NOTE MÉTHODOLOGIQUE

Projet n°7 : Implémentez un modèle de scoring

Table des matières

1 Introduction

L'objectif du projet est de mettre en œuvre un outil de calcul de la probabilité qu'un client rembourse son crédit afin de classifier sa demande en crédit accordé ou refusé.

Un modèle de « Machine Learning » qui une technique de programmation informatique qui utilise des probabilités statistiques pour donner aux ordinateurs la capacité d'apprendre par eux-mêmes sans programmation explicite sera mis en place pour atteindre cet objectif et détaillé dans cette note méthodologique.

2 Prétraitement des données et choix du modèle

La méthodologie du choix du modèle consiste après un prétraitement des données, d'effectuer des prédictions via les différents modèles et d'en retenir celui offrant les meilleures performances.

2.1 Prétraitement du jeu de données

Les algorithmes de machine learning apprennent des données que nous leur fournissons. Ainsi, si ces données sont de mauvaise qualité, nous aurons de mauvais modèles car ils apprendront de ce qu'ils verront dans les données. Il est donc très important de bien préparer nos données avant l'implémentation des modèles.

2.2 Traitement des données aberrantes et manquantes et feature engineering

Un « kernel » qui est une fonction est choisie pour effectuer les opérations de fusions des différentes tables de la base de données et pour la partie feature engineering qui consiste à créer des variables intéressantes et pertinentes pour notre étude.

La méthode interquartile qui consiste de tenir compte des seules observations entre le troisième et premier quartiles (valeurs qui partagent la distribution en quatre parties égales) sera utilisée pour le nettoyable des valeurs aberrantes de notre jeu de données. Les données manquantes des variables quant à elles ont été remplacées par la valeur médiane de la variable en question.

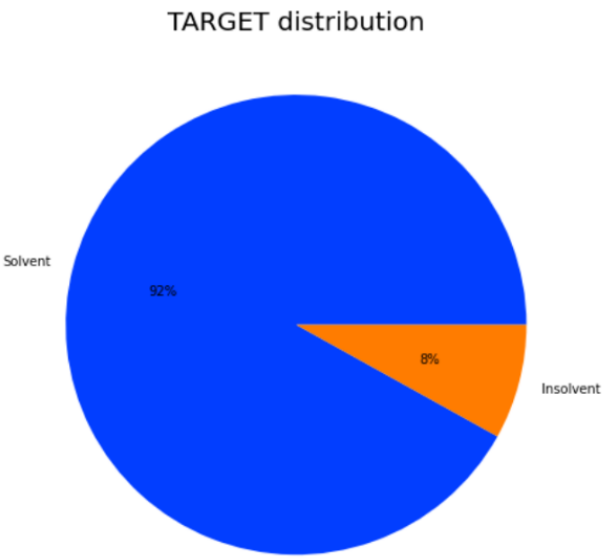
2.3 Exploration du jeu de données

Une rapide analyse exploration du jeu de données a été effectuée pour mieux comprendre les clients à l'étude.

Il en ressort entre autres que la plupart des clients sont solvables c'est-à-dire que leurs prêts ont été remboursés dans les délais préalablement fixés.

Les proportions des données entre clients solvables et non solvables sont de l'ordre de 92% contre 8% ce qui donne une répartition des deux classes assez déséquilibrée ce qui peut fausser grandement l'interprétation des résultats fournies par les modèles.

Il sera donc intéressant de travailler sur cet suréchantillonnage en ajustant la distribution des classes de manière à avoir une répartition plus égalitaire.



2.4 Sélection du meilleur modèle

Une comparaison des performances des différents modèles est effectuée afin d'en choisir le meilleur.

2.4.1 Standardisation et rééchantillonnage

La standardisation qui consiste à mettre les variables quantitatives à la même échelle est effectuée par l'intermédiaire de MinMaxScaler et le rééchantillonnage de nos données est effectué via SMOTE dont le principe est de générer les nouveaux échantillons en combinant les données de la classe minoritaire avec celles de leurs voisins proches.

Afin d'entrainer les modèles, nous séparons nos données en entrainement et test avec une proportion de 30% pour le test.

