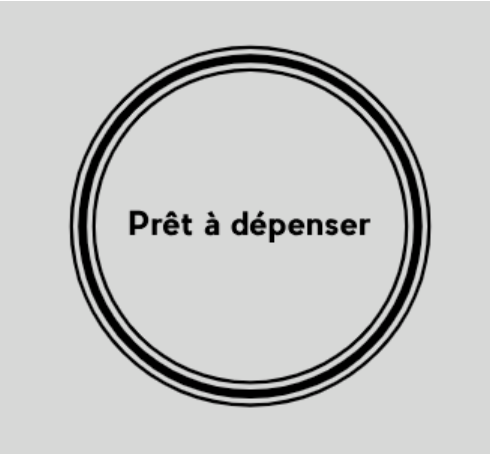
**Bastien BOYER Année 2023**

**Data Scientist**

**Projet 7 :**

**Implémentez un modèle de scoring**



**Note Méthodologique**

**Contexte :**

Demande de mise en œuvre d’un modèle de scoring pour la société financière « Prêt à dépenser ». La société propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d’historique de prêt.

Le modèle de scoring aura pour but de mettre en œuvre un outil qui permet de calculer la probabilité d’un client de pouvoir rembourser son crédit et classifier la demande en crédit accordé ou refusé. La société souhaite donc développer un algorithme de classification en s’appuyant sur des sources de données variées (données comportementales, données provenant d’autres institutions financières…).

Les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeur de transparence vis-à-vis des décisions d’octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l’entreprise veut incarner.

La société « prêt à dépenser » souhaite donc le développement d’un dashboard interactif pour que les chargés de relation 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.

**Sommaire :**

1. Méthodologie d’entrainement du modèle
2. Traitement du déséquilibre des classes
3. Fonction coût métier, algorithme d’optimisation et métrique d’évaluation
4. Tableau de synthèse des résultats
5. Interprétabilité globale et locale du modèle 1 page max
6. Limites et améliorations possibles 1 pages max
7. Analyse du Data Drift 1 page max

**I) Méthodologie d’entrainement du modèle :**

Afin de faciliter le prétraitement des données, ce projet s’est inspiré du repository publique dont le lien est :

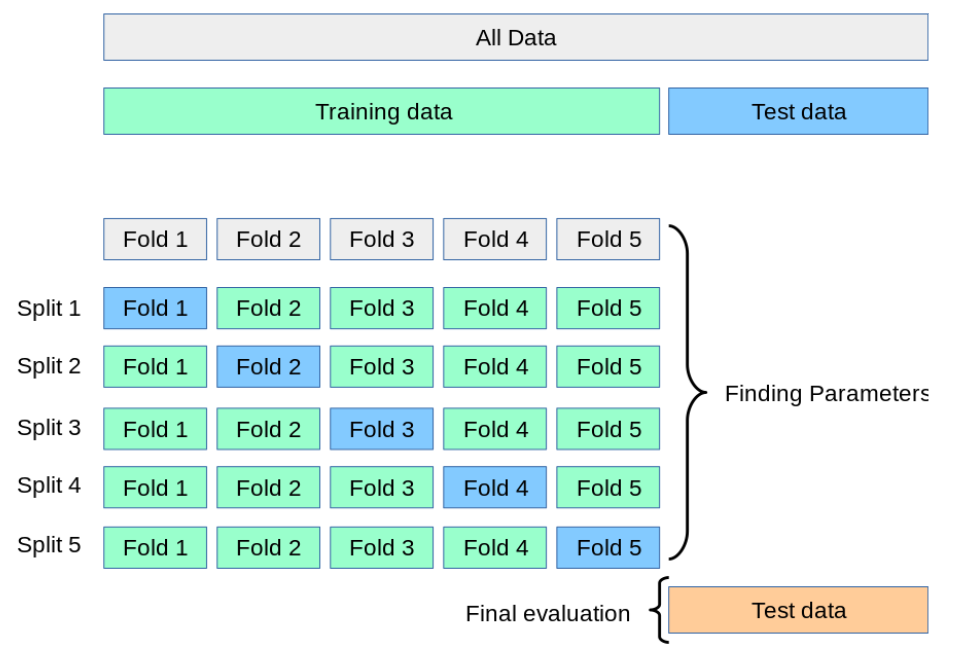
<https://github.com/KeyvaNoroozikia/ScoringModel.git>

La méthodologie d’entrainement du modèle consiste à entrainer différents modèles de classification mais au préalable nous allons séparer notre jeu de données en deux parties. Un jeu d’entrainement représentant 80% des données et un jeu de test représentant 20% des données.

