Note méthodologique
Projet : Implémentez un modèle de scoring

0) Problématique et données :
On souhaite développer un modèle de scoring pour déterminer si une personne demandant un crédit va le rembourser ou non.
On dispose pour cela de plusieurs tables contenant des données diverses sur un ensemble de clients auquel un prêt a été accordé, dont la plupart ont pu le rembourser.

I) Méthodologie d'entraînement :

1) Prétraitement des données :
-Certaines données aberrantes ont dû être supprimées car leurs valeurs étaient nettement en-dehors des possibilités.
-Il a fallut supprimer les colonnes avec trop de données manquantes, puis des lignes avec des données manquantes : la quantité de données supprimée était négligeable par rapport au total, et la quantité originale de données était trop importante et nécessitait d'être réduite pour pouvoir être utilisée.
-Un one-hot encoding des données textuelles a ensuite été réalisé : les différentes valeurs n'étaient pas ordonnées.
-On a ensuite procédé à l'agrégation des lignes représentant un même client à l'intérieur de chaque table, en gardant le min, max et/ou la moyenne des variables en fonction de leur signification pour le métier
-La Fusion des tables a enfin été réalisée, via des jointures intérieures sur l'id des clients ; la table credit_card_balance contenait trop peu de lignes par rapport au reste et a été abandonnée.

-On a observé la distribution des variables :
--Individuellement à l'intérieur des tables
--Relativement à la cible dans la table finale

2) Préparations avant prédiction :
En premier lieu, il a fallut standardiser les variables d'entrées pour pouvoir faire fonctionner les modèles : le Gaussian Naïve Bayes, par exemple, a besoin de données d'entrée standardisées pour que le modèle mathématique qui lui est sous-jacent soit applicable et que les calculs aient des chances de donner des résultats corrects, ce qui a été réalisé avec StandardScaler.
Une analyse en composante principale a été testée, mais le problème immédiat de la perte d'interprétabilité des résultats c'est posé et cette approche a donc été abandonnée.

Les données considérés portent sur des clients qui se sont vu accorder un prêt ; le but des modèles considérés est de prédire si un client donné a eu des difficultés de remboursement. De fait on dispose déjà de cette information (ce qui permet d'évaluer la performance des modèles) : il s'agit de la variable cible (nommée Target), qui indique si le client considéré à fait face à des difficultés pour rembourser son prêt ou non. Concrètement, la variable prend la valeur 1 si le client a eu des problèmes et 0 si il n'a pas eu de problème.

Un oversampling et un undersampling des données d'entraînement des modèles a eu lieu pour équilibrer la distribution des valeurs de la cible, afin de contrebalancer le manque de clients en faillite dans le dataset, et d'éviter qu'un modèle qui classerait tous les clients

dans la classe majoritaire puisse être considéré comme bon par manque de clients sur lequel il se trompe : à la base, on avait 224 000 négatifs et 18 000 positifs ; via Synthetic Minority Oversampling Technique (SMOTE), on est monté à 24 000 positifs, et via Undersampling on est descendu à 112 000 négatifs.