2.4.2 Comparaison des différents modèles

La comparaison des performances des modèles de classification DummyClassifier, Logistic Regression, GaussianNB, Random Forest, Xgboost, AdaBoostClassifier, Multi-layer Perceptron, LGBM est effectuée en calculant les scores de l'aire sous la courbe ROC qui une courbe sensibilité(capacité à donner un résultat positif lorsqu'une hypothèse est vérifiée)/spécificité(capacité d'un test à donner un résultat négatif lorsque l'hypothèse n'est pas vérifiée) au travers d'une validation croisée (méthode statistique qui permet d'évaluer les performances des modèles d'apprentissage automatique) sur les données de d'entrainement.

Il ressort que les trois modèles offrant les meilleures performances sont :

- LGBMClassifier
- XGBClassifier
- Logistic Regression

2.4.3 Réduction de la dimension

Les données contiennent 796 variables dont beaucoup peuvent ne pas contenir d'informations utiles. Nous réduisons la dimension via RFECV de Scikit-learn qui applique une validation croisée pour trouver l'ensemble des variables optimales qui maximise nos performances. Nous utilisons pour cette recherche de variables optimales le classifier LGBMClassifier car offrant les meilleures performances.

2.4.4 Optimisation des hyperparamètres d'un point de vue technique

L'optimisation des scores l'aire sous la courbe ROC des trois modèles sélectionnés est effectué via une GridsearchCV qui est une combinaison des valeurs des différents hyperparamètres des modèles afin d'en extraire la combinaison donnant le meilleur score.

2.4.5 Choix du meilleur modèle

En faisant un compromis entre le meilleur score AUC(aire sous la courbe ROC) et temps d'entrainement et de prédiction, il en ressort que pour le jeu de donnée de notre étude, le modèle retenu pour la classification est :

- LGBMClassifier

LGBM , abréviation de Light Gradient Boosting Machine est une infrastructure logicielle d'amplification de gradient basé sur des algorithmes d'arbre de décision et utilisé pour le classement, la classification et d'autres tâches d'apprentissage automatique.

3 Optimisation du modèle d'un point de vue métier

L'approche métier consiste à optimiser le modèle relatif à une problématique bien précise. Dans le cas à l'étude il s'agit de maximiser les gains pour l'entreprise en évitant d'accorder les prêts aux clients non solvables.

3.1 Fonction coût métier

Une fonction coût bien précise a été développée dans l'objectif de maximiser les gains obtenus par validation ou non des demandes de prêts bancaires.

S'agissant d'un problème de classification binaire, l'étiquette est soit égale à 0 (négatif, solvable) soit égale à 1 (positif, non solvable). Il existe donc 4 combinaisons possibles :

- TN : True Negatif, le modèle prédit 0 et la valeur réelle est 0
- TP : True Positif, le modèle prédit 1 et la valeur réelle est bien de 1
- FN : False Negatif, le modèle prédit 0 alors que la valeur réelle est bien de 0
- FP : False Positif, le modèle prédit 1 alors que la valeur réelle est de 0

Le modèle peut donc se tromper de deux manières différentes, soit en prédisant positif alors que l'individu est négatif (False Positif) soit en prédisant négatif alors que l'individu est positif. En revanche, la perte d'argent est plus conséquente pour un FN (prêt accordé alors que le client n'est pas solvable) qu'un manque à gagner pour un FP (prêt non accordé alors que le client est solvable). Une fonction coût :
J = 10*FN + FP a donc été créée pour tenir compte de l'importance relative de chaque erreur.

3.2 Algorithme d'optimisation

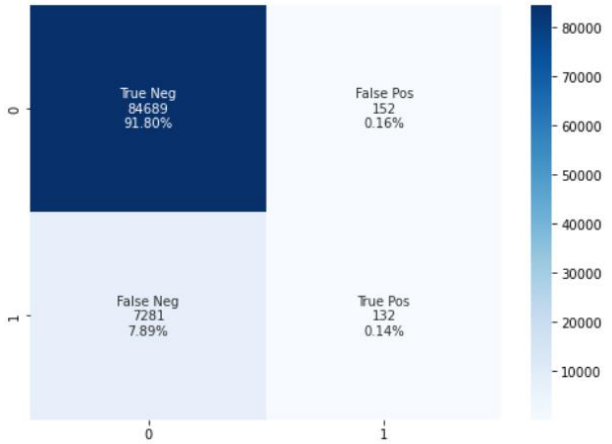
Pour déterminer quelle combinaison d'hyperparamètres donne les meilleurs résultats c'est à dire minimiser la fonction coût créée, l'algorithme d'optimisation « hyperopt » a été utilisé.

