Projet 7 : Implémenter un modèle de scoring
Note méthodologique

- 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

I. Contexte :

L'entreprise Prêt à dépenser souhaite mettre en œuvre un outil de "scoring crédit" pour calculer la probabilité qu'un client rembourse son crédit, puis classifie la demande en crédit accordé ou refusé. Elle souhaite donc développer un algorithme de classification en s'appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

L'entreprise souhaite également développer un dashboard interactif pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d'octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

II. La méthodologie d'entraînement du modèle :

Le jeu de données initial contient 122 variables et 307511 observations . Après une première exploration des données avec la librairie : Pandas Profiling Report . Un nettoyage du jeu de données a été fait en se basant sur le taux de remplissage .On a supprimé les variables ayant plus de 40% de valeurs manquantes . La nouvelle dimension est de 73 variables : réparties entre 12 variables qualitatives et 61 variables quantitatives. De nouvelles variables métiers ont été créés on se basant sur le kernel suivant : Kernel

Pour l'encodage des variables qualitatives . On a utilisé OneHotEncoder et pour la normalisation des variables quantitatives on a utilisé StandardScaler.

Pour choisir les variables pertinentes à notre modélisation .On a utilisé le module feature selection avec les méthodes (Threshold, Select kbest, Select From Model).

Le jeu de données initial a été séparé en plusieurs parties de façon à disposer :

D'un jeu de training (75% des individus) qui a été séparé en plusieurs folds pour entraîner les différents modèles et optimiser les paramètres (cross validation) sans overfitting. D'un jeu de test (25 % des individus) pour l'évaluation finale du modèle

III. Problématique de l'étude :

La présente étude concerne une classification binaire avec deux classes déséquilibrées (8% pour les clients en défaut encodé par la valeur 1 contre 92% pour les clients sans défaut de paiement encodé par la valeur 0).

Ce déséquilibre peut impacter la performance du modèle et fausser les résultats. En effet, le modèle aura tendance à prédire la classe en majorité, dans notre cas les clients sans défaut.

Pour remédier à ce problème trois approches peuvent être utilisées:

- Undersampling : diminue le nombre d'observations de la classe majoritaire afin d'arriver à un ratio satisfaisant.
- Oversampling : augmente le nombre d'observations de la classe minoritaire afin d'arriver à un ratio satisfaisant.
- Génération d'échantillons synthétiques : crée des échantillons synthétiques à partir de la classe minoritaire. Ces méthodes sont à appliquer avant l'entraînement du modèl

Dans notre cas , On a utilisé la dernière approche avec le module SMOTE de Imblearn combinée avec la méthode cross validation avec une stratégie de stratification adaptée à ce genre de problématique qui est la méthode de RepeatedStratifiedKFold

IV. Fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation:

Dans le contexte bancaire, ce qui est important c'est de minimiser le risque financier lié à l'insolvabilité des clients en d'autres termes c'est de réduire le taux des faux négatifs des clients à défaut qui sont prédits comme des bons clients .

	Clients prédits en défaut	Clients prédits sans défaut
Clients réellement en défaut	Vrais positifs	**Faux négatifs**
Clients sans défaut	Faux positifs	Vrais négatifs

La courbe ROC représente le taux de vrais positifs (TPR) par rapport au taux de faux positifs (FPR). La métrique ROC_AUC correspond à l'aire sous la courbe ROC, elle est comprise entre 0 et 1.

Contrairement à La métrique ROC_AUC qui ne tient pas en compte le seuil de classification .Une autre métrique sera utilisée dans ce sens qui est le fbeta score . Cette métrique permet de définir le poids qu'on souhaite attribuer au recall ou à la précision. Pour rappel les formules de recall, précision et fbeta score sont les suivantes:

$$recall = \frac{Vrai\ Positif}{Vrai\ Positif + Faux\ Négatif} \qquad precision = \frac{Vrai\ Positif}{Vrai\ Positif + Faux\ Positif}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathbf{precision} \cdot \mathbf{recall}}{(\beta^2 \cdot \mathbf{precision}) + \mathbf{recall}}.$$

Comme mentionné déjà , on préfère limiter un risque financier plutôt que le risque de perdre un client donc le recall est plus important que la précision pour exprimer

ça dans l'équation de fbeta score on donne une valeur de beta égale à 2 . (C'est une métrique qu'on cherche à maximiser).

V. Modèle testés et choix final :

Dans le cadre de la présente problématique ,différents modèles de classification ont été testés : GaussianNB, LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier et XGBClassifier.

Parmi ces modèles, les meilleurs résultats sont obtenus avec LogisticRegression qui est un algorithme de classification binaire .