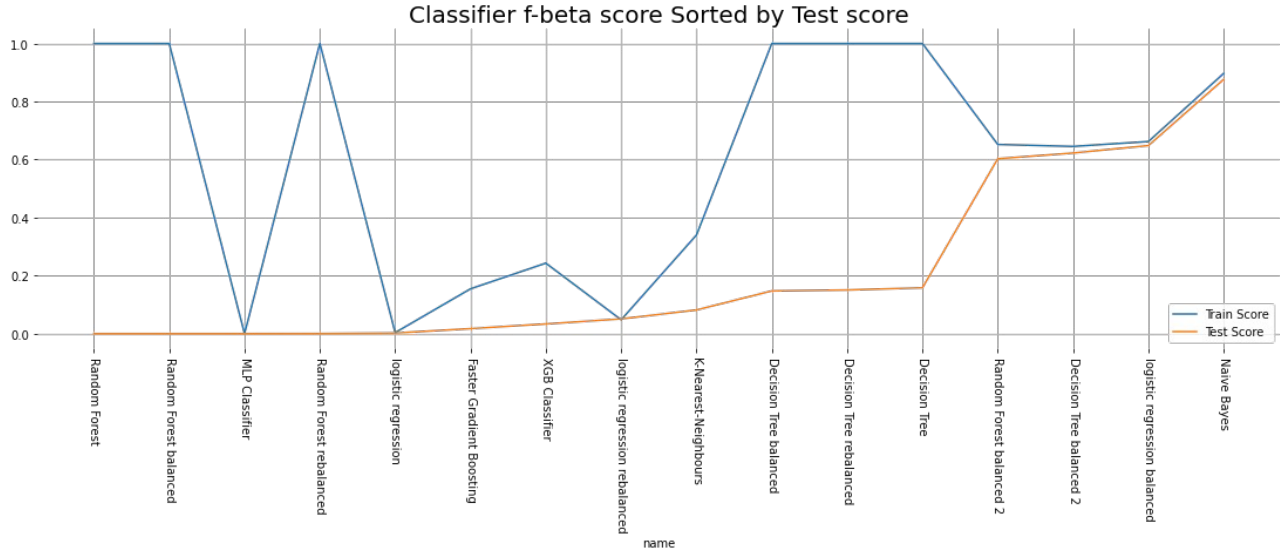
3) Métriques utilisées :

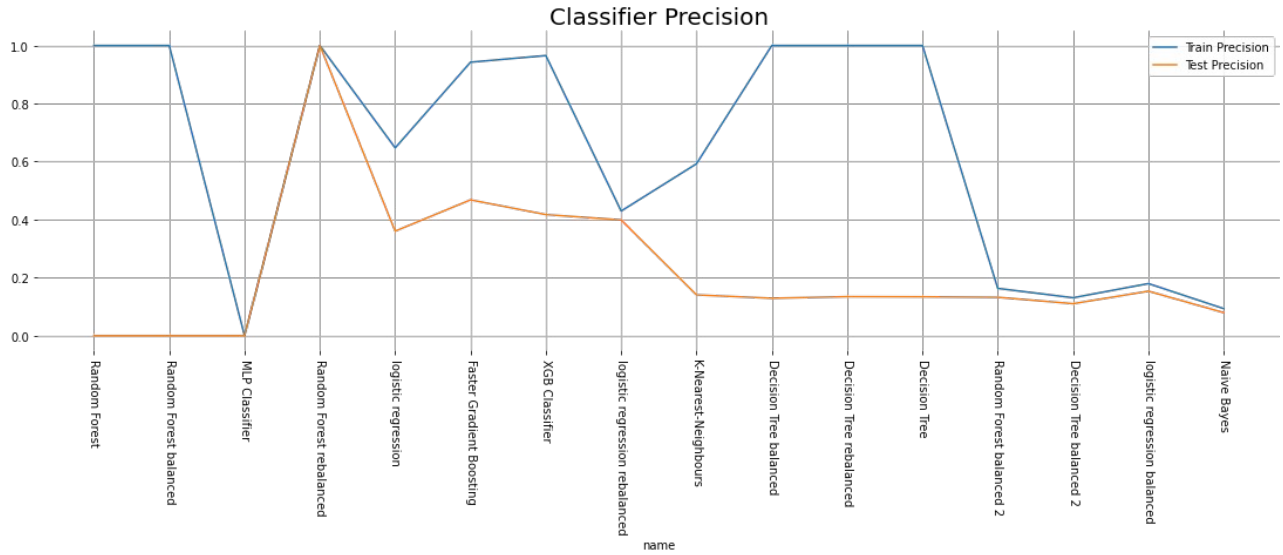
Les métriques choisies pour évaluer les différents modèles sont :
-La précision, qui mesure le taux de vrais positifs (clients ayant eu des difficultés de remboursement et identifiés comme tels par le modèle) sur le taux de positifs (clients que le modèle a identifiés comme ayant des difficultés, qu'ils en aient vraiment eu ou non)
-Le recall, qui mesure le taux de vrais positifs sur le taux de vrais positifs et de faux négatifs (clients ayant eu des difficultés de remboursement que le modèle n'a pas identifiés comme tels)
-L'aire sous la courbe ROC : la courbe Receiver Operating Characteristic indique le nombre de vrais positifs et faux négatifs en fonction du seuil de décision de l'algorithme (la probabilité de non-remboursement à partir de laquelle celui-ci considère qu'un client ne va pas rembourser son prêt)
-Le f-bêta score : Moyenne harmonique pondéré de la précision et du recall ; il permet de favoriser le recall sans abandonner la précision, ce qui permet de l'utiliser comme fonction de coût métier avec béta égal à 10.

