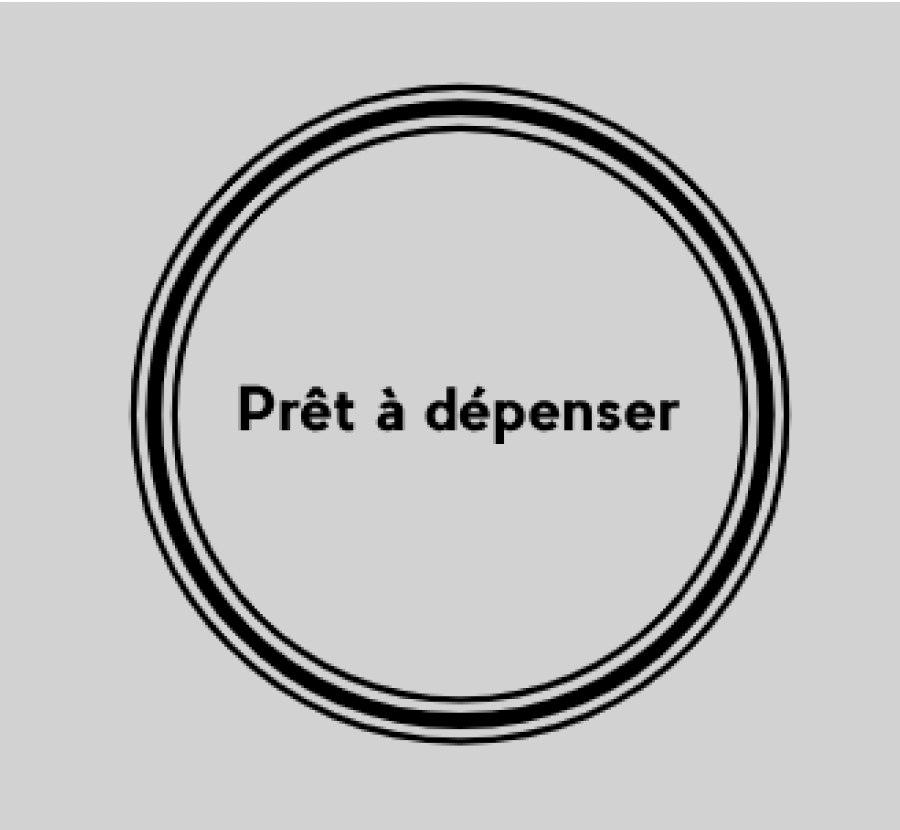


Note Méthodologie

Projet 7 – Data Scientist

*Implémentez un Modèle Scoring*

Samuel Dieudonné YEBEL



***Sommaire :***

I. Présentation du projet

II. Traitement des Données

III. Modélisation

A. Division des Données

1. Choix des Algorithmes
2. Choix des Hyperparamètres de chaque Algorithme
3. Rééquilibrage des données
4. Choix d’un Algorithme Optimal
5. Fonction Coût Métier
6. Métriques d’Evaluation
7. Coefficient Optimal
8. Seuil de Probabilité
9. Analyse des résultats obtenus
10. Interprétabilité globale et locale du modèle
11. API & DASHBOARD
12. Les limites et les améliorations possibles
13. Analyse du data drift

IV. Conclusion

I. Présentation du projet

La société financière « Prêt à dépenser » qui propose des crédits à la consommation aux personnes ayant peu ou pas du tout d'historique de prêt, sollicite son Data Scientist dans le cadre d’un projet de Scoring, afin de calculer la probabilité de défaut sa clientèle et classifier ensuite la demande en crédit accordé ou refusé.

Afin de répondre à cette problématique, l’entreprise fournit une base de données sur sa clientèle, contenant de nombreuses informations essentielles pour ce projet telles que le montant des crédits, mais aussi les revenus annuels, des données comportementales, des données provenant d'autres institutions financières, etc.

Aussi, afin de répondre à la demande de transparence des clients, l’entreprise met en place une API ***Flask*** permettant au modèle entraîné de donner une prédiction sur la probabilité de solvabilité d’un client de façon automatique.

Enfin, la création d’un Dashboard interactif via ***Streamlit*** permet un maximum de transparence dans la décision de la société et offre un service client illustratif de meilleure qualité à la clientèle.

II. Traitement des Données.

Les données disponible pour réaliser ce travail voir sur le site de [Kaggle](https://www.kaggle.com/c/home-credit-default-risk/data). Elles contiennent 307.511 clients et 122 variables, avec : des informations générales (âge, sexe, revenu), des informations bancaires sur les prêts précédents (nombre de prêts précédents, durée, solde bancaire) etc , sur chacun des clients.