Pour chaque modèle nous utiliserons GridSearchCV afin de pouvoir identifier les meilleurs hyper paramètres.

En effet grâce à cette librairie nous avons réalisé une cross validation (validation croisée) et mis ce paramètre à 5. Cette méthode permet de vérifier la stabilité de l’apprentissage en découpant le jeu d’entrainement en 5 parties. Cela évite que le modèle ne soit trop spécialisé sur les données qui ont servis à l’apprentissage et donc limite l’overfitting. GridSearchCV va donc tester une série de paramètres et comparer les performances pour en déduire le meilleur paramétrage.

Vous pouvez voir le principe de validation croisée dans l’image ci-dessous :



Avec la cross validation, nous avons une augmentation du temps d’entrainement dû au fait que le modèle sera entrainé et testé plusieurs fois.

Nous avons appliqué cette méthode à modèles différents qui sont :

* DummyClassifier
* LogisticRegression
* LIGHTGBM
* XGBoostClassifier
* RandomForestClassifier

Le modèle DummyClassifier a servis de modèle de base dans notre étude et nous donne des résultats que nous pourrons comparer à ceux trouver ultérieurement. La stratégie d’entrainement utilisé est prior. Cela renvoi une valeur qui est la plus fréquente dans l’ensemble de l’apprentissage.

Ces étapes ont permis de constater un déséquilibre des classes. Nous avons 19797 observations pour la classe 1 de la variable Target (variable que l’on cherche à prédire) contre 226241 de la classe 0. Nous allons voir comment le traiter.

**II) Traitement du déséquilibre des classes :**

Pour rectifier ce déséquilibre, nous allons faire un sur-échantillonnage (oversampling) à l’aide de la méthode SMOTE. Elle augmente de façon artificielle le nombre d’observations à partir des données existantes. On a donc un nombre d’individus minoritaire augmenté pour que ceux-ci est plus d’importance lors de la modélisation.

Le SMOTE évite de réaliser un simple clonage des données mais génère de nouveaux individus qui ressemblent aux autres sans être totalement identiques. On obtient une densification plus homogène de la population traitée.

Un des point d’attention dans l’utilisation du SMOTE est l’importance de standardiser les données. Nous avons utilisé RobustScaler pour cette tâche qui supprime la médiane et met à l’échelle les données en fonction de l’intervalle des quantiles. En effet, le SMOTE repose sur le calcul de distances euclidiennes entre les points. L’échelle des variables a donc un impact et cette méthode aura pour conséquence de pouvoir traiter toutes les données de la même manière.

Pour conclure le SMOTE améliorera l’overfitting en générant des nouvelles instances au lieu de les répliquer.

**III) Fonction coût métier, algorithme d’optimisation et métrique d’évaluation :**

1. Fonction coût métier :

Nous sommes partis du constat qu’il est préférable de privilégier un taux de faux négatifs faible plutôt qu’un taux de faux positifs faible. En effet, les faux négatifs représentent ici les crédits que nous avons accordés mais pour lesquels clients ont eu des difficultés de remboursement et par conséquent un risque de perte financière pour l’entreprise à cause du défaut de paiement. Les faux positifs quant à eux représentent des clients pour lesquels un refus de financement a été donné à des clients éligibles à l’octroi du prêt. Le risque est un manque un gagner pour la banque. Les étapes précédentes nous ont permis de déterminer que le modèle à privilégier est le modèle XGBoostClassifier car il a un bon score Accuracy et il possède un taux de faux négatifs bas (scores des modèles dans la partie IV).

1. Algorithme d’optimisation :

Pour l’étude nous allons estimer qu’un faux négatif est 10 fois plus importants (= un coût financier 10 fois plus importants) qu’un faux positif.