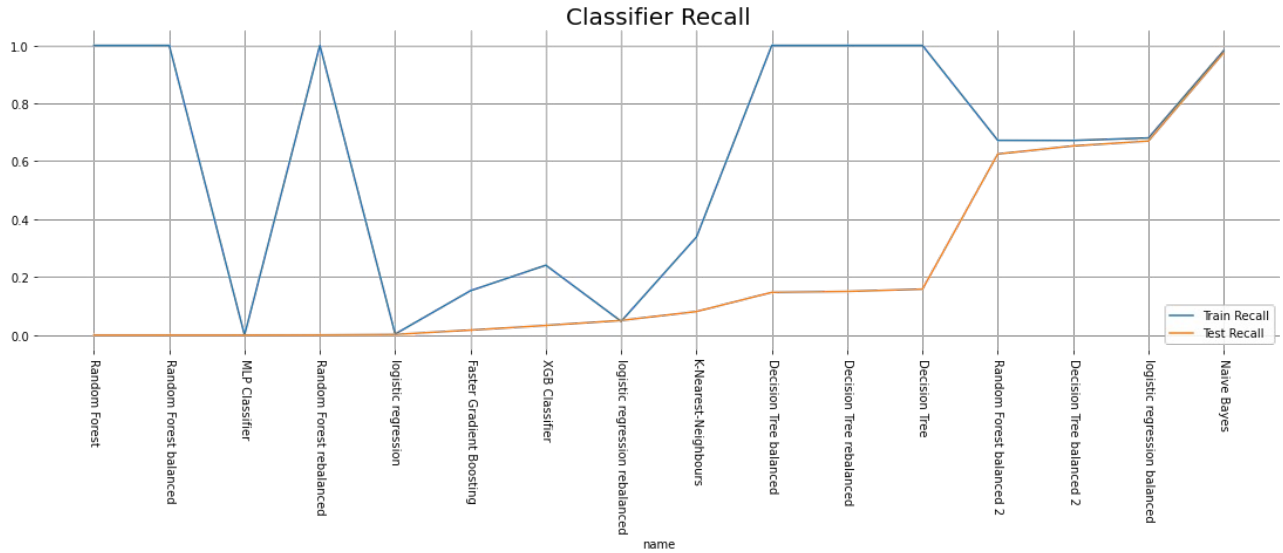
4) Évaluation des modèles :

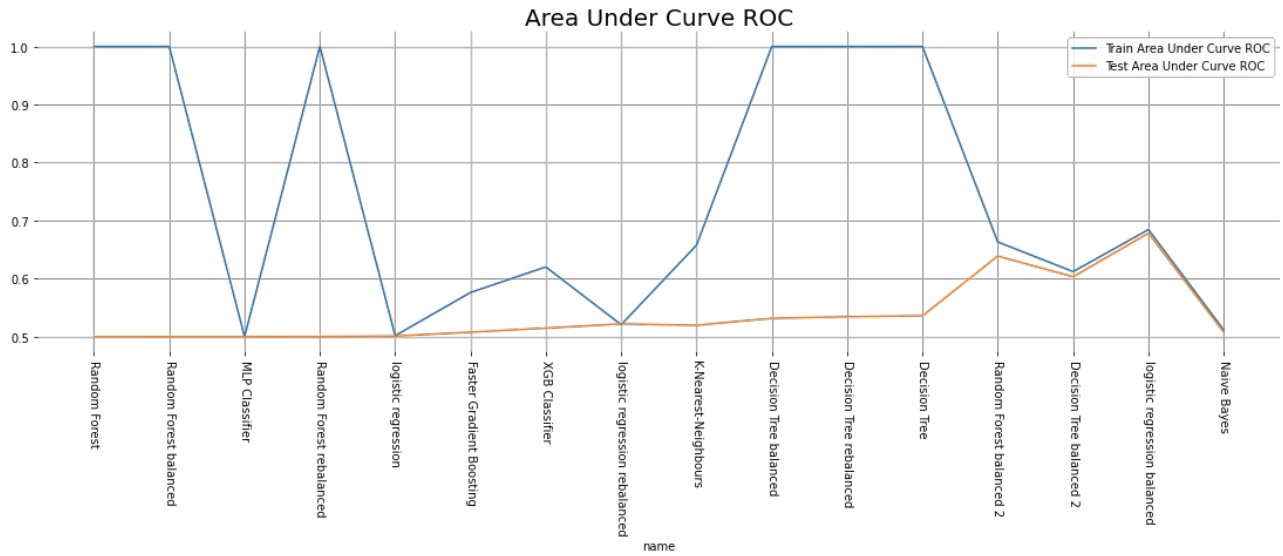
Les modèles évalués sont :
Un modèle classique, le Gaussian Naive Bayes ;
Un modèle de régression, Logistic Regression ;
Des modèles ensemblistes : Decision Tree, Random Forest, XGB Classifier ;
Un modèle de réseau de neurones : le MLP Classifier.