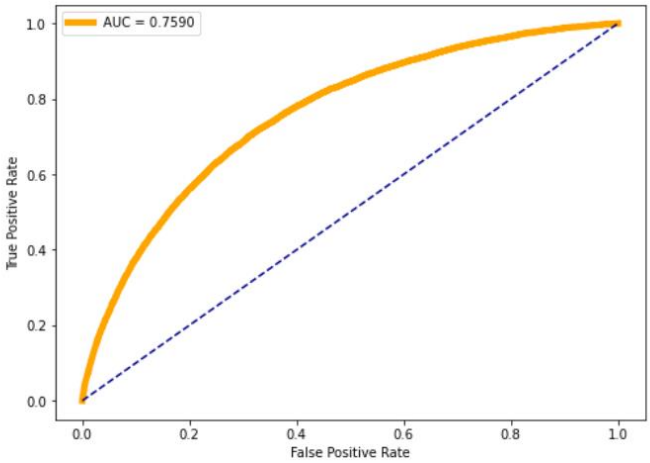
L'algorithme « hyperopt » utilise une approche Bayésienne pour déterminer les meilleurs paramètres d'une fonction. A chaque étape, il essaye de construire un modèle probabiliste de la fonction et choisi les paramètres les plus promettant pour la prochaine étape.

3.3 Métrique d'évaluation

Le modèle retourne un score entre 0 et 1 et par défaut il attribue la classe 1 lorsque le score est supérieur à 0.5 et 0 sinon. Il a été décidé d'optimiser ce seuil qui permet de définir si un client est solvable ou non. Le seuil optimal déterminé par « hyperopt » est de 0.075, c'est-à-dire quand le score retourné par le modèle est supérieur au seuil optimal, le client est dit non solvable.

La matrice de confusion optimisée via la fonction coût est la suivante :



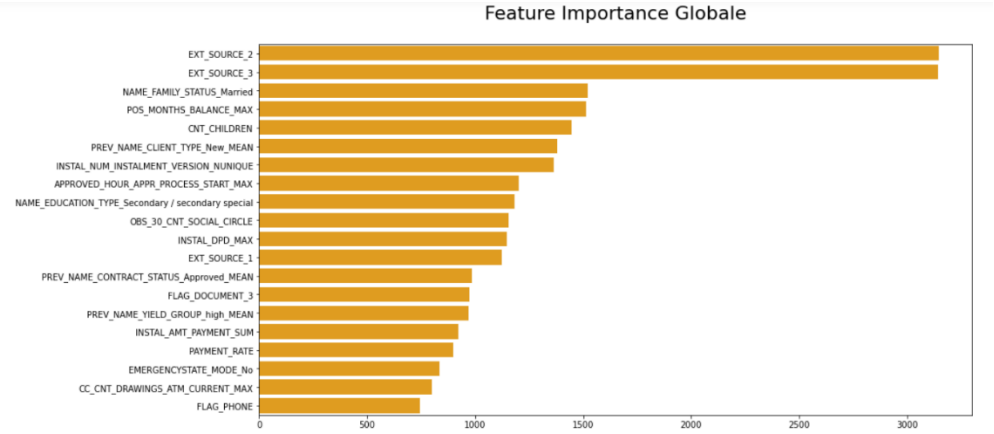


4 Interprétabilité globale et locale du modèle

L'interprétabilité consiste en la capacité d'expliquer ou de présenter les résultats fournis par le modèle dans des termes humainement compréhensibles.

4.1 Interprétabilité globale

Le modèle LGBM fournie de manière globale les variables qui ont eu plus d'importance dans le résultat de prédiction. Il en ressort que les variables EXT_SOURCE_2 et EXT_SOURCE_3 qui sont des scores normalisés provenant d'autres données sont celles ayant plus d'impact dans le résultat de prédiction.