Nous devons donc créer un score qui sera le résultat du calcul suivant :

**Score = 10 \* Taux de faux négatifs + Taux de faux positifs**

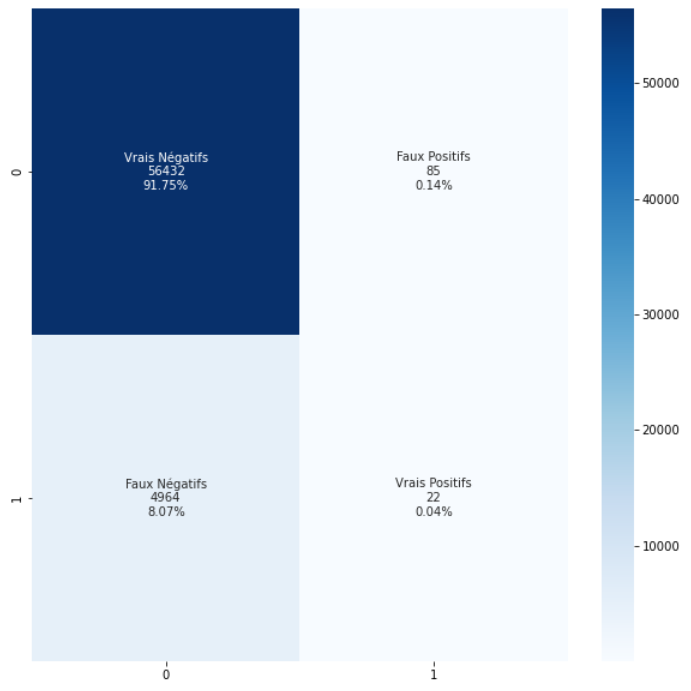
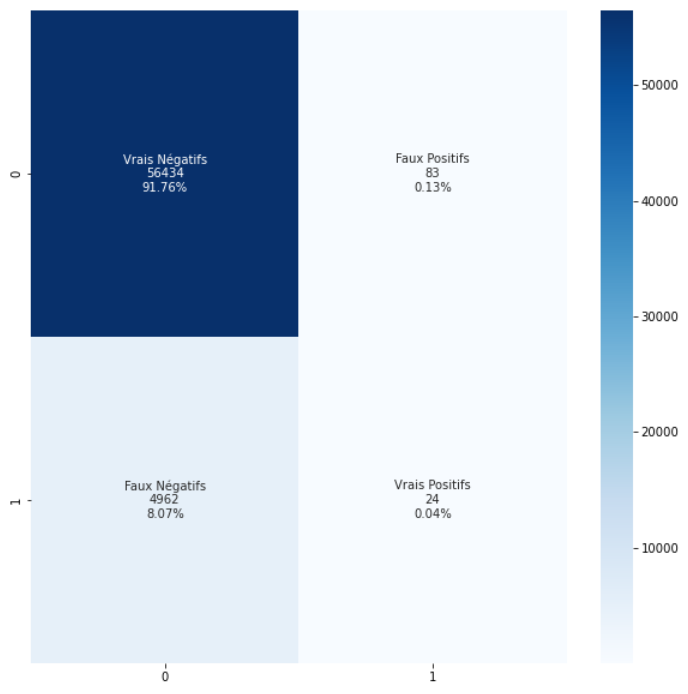
1. Métrique d’évaluation :