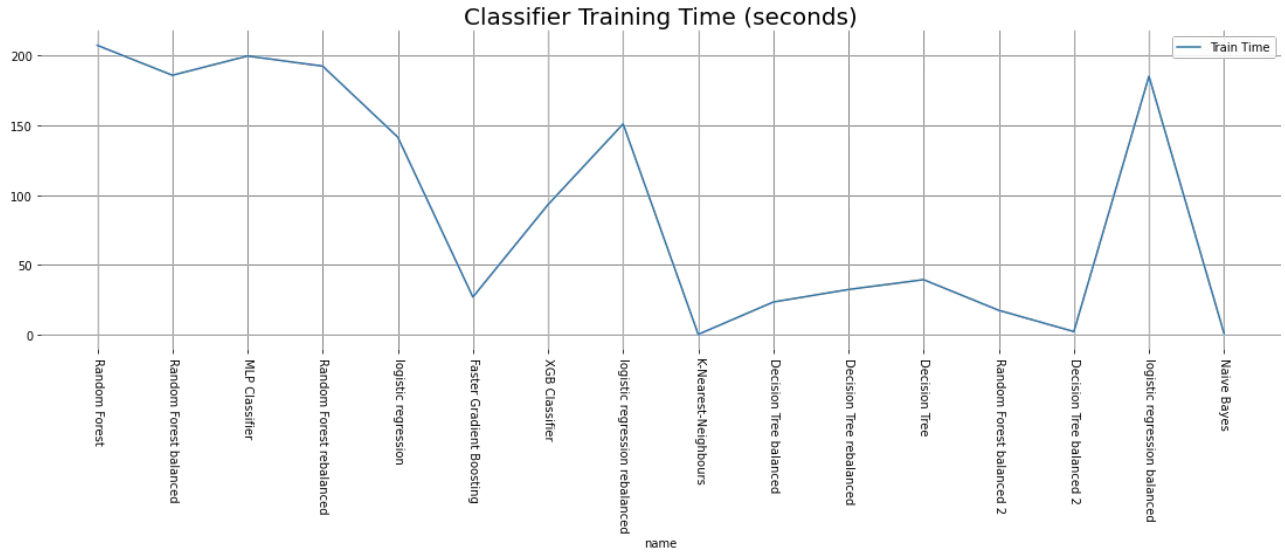
Les scores des modèles :




Classifier Precision
Train Precision
Test Precision
1.0
0.8
0.6
0.4
0.2
0.0
Random Forest
Random Forest balanced
MLP Classifier
Random Forest rebalanced
logistic regression
Faster Gradient Boosting
XGB Classifier
logistic regression rebalanced
K-Nearest-Neighbours
Decision Tree balanced
Decision Tree rebalanced
Decision Tree
Random Forest balanced 2
Decision Tree balanced 2
logistic regression balanced
Naive Bayes
name


Classifier Recall
Train Recall
Test Recall
1.0
0.8
0.6
0.4
0.2
0.0
Random Forest
Random Forest balanced
MLP Classifier
Random Forest rebalanced
logistic regression
Faster Gradient Boosting
XGB Classifier
logistic regression rebalanced
K-Nearest-Neighbours
Decision Tree balanced
Decision Tree rebalanced
Decision Tree
Random Forest balanced 2
Decision Tree balanced 2
logistic regression balanced
Naive Bayes
name


