minoritaire et moins de poids à la classe majoritaire.

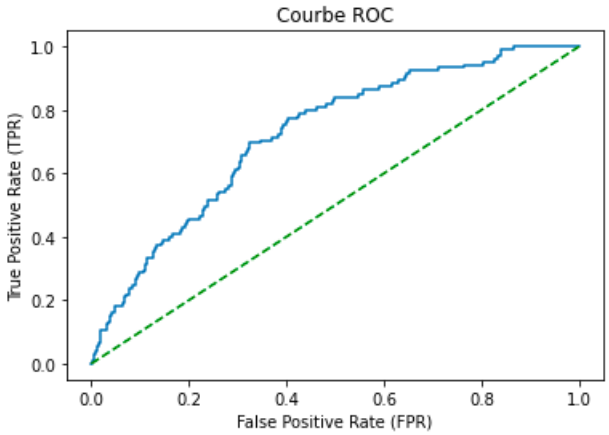
Concernant les modèles de machine leraning utilisés, nous avons choisi de tester 6 modèles:

- Random Forest Classifier
- K nearest Neighbors Classifier
- Gradient Boosting Classifier (XGboost)
- Light Gradient Boosting Machine (LGBM) Classifier
- Support Vector Classifier (SVC)
- Logistic Regression

Pour chacun de ces modèles nous avons optimisé les hyperparamètres grâce à la méthode GridSearchCV utilisant 5 "folds". Une fois les meilleurs hypérparamètres établis, on calcule les scores de chaque modèle sur le jeu de validation. Nous obtenons les résultats suivants:

	lgbm_pred	knn_pred	rf_pred	loglin_pred	xgboost_pred	svc_pred
scores	0.722217	0.480611	0.710665	0.641634	0.697709	0.651426

Le modèle que nous avons retenu est LGBM Classifier puisqu'il offre de meilleurs scores pour le jeu de validation.

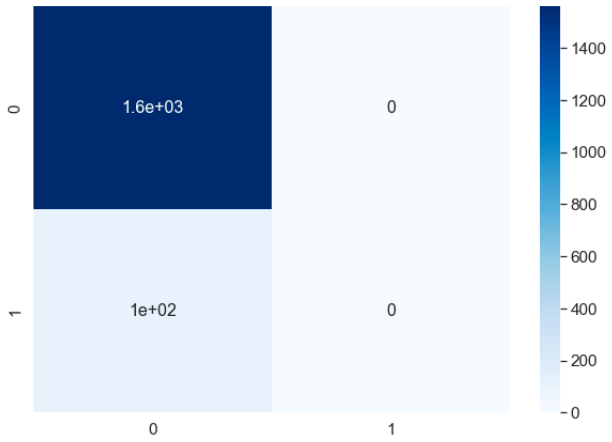



```
|: # On retrouve le score Auc de performance du modèle lgbm.

   print(auc_roc)

   # Nous pouvons dire que le modèle est assez performant.
```

```
0.7222171379762667
```



2.1 Fonction de coût métier

Nous souhaitons déterminer le seuil à partir duquel la probabilité calculée se transforme en la valeur 1. En effet, le résultat de la prédiction est un vecteur de qui a chaque client lui associe sa probabilité de rembourser le crédit. La problématique métier stipule que nous devons faire d'avantage attention aux faux positive plutôt qu'au faux négatif. Accorder un crédit à un client non

solvable est plus couteux pour la banque que le manque à gagner d'un client dont le crédit est refusé bien que celui-ci soit en capacité de le rembourser. Nous allons donc postuler qu'un faux négatif coûte 10 fois plus cher qu'un faux positif.

$$FC(x) = 10 \times FN(x) + FP(x)| \quad x \in [0, 1] \tag{1}$$

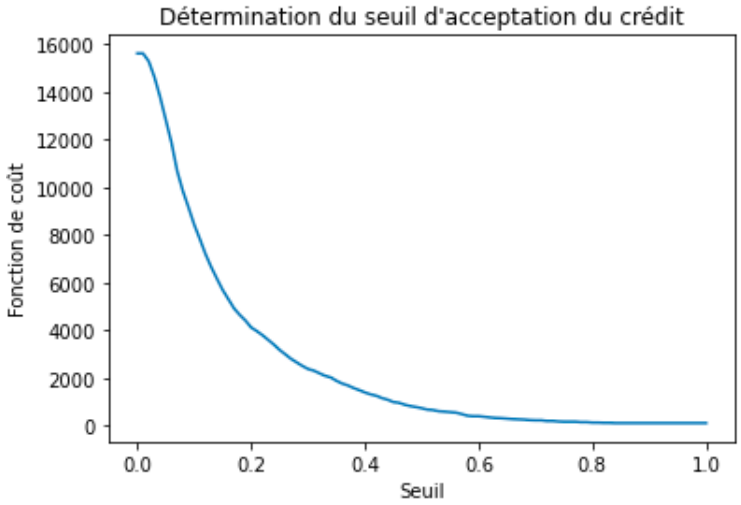
Avec

- FC : fonction de coût
- FP : faux positif
- FN : faux négatif

Pour obtenir le seuil permettant de décider si oui ou non le crédit est accordé, nous traçons la courbe FC en de différentes valeurs de seuils (compris entre 0 et 1) et nous choisissons le minimum de cette courbe.
Il vient,