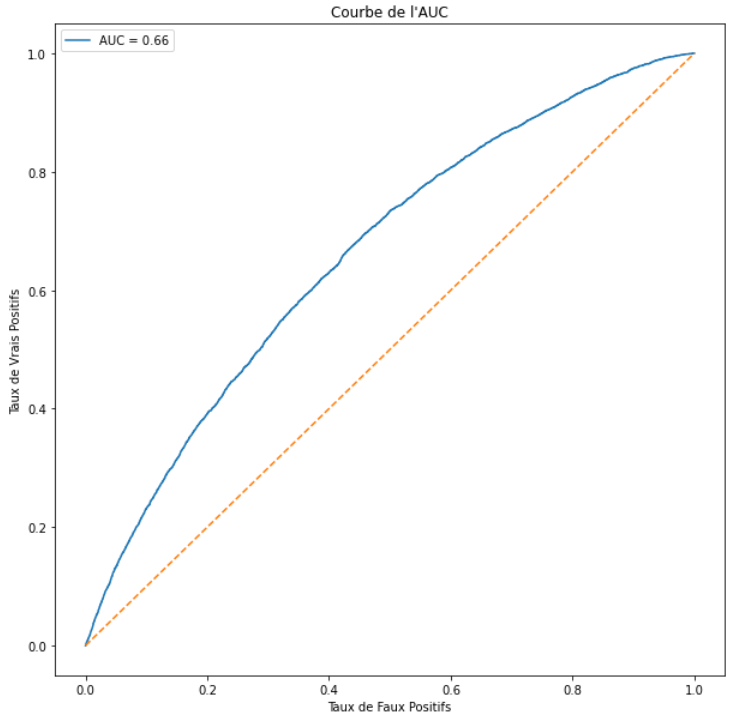
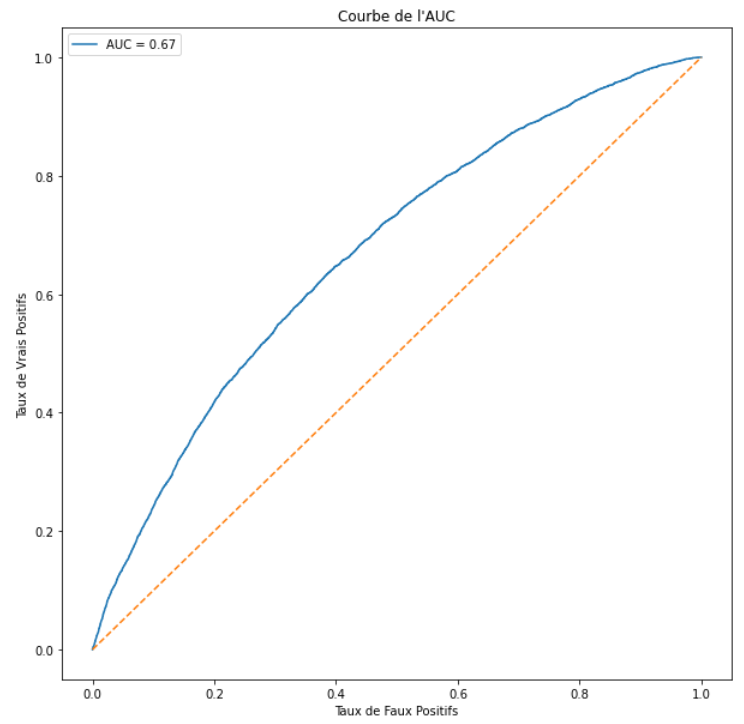
Une fois le modèle choisi (XGBoostClassifier) et les hyper paramètres optimisés (colsample\_bytree, learning\_rate, n\_estimators, subsample, max\_depth) d’un point de vue technique en utilisant les scores AUC et Accuracy, nous avons appliqué la fonction coût métier élaborée précédemment à l’aide du calcul d’un Score. Notre objectif est de réduire le résultat de ce score. Nous aurons donc limité les pertes de l’entreprise en appliquant ces étapes.

**IV) Tableau de synthèse des résultats :**

|  |  |  |  |
| --- | --- | --- | --- |
| NOM DU MODELE | TEMPS D’ENTRAINEMENT | AUC | ACCURACY |
| DummyClassifier | NC | 0,50 | 0,50 |
| LogisticRegression | 36,2 ms | 0,70 | 0,65 |
| LIGHTGBM | 28,2 ms | 0,66 | 0,92 |
| XGBoostClassifier | 46,6 ms | 0,67 | 0,88 |
| RandomForestClassifier | 43,3 ms | 0,66 | 0,89 |