Area Under Curve ROC
Train Area Under Curve ROC
Test Area Under Curve ROC
1.0
0.9
0.8
0.7
0.6
0.5
Random Forest
Random Forest balanced
MLP Classifier
Random Forest rebalanced
logistic regression
Faster Gradient Boosting
XGB Classifier
logistic regression rebalanced
K-Nearest-Neighbours
Decision Tree balanced
Decision Tree rebalanced
Decision Tree
Random Forest balanced 2
Decision Tree balanced 2
logistic regression balanced
Naive Bayes
name



Les modèles les plus performants du point de vue de l'aire sous la courbe ROC furent la régression logistique et la forêt aléatoire, on a donc cherché à optimiser leurs performances.

3) Choix des hyperparamètres :

L'optimisation des performances des deux meilleurs modèles a eu lieu via l'utilisation de GridSearchCV pour trouver les hyperparamètres qui maximisent le score métier (f-béta score avec béta = 10) pour chaque modèle.
Les meilleurs hyperparamètres pour chaque modèle ainsi que les scores associés sont résumés dans les tableaux suivants :

Modèle	Régression logistique
Hyperparamètres	Valeurs
C (Inverse du poids de régularisation)	0.1
fit_intercept	False
class_weight	balanced
F-béta score	0,75

Modèle	Forêt aléatoire
Hyperparamètres	Valeurs
n_estimators	1000
max_depth	5
class_weight	balanced
F-béta score	0,61

Le meilleur score métier ayant été obtenu par la régression logistique, c'est ce modèle qui a été sélectionné pour la pipeline finale.

4) Modèle final :
La pipeline finale consiste en un Scaler (StandardScaler), un Oversampler (Smote), un Undersampler (RandomUnderSampler), et enfin un algorithme de classification (Logistic regression) ;
Après création de cette pipeleine, l'importance de chaque variable a été estimée, et les variables les moins importantes ont été supprimées pour alléger les calculs et ainsi obtenir un modèle fonctionnant plus rapidement. L'espoir était également d'améliorer les performances du modèle en supprimant ces variables, mais cela n'a pas été le cas : au

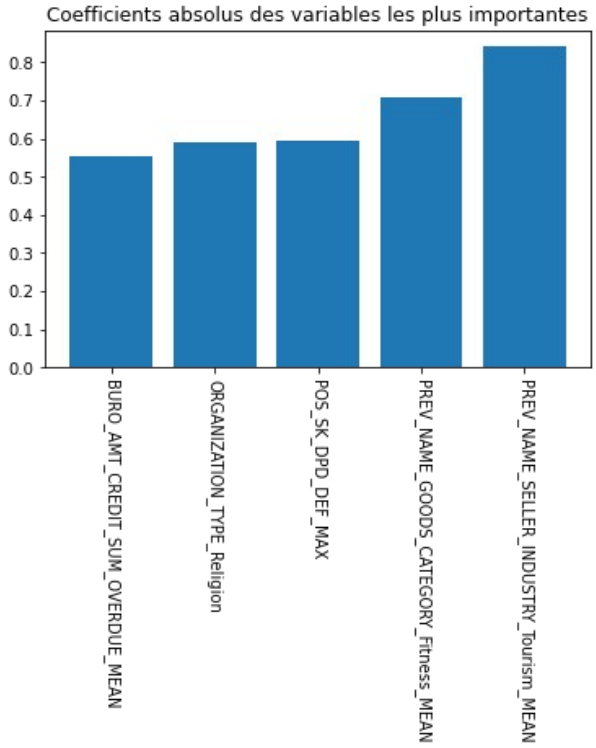
contraire, ces performances ont légèrement diminué, le score métier après cette suppression étant tombé à 0,73.
Cette diminution est cependant minime et le gain de vitesse dans les calculs est appréciable, donc c'est bien le modèle avec un ensemble réduit de variables qui a été conservé au final.

