PROJET VII

NOTE MÉTHODOLOGIQUE

Ameur Ltaief

CONTEXTE

Cette note est un livrable détaillant la méthode d'entraînement du modèle déployé, de l'optimisation de ce dernier.
Mon travail s'appuie sur un jeu de données fournies par la société " Prêt à dépenser".
Notre mission étant d'effectuer une classification binaire pour savoir si le futur client est en capacité de rembourser son empreint ou pas, on s'appuient sur un historique de clients.

Qu'es que l'entrainement du modèle ?

Pour qu'un modèle de prédiction puisse fonctionner, il y a une phase primordiale que l'on nomme l'entrainement du modèle.
Cela permet à notre algorithme de trouver le lien entre les données en entrée et la valeur en sortie.

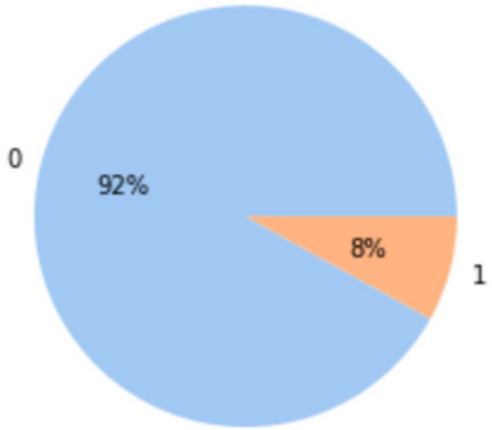
Dans notre cas, nous souhaitons effectuer une classification binaire, savoir si notre client est en capacité ou non de rembourser son emprunt.

Notre choix va donc tout naturellement s'orienter vers des algorithmes de classification.
Mais pour que les algorithmes puissent comprendre les données en entrée il faut respecter certaines conditions.
Tout d'abord n'avoir que des valeurs numériques en entrée, n'avoir aucune valeur manquante ou aberrante.
Soumettre un jeu de données équilibrées entre les deux valeurs de sortie pour une meilleure compression des liens. Et pour finir une mise en échelle des données pour qu'une variable d'entrée ne soit pas considéré plus importante que les autres.

MÉTHODOLOGIE D'ENTRAÎNEMENT DU MODÈLE

ANALYSE DES DONNEES

Le travail commence toujours par une analyse minutieuse des données.
Cette analyse nous permet de remarquer les valeurs manquantes et aberrantes.
Dans ce jeu de données, nous avons un déséquilibre de la valeur à prédire.



La valeur 0 représente les crédits acceptés et la valeur 1 représente les crédits refusés.
Un déséquilibre des valeurs à prédire est problématique pour les algorithmes de classification, le résultat peut être faussé. Il faut donc résoudre ce déséquilibre pour avoir des modèles performants.

VALEURS MANQUANTES ET ABERRANTES

Les valeurs aberrantes sont les valeurs non appropriées à une réalité exemple:
"une femme âgée de 350 ans".
Elles sont tout simplement remplacées par zéro.
Les valeurs manquantes sont remplacées par la médiane de la variable.

MÉTHODOLOGIE D'ENTRAÎNEMENT DU MODÈLE

FEATURE ENGINEERING

Cette phase permet d'encoder chaque variable non numérique pour avoir au final un jeu contenant seulement des valeurs numériques.
Il y a 2 méthodes:

- pour les variables contenant 2 valeurs exemple: le sexe "Homme" ou "Femme"
- Pour les variables contenants plus de 2 valeurs exemple: le contrat de travail "CDD", "CDI" ou bien "intérim".

Nous avons aussi créé de nouvelle feature synthétique qui aura une meilleure corrélation avec la valeur à prédire et qui permettra d'avoir des meilleurs résultats.

ECHANTIONNAGE

Notre jeu de données contenant la valeur à prédire est coupé en 2 parties.
Une première de 80% des données qui servira à l'entrainement et seconde de 20% de données qui servira de test pour tester notre modèle et comparer les résultats de prédiction avec les résultat réels.