Avant fonction coût métier Après fonction coût métier

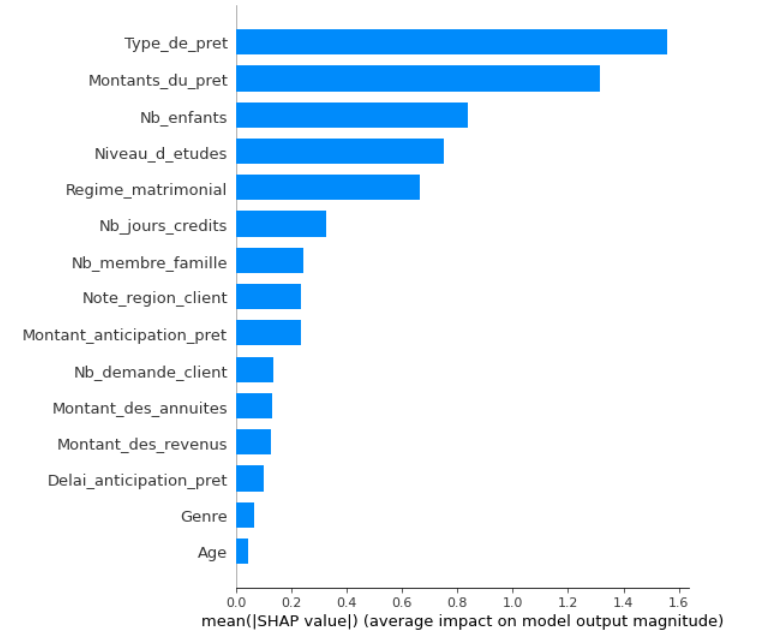
 

**V) Interprétabilités globale et locale du modèle :**

Afin de pouvoir trouver quels sont les éléments qui ont eu une incidence sur le résultat de la prédiction, nous avons utilisé une approche globale, c’est-à-dire l’importance des variables dans leur globalité puis une approche locale, c’est-à-dire l’importance des variables pour un client sélectionner.

Pour faire ce travail la méthode SHAP (SHapley Additive exPlanations) qui a une approche très précise de la cohérence entre les variables cependant en cas d’un nombre important de données, le temps d’entrainement peut être très long.

1. Interprétabilité globale :



On peut voir que les variables qui impactent fortement le modèle de prédiction sont le type de prêt ainsi que le montant du prêt.

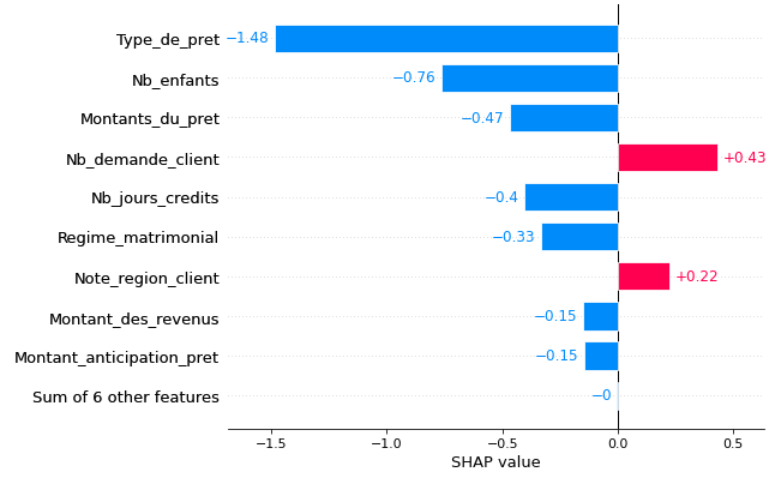
L’âge quant à lui a un impact mineur sur le résultat cependant cette information fait partie des éléments obligatoire d’identification lors d’une demande de prêt (date de naissance).

1. Interprétabilité locale :

En lien avec l’exigence de loyauté et transparence vis-à-vis de ses clients, le graphique de gauche permet d’informer le client des éléments qui ont permis de donner un accord ou un refus à sa demande de prêt.

Ici nous sommes en présence d’un refus et on peut voir que les variables qui ont motivé ce refus sont le type de prêt et le nombre d’enfants ont eu impact négatif car cela touche la capacité de remboursement du client.

En revanche, le nombre de demande de prêt du client ainsi que son lieu d’habitation ont eu impact positif car cela touche la capacité de remboursement du client.



**VI) Limites et améliorations possibles :**

On peut constater dans les différentes interprétabilités que nous avons des variables ayant un faible impact. Il pourrait donc être judicieux de voir si la suppression d’une d’entre elle ne remet pas en cause les résultats et pourrai diminuer le volume de données traitées/collectées.