II) Interprétabilité du modèle :

1) Interprétabilité globale :

Les variables initiales restent reconnaissables après l'agrégation qui a eu lieu, on eut donc récupérer les variables que le modèle considère comme importantes et comprendre en les lisant les paramètres métier qui sont les plus importants.
Les variables les plus importantes pour la pipeline finale, déterminées par les coefficients attribués par le modèle et par l'importance de permutation (en prenant la liste des 5 plus importantes à chaque fois et en fusionnant ces deux listes) sont :
PREV_CHANNEL_TYPE_Creditandcashoffices_MEAN,
BURO_CREDIT_TYPE_Consumercredit_MEAN,
PREV_NAME_GOODS_CATEGORY_XNA_MEAN,
PREV_NAME_SELLER_INDUSTRY_XNA_MEAN, OBS_60_CNT_SOCIAL_CIRCLE,
OBS_30_CNT_SOCIAL_CIRCLE, BURO_AMT_CREDIT_SUM_OVERDUE_MEAN, et
BURO_CREDIT_DAY_OVERDUE_MEAN.

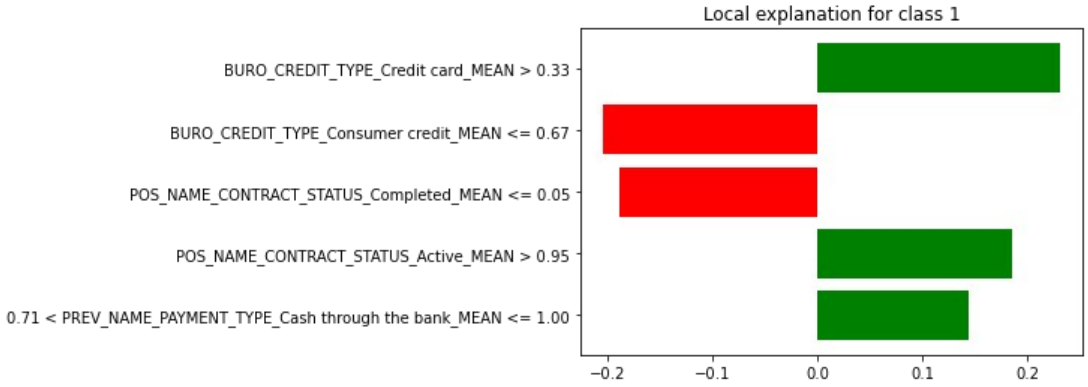
Ce graphique donne les coefficients absolus des variables pour lesquels ces coefficients sont les plus importants dans le modèle de la régression logistique :



2) Interprétabilité locale :

On peut calculer, pour un client donné, les variables les plus importantes et leur effet sur la probabilité que le client rembourse ou non son prêt.
La bibliothèque Shapley a été utilisée en premier lieu, mais le format des données calculées par cette bibliothèque se prête mal à l'export, alors que les données calculées via la bibliothèque Lime s'y portent très bien ; c'est donc cette dernière qui a été utilisée au final.

Un exemple des variables les plus importantes et leur effet pour un client donné :



On voit ici que le client va voir sa demande de prêt refusée (le modèle prédit que ce client appartiendra à la classe 1, qui regroupe les clients en défaut de paiement) en raison de plusieurs facteurs : le premier est que plus d'un tiers de ses emprunts passés étaient pour des cartes de crédit (BURO_CREDIT_TYPE_Credit card_MEAN > 0,33), ce que le modèle interprète comme augmentant ses chances de ne pas rembourser de plus de 20 %; le second que moins de deux tiers de ses emprunts étaient pour un crédit à la consommation, ce qui diminue ses chances de ne pas rembourses de 20 %; le troisième facteur est que moins de la moitié des contrats de prêt qu'il a signé sont qu statut comptété, ce qui diminue ses chances de ne pas rembourser de 18 %; ainsi que d'autres facteurs.

III) Limites et amélioration possibles :

-Optimiser les features créées lors de l'aggrégation en fonction des performances et des critères métiers via des échanges avec des spécialistes du métier pour récupérer notamment les variables qui portent le plus de sens par rapport au métier ;
-Gestion du déséquilibre : explorer d'autres méthodes, par exemple en modifiant la proportion de positifs créés ou de négatifs supprimés ;
-Tester d'autres fonctions de coût.