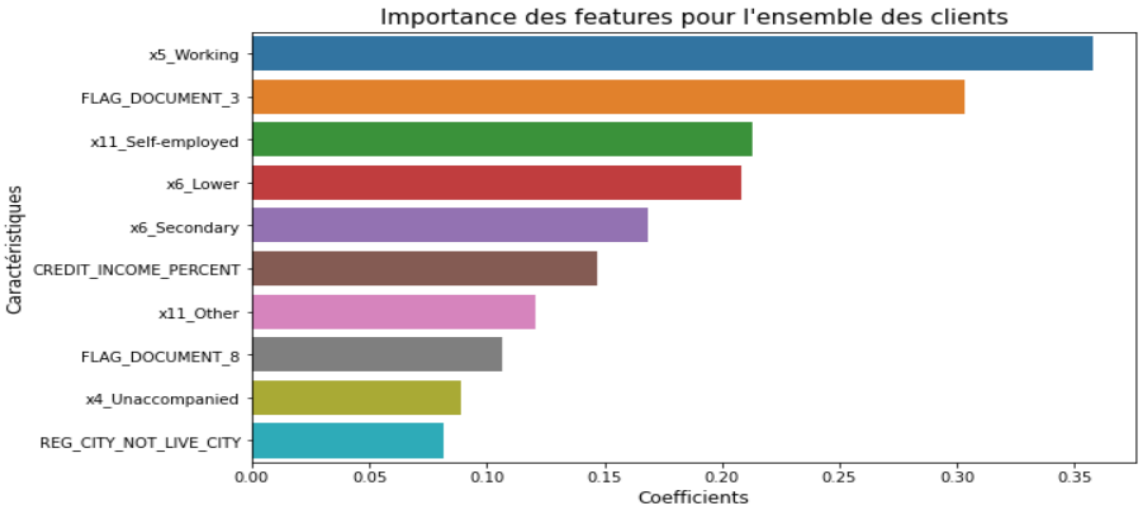
Pour l'optimisation et le choix des hyperparamètres du modèle sélectionné .On a choisi le module HyperOpt

VI. L'interprétabilité globale et locale du modèle:

Pour l'interopérabilité du modèle qui sera destinée soit à l'équipe métier soit aux clients , on a utilisé deux méthodes :

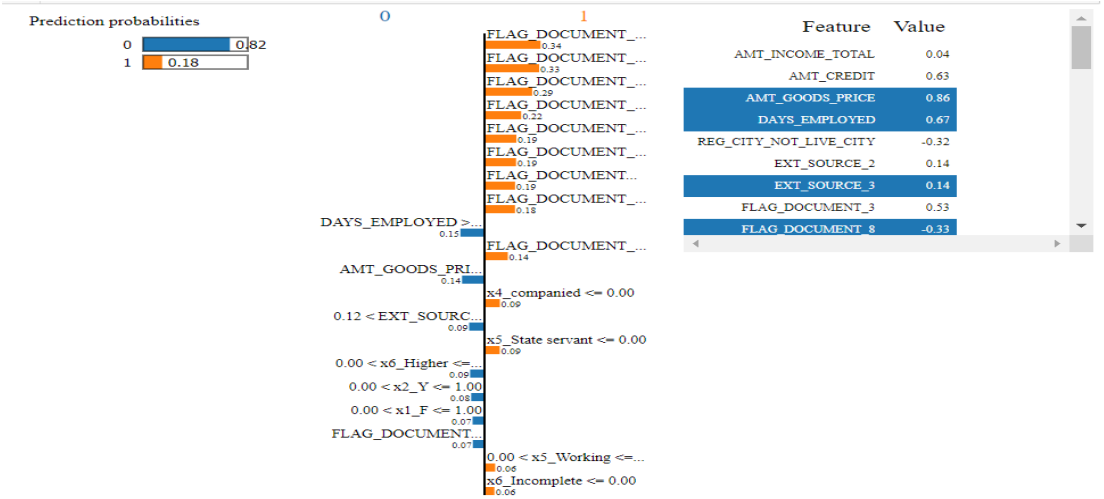
- Une méthode d'interprétabilité globale qui tiendra en compte l'ensemble des données et qui dépendra du modèle choisi dans notre cas la régression Logistique.

  Notre modèle de régression logistique nous donne l'importance des variables suivantes :



- Une méthode d'interopérabilité locale c'est à dire pour chaque client choisit on cherche à savoir les variables qui ont influencé le plus sur le score final. On a utilisé les méthodes SHAP et LIME

  Ci joint un exemple d'interprétabilité locale avec LIME pour une seule observation(un seul client).



VII. Les limites et les améliorations possibles :

Le Beta de la métrique fbeta score a été fixé sur l'hypothèse que le recall et plus important que la précision ou dans le sens métier que le risque financier est plus important que le risque de perte d'un client chose qui n'est pas confirmée ou rejetée par les gens du métier

La même chose peut être dite pour le seuil de classification qui a été fixé à 50% de façon aléatoire sans se baser sur un critère métier fiable.

La préparation et le feature engineering peut être de façon beaucoup plus affinée avec plus de temps.