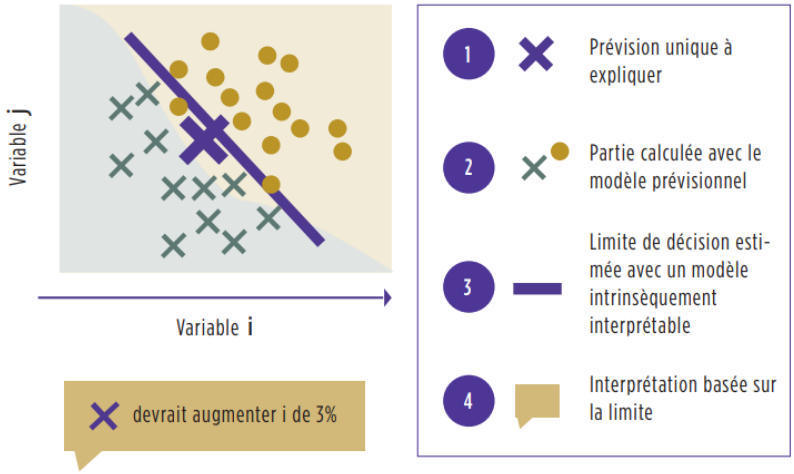
4.2 Interprétabilité locale

La librairie LIME a été utilisée pour l'interprétabilité locale liée au résultat de prédiction d'un client spécifique.

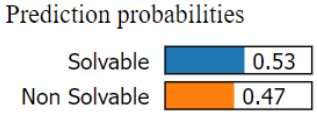
LIME permet de déterminer pour chaque observation les variables qui ont le plus d'impact dans le résultat de la prédiction.

Lime fonctionne en 3 étapes :

- Lime génère aléatoirement des nouveaux individus fictifs proches de l'individu sélectionné
- Lime calcule la prédiction en fonction du modèle
- Lime calcule un modèle linéaire (interprétable) sur ces nouvelles données

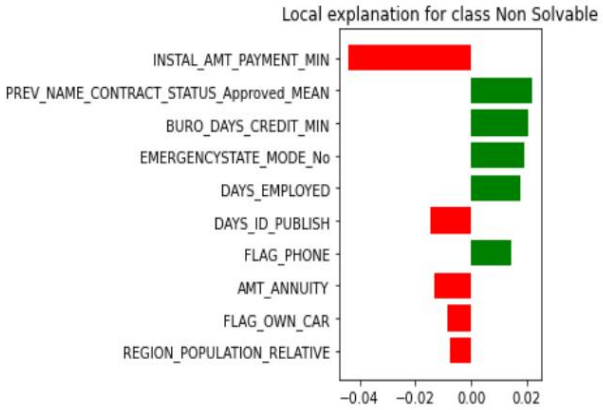


Exemple pour un client donné



La non solvabilité du client '62255' est supérieur au seuil de solvabilité de 0.075, donc le client est non solvable.

La variable qui joue le plus en sa défaveur est 'INSTAL_AMT_PAYMENT_MIN' relatif à ses payements. À l'inverse la variable qui joue le plus en sa faveur est 'PREV_NAME_CONTRACT_Aprroved_MEAN', relatif aux anciens contrats approuvées.



5 Limites et les améliorations possibles

Des limites ont été relevées dans l'implémentation du modèle et des améliorations de ce dernier sont possibles**.**

5.1 Performance du serveur

Le temps d'entrainement et prédiction a été un critère discriminant dans le choix du modèle. Un serveur plus conséquent en taille mémoire et en CPU permettrait des tester des modèles prometteurs comme XGBClassifier et CatBoost.

5.2 Feature engineering

Un autre axe d'amélioration serait de discuter avec les équipes métiers du domaine afin de déterminer les variables encore plus pertinentes au vu de leur expérience dans leur choix d'accorder un prêt ou non à un client donné.

5.3 Fonction coût métier

Les coefficients 10 et 1 affectés à FN et FP dans la fonction coût l'ont été de manière arbitraire sans connaissance des poids réels entre la perte d'argent un FN (prêt accordé alors que le client n'est pas solvable) et un manque à gagner pour un FP (prêt non accordé alors que le client est solvable).

Une définition ces coefficients en concert avec l'entreprise permettrait d'améliorer le gain de cette dernière.