PONCET RONAN

Parcours Data Scientist

Projet 7 : Implémentez un modèle de scoring

Note méthodologique

Cette note constitue l'un de livrable du projet 7 du parcours de Data Scientist

L'entreprise "Prêt à dépenser" nous demande de développer un outil de classification pour mettre en œuvre un outil de "scoring credit" afin de calculer la probabilité qu'un client rembourse son prêt. Puis d'estimer si le crédit doit être accepté où refuser en fonction de la probabilité obtenue.

Présentation et traitement du jeu de données

Nous avons à notre disposition un jeu de données composé de huit Dataframe qui rassemblent 111 features pour 307 000 clients différents.

La variable à prédire, noté 'TARGET' vaut 1 si le client a eu un incident de paiement et 0 sinon. Cependant celle-ci est inéquitablement reparti dans le jeu de données : 92% de clients appartiennent à la catégorie 0 contre 8% pour la catégorie 1

Features enginering:

Pour le prétraitement des données nous utiliserons le Kernel kaggle : https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script

Celui-ci consiste notamment à effectuer un OneHotencoding sur les variables catégorielles et à extraire les valeurs statistiques principales des variables numériques(mean, max , min ,var ,sum)

Nous obtenons un jeu de donnée composé de 799 features auxquels on retirer toutes les variables possédant plus de 80% de nan. Nous imputerons ensuite les valeurs nan restante par 0. Le jeu de données final est composé de 507 features.

Selection du modèle :

Pour choisir le modèle à utiliser pour notre prédiction nous entraineront quatre modèles :

- DecisionTreeClassifier
- RandomForestClassifer
- Xgboost
- Lgbm

Pour chaque modèle nous choisirons les hyperparamètres optimaux avec une BayessearchCV effectué sur 50 itérations.

De plus pour traiter le problème de balance des classes évoqué précédemment nous utiliserons deux méthodes :

- En ajoutant le paramètre Class_weight lors de notre BayessearchCV
- En effectuant un OverSampling a l'aide de Smote()

Optimisation du modèle choisit:

Une fois la sélection de modèles effectuée, nous tenterons d'améliorer les performances du modèle en nous analysants les features les plus importantes avec Shap, Puis nous entraineront le modèle en substituant au jeu de données le 50, 100, 150, 200 features les moins impactantes.

Métrique d'évaluation, fonction cout et algorithme d'optimisation

Algorithme d'obtimisation:

La méthode BayessearchCV utilisé pour optimiser les hyperparamètres consiste à entrainer nous données en Cross Validation pour différents hyperparamètres et en choisissant les paramètres à tester a chaque itération par optimisation bayésienne.

Metrique d'evaluation:

La métrique d'évaluation utilisé lors de la sélection d'hyperparamètres est le score roc_auc qui indique l'air de la courbe auc du modèle

Cette courbe prend le threshold appliqué aux probabilités prédites par le modèle en abcysses et le rapport (Taux de vrai positif)/(Taux de faux positif) en ordonnée

Fonction cout_metier

Pour notre fonction cout metier nous cherchons à ce que les clients en défault ne soit pas prédit dans la catégorie 0 ce qui représenterais une perte direct. Nous voulons donc maximiser le nombre de vrai positif et minimiser le nombre de vrai positif. Une métrique appropriée pour cela est le recall.

Cependant les faux positifs representent aussi une perte potentielle et donc la métrique de précision ne doit pas être négligé.

Il existe une famille de métrique permettant d'optimiser la précison et le recall en donnant plus d'importances a l'un des deux : Les F-Beta score.

Cependant il est necessaire de preciser un coefficient Beta qui permet de regler le rapport recherché entre les deux métriques et les données du projet ne

permettent pas de quantifier l'importance des faux positifs et négatifs : on fixera donc arbitrairement le coefficient Beta a 2 ce qui maximise légerement le recall

Interprétabilité globale et locale du modèle

Le modèle étant destiné à des équipes opérationnelles devant être en mesure d'expliquer les décisions de l'algorithme a des clients réels nous utiliserons la librairie Shap afin de mettre à leurs dispositions différentes informations :

- Un summary plot représentant l'importance globale des features dans les prédictions du modèle

- Pour chaque client : Un waterfall plot représentant l'importance locale des features pour la prédiction de ce client

-Pour chaque variable : un dépendance plot de la variable avec laquelle elle est le plus corrélée afin de pouvoir expliquer l'influence de cette variable dans la prédiction

Limites et les améliorations possibles:

La principale limite et amélioration possible du modèle se trouve dans sa fonction cout métier : En effet il serait nécessaire de pouvoir estimer la perte que représente un faux négatif ou un faux positif en fonction du montant de leurs crédits, cela permettrait de choisir un coefficient beta minimisant les pertes réelles de notre client.

Une seconde amélioration pourrait se trouver dans le traitement des données, d'autre préprocessings pourrait être essayer pour tenter d'obtenir de meilleurs résultats