ENTRAINEMENT

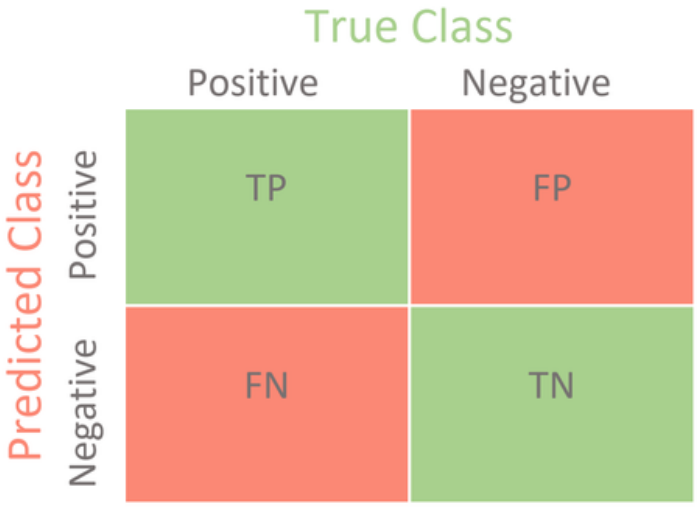
Pour corriger notre déséquilibre de valeurs de sortie, nous avons fait appel à la fonction SMOTE qui permet de créer des informations synthétiques en faveur de la valeur de sortie faiblement présente.
Nous avons au final un jeu de données avec autant d'individus pour les 2 valeurs.
Ce jeu de données ne servira que pour la phase d'entrainement de nos algorithmes de classification.

METRIQUES D'EVALUATION

Après avoir entrainé plusieurs algorithmes de classification avec notre jeu de données équilibré, il nous faut identifier le modèle le plus adapté à notre problématique métier.
Pour évaluer nos modèles nous avons utilisé plusieurs métriques.

Lorsque nos algorithmes sont entrainés, on utilise le set de test pour tester nos modèles.
Notre tâche est une classification binaire et nous allons comparer les classes prédites avec les classes réelles.
Pour ce faire dans un premier temps , on utilise une matrice de confusion pour comparer les valeurs réelles avec les valeurs prédites.
Dans cette classification binaire, la valeur 0 sera nommée la valeur négative et la valeur 1 sera la valeur positive.

Matrice de confusion:



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

II y a 4 catégories différentes présentes dans notre matrice.
Comme on peut le lire sur le tableau, FP représente par exemple le nombre d'individus qui ont une valeur négative dans les classes réelles et une valeur positive dans les classes prédites.

METRIQUES D'EVALUATION

Le résultat des différentes métriques selon le modèle :

modele	accuracy	precision	recall	f1_score
Logistic Regression Imbalanced	0.92	0.00	0.00	0.00
Logistic Regression SMOTE	0.69	0.16	0.66	0.25
KNN	0.85	0.35	1.00	0.52
Random Forest	0.67	0.14	0.62	0.23
Gradient Boosting	0.90	0.23	0.09	0.13

(Voir page précédente pour les formules de nos métriques)

Le F1-score est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus. Il est particulièrement utilisé pour les problèmes utilisant des données déséquilibrées.
Le F1-score permet de résumer les valeurs de la precision et du recall en une seule métrique.

Notre choix se porte sur le modèle avec le meilleur score F1, dans notre cas le modèle retenu est le KNN Classifier.

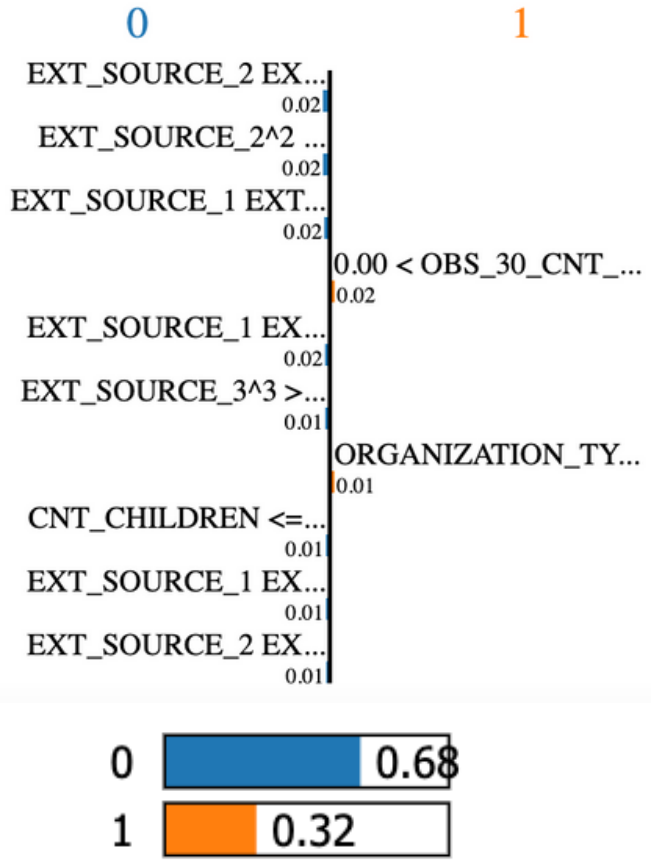
Après avoir choisi l'algorithme de prédiction, on cherche les meilleurs hyperparametres pour l'optimisation.
Après différents tests, le choix optimal est un paramètre de 3 voisins (n_neighbors = 3).