Nous aurions pu utiliser la méthode LIME pour les interprétabilités globale et locale. En effet LIME n’a pas besoin de beaucoup de variable pour son étude et on peut l’utiliser sur des données textuelles ou sur des images (ce n’est pas le cas dans notre projet). Un autre avantage est son délai d’exécution plus rapide qu’en utilisant la méthode SHAP. Toutefois cette méthode peut donner des combinaisons de valeurs non cohérentes (= lien entre le résultat et la variable non cohérent).

Afin d’améliorer les performances de la modélisation nous pourrions collecter une information supplémentaire qui serait l’adossement ou non d’une garantie sur le prêt réalisé. En effet un gage sur un produit lors d’un prêt à la consommation pour financer ce même produit réduirai considérablement le risque de perte financière pour la banque.

Une des limites importantes pour cette étude est le volet RGPD qui a un impact sur notre résultat. En effet, plus nous avons de données, meilleur le résultat est cependant certaines d’entre elles peuvent être considérées comme discriminantes ou demandent un accord explicite du client.

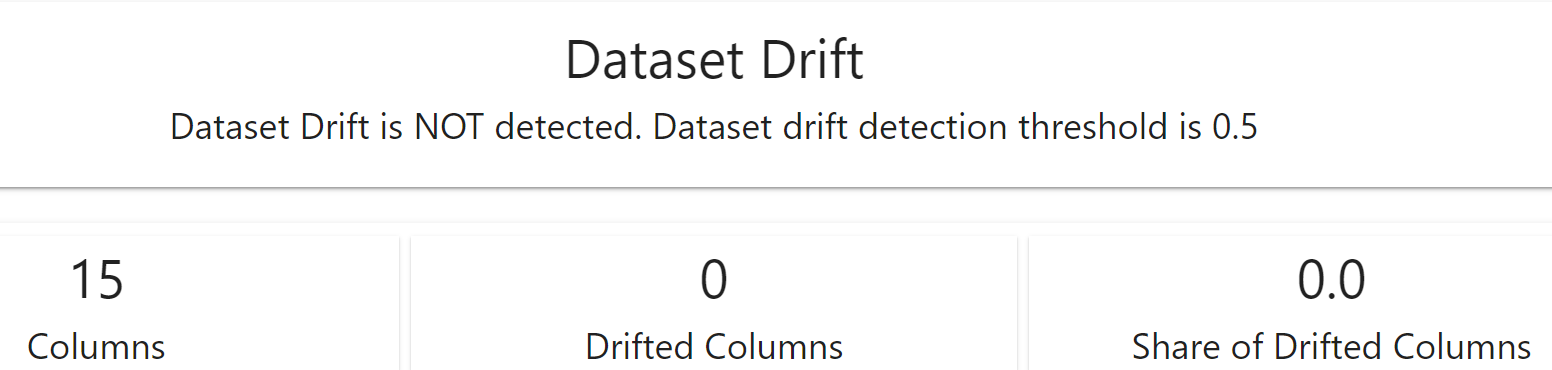
Enfin un autre des axes d’amélioration serait de travail sur la valeur du poids utilisée dans la fonction coût métier. Il était demandé explicitement de prendre un poids de 10 mais une autre valeur pourrait nous permettre d’améliorer les résultats de la modélisation.

**VII) Analyse du Data Drift :**

Afin d’identifier si la distribution de données dérive dans le temps, nous allons vérifier la présence de data drift. Pour cela nous utiliserons la librairie Evidently qui permet de faire cette mesure et de générer un tableau html avec les résultats correspondants.

Nous avons réalisé l’analyse sur notre dataset comportant les variables utiles seulement et nous l’avons scindé en deux partie avec une proportion 80/20. Nous avons donc obtenu un dataset d’entrainement de 246008 lignes sur 15 colonnes et un dataset de test de 61503 lignes sur 15 colonnes.

Voici les résultats que nous obtenons dans la page html générée (vous pouvez retrouver le détail des résultats dans le notebook joint) :



On peut constater que nous ne sommes pas en présence de data drift (nous sommes inférieurs au seuil e 0.5). Toutefois il faudra penser à réaliser des analyses biannuelles ou trimestrielles afin de vérifier que notre modélisation n’est pas perturbée par du data drift.