$$Seuil = argmin\{FC(x)\} \quad |x \in [0, 1] \tag{2}$$

Ainsi le seuil obtenu va permettre de minimiser le recall (taux d'individus positifs détectés par le modèle) et de maximiser la précision (taux de prédictions correctes parmi les prédictions positives.



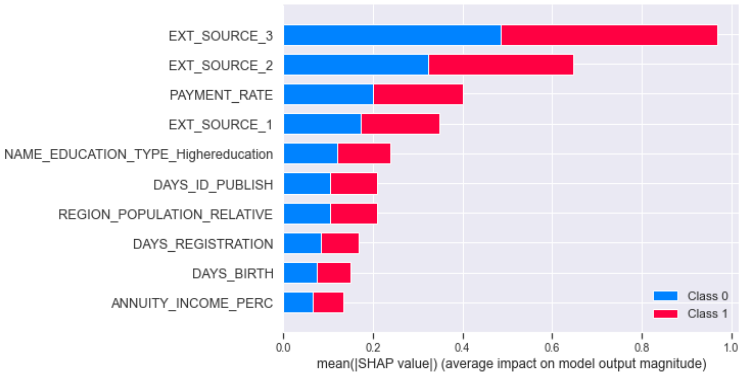
2.2 Interpretation locale et globale du modèle

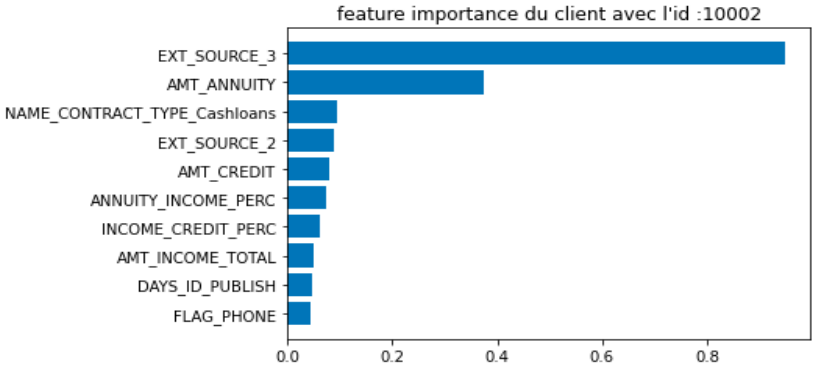
Les utilisateurs du modèle doivent être capables d'expliquer les caractéristiques spécifiques d'un client (caractéristiques locales) et de les comparer avec ceux d'autres clients (caractéristiques globales). Cela va permettre de comprendre

pourquoi un client plutôt qu'un autre s'est vue son crédit accordé. Il est donc nécessaire de comprendre quels sont les variables du modèle qui jouent un rôle important dans l'attribution d'un crédit. Il s'agit donc d'évaluer les "feature importance" du jeux de données mais aussi de chaque client. Pour ce faire nous avions deux choix:

- utiliser la méthode "feature_importances_" de certains modèles (XGboost, Random Forest).
- utiliser la librairie Shap

Nous avons dans le projet choisit d'utiliser la libraire Shap car elle permet le calcul de feature importance globale indépendante de l'algorithme utilisé. En effet, lorsque l'on choisit la méthode "feature_importances_", on obtiens des résultats tributaires de l'algorithme utilisé.





2.3 Limites et améliorations possibles

Le choix des hyperparamètres est une limite, car si l'on ne possède pas d'ordinateur suffisamment puissant, il sera difficile d'obtenir les choix des meilleurs hyperparamètres dans des temps raisonnables. D'autre part, il est possible avec le métier de se mettre d'accord avec une meilleure fonction de coût pour déterminer le seuil. Il est aussi possible d'utiliser de meilleurs techniques d'imputation de variables qui se baseront par exemple sur les distributions de chaque feature. Enfin, une technique de réduction de dimensions aurait pu permettre de supprimer les corrélations entre les variables et permettre de ne garder que les variables les plus pertinentes.