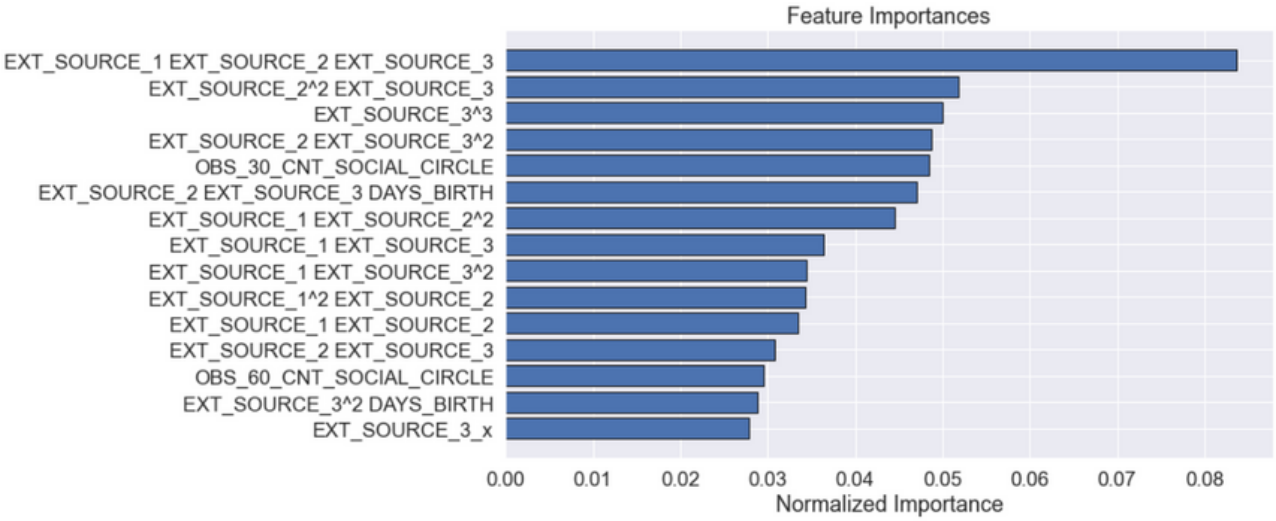
INTERPRETABILITE DU MODELE

L'idée est de comprendre au mieux notre modèle de prédiction et savoir quels features permet de prendre la décision finale.

Au niveau global l'algorithme retenu (KNN Classifier) ne nous permet pas d'avoir les features les plus importantes, je me suis donc appuyé sur un autre algorithme pour avoir un rapide aperçu (voir ci-bas).
Cela confirme l'utilité des features synthétiques.

Dans une approche locale (voir ci-contre), les résultats des features sont difficile à interpréter mais on retrouve les features synthétiques créent avec la fonction PolynomialFeatures.
Une coopération avec une équipe métiers permettra d'en comprendre un peu mieux le sens.





LIMITES ET AMELIORATION

Nous avons rencontré des limites concernant le déploiement de l'application, un modèle qui a été entrainé avec plusieurs centaines de milliers de données dépassera facilement les 100 Mo lors de sa sauvegarde.
Il serait donc intéressant de faire évoluer nos solutions de cloud vers une offre qui nous permettra d'exploiter pleinement de notre modèle de prédiction.

Nous avons aussi rencontré des difficultés dans l'interprétabilité de notre modèle, être accompagné par un spécialiste du crédit nous aurais permis de mieux comprendre nos données, et peut-être même faire évoluer nos résultats de prédiction.