**Sélection des données** : Garder des données avec moins 50% de complétion.

III. Modélisation

A. Division des Données.

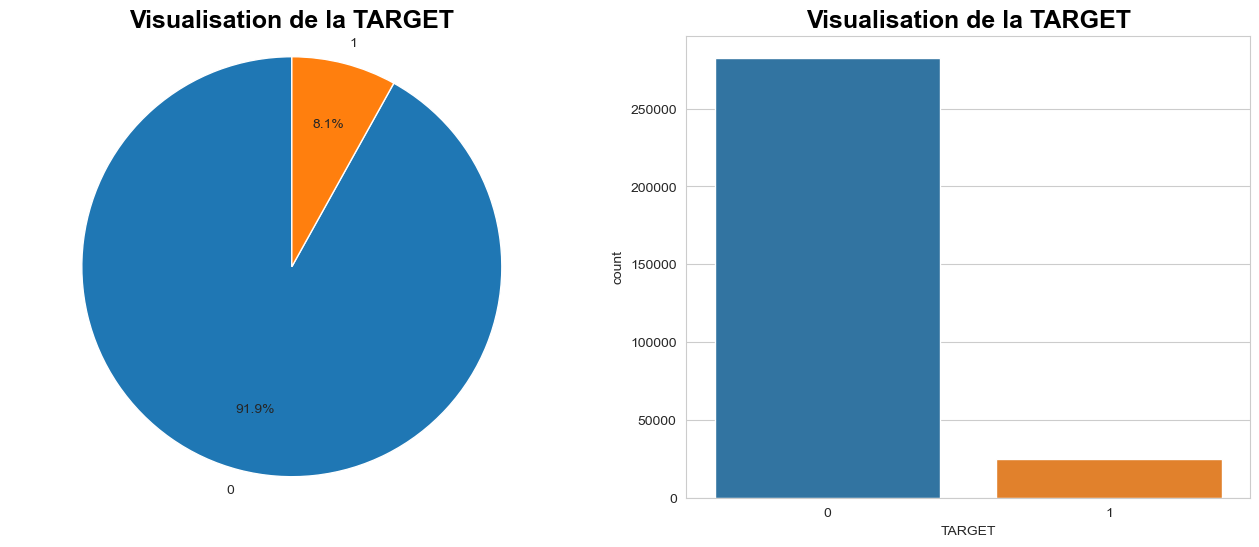
**Séparation des données** : Données d’entrainement (80%) et de test (20%).

**Imputation des données** :

* Pour les variables numériques : Remplacer les valeurs manquantes par la médiane.
* Pour les variables catégorielles : Remplacer les valeurs manquantes par la valeur la plus fréquente.

**Normalisation des données** : Standardiser les données numériques.

Problème de classification binaire, d’une base de données avec des classes déséquilibrées (« 0 » ---🡪91.9 % des individus seront capables de rembourser leurs crédits, contre « 1 » ---🡪8.1 % des individus ne seront pas capables de rembourser leurs crédits). ***Nous serons amenés à appliquer des différentes approches lors de l’étape de modélisation pour rééquilibrer les classes afin de réduire l’impact.***



1. Choix des Algorithmes.

Deux algorithmes ont été testés. Un algorithme de régression binomiale (régression logistique comme baseline) et un autre algorithme de gradient boosting Light-GBM.

1. Choix des Hyperparamètres de chaque Algorithme.

Rechercher des hyperparamètres pour que le modèle soit le plus adapté aux données. Le dataset train permet à un modèle d’ajuster sa prédiction par le biais d’une validation croisée. Durant cette étape, cet échantillon est séparé en plusieurs sous-ensemble sur lesquels les algorithmes apprennent pour ajuster le modèle. Le dataset test permet d’évaluer les performances d’un modèle en comparant les prédictions du modèle aux vraies valeurs de la variable à expliquer. La version du modèle avec le score d’erreur minimisé est retenu avec les paramètres optimums.

1. Rééquilibrage des données.

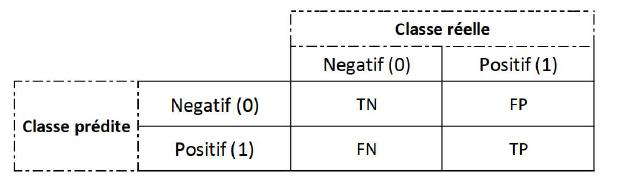
Déséquilibre dans les données entre les clients solvables et non solvables.

Pour résoudre ce problème, 4 approches sont appliquées au train dataset :

* Classe Weights : Méthode gérée par les modèles, permet de pénaliser les poids associés aux observations de la classe sur-représentés (ici solvable).
* Over-Sampling : Méthode qui duplique aléatoirement des données existantes de la classe sous représentée pour que chaque classe ait le même nombre de données que la classe sur-représentée à l’origine.
* SMOTE : Méthode qui crée de nouvelles données pour la classe sous-représentée à partir des données existantes pour que chaque classe ait le même nombre de données que la classe sur-représentée à l’origine.
* Under-Sampling : Méthode qui sélectionne une partie des observations sur-représentées afin que chaque classe ait le même nombre de données que la classe sous-représentée à l’origine.

1. Choix d’un Algorithme Optimal.
2. Fonction Coût Métier.

Pour déterminer l’algorithme optimal, entrainer chacun des modèles selon les différentes méthodes de rééquilibrage. Comparer les classes prédites aux classes réelles et créer une matrice de confusion pour représenter les erreurs commises par le modèle :



Classe «0» 🡺 clients solvables et négatifs au refus de l’accord de prêt.

Classe «1» 🡺 clients insolvables et positifs au refus de l’accord de prêt.

\* Accorder un crédit à un client ne peut pas le rembourser (**FN**) 🡺 Perte.

\* Accorder un crédit à un client qui le remboursera (**TN**) 🡺 Gain.

\* Ne pas accorder le prêt à un client qui ne peut pas rembourser (**TP**) 🡺 Neutre.

\* Ne pas accorder le prêt à un client qui peut le rembourser (**FP**) 🡺 Perte.

Les (**FN**) coûtent 10 fois plus cher qu’un (**FP**), la fonction de coût est :

2. Métriques d’Evaluation.

S’appuyer sur différentes métriques :

\* Accuracy : Total de prédiction correctes /Total de prédictions.

\* Precision : Vrais positifs /tous positifs, il donne indique sur les faux positifs.

\* Recall : Proportion dans laquelle notre modèle identifie correctement les

vrais positifs.

\* Le Score F1: Moyenne harmonique de la précision et du rappel. Un bon score F1 indique une bonne précision et une bonne valeur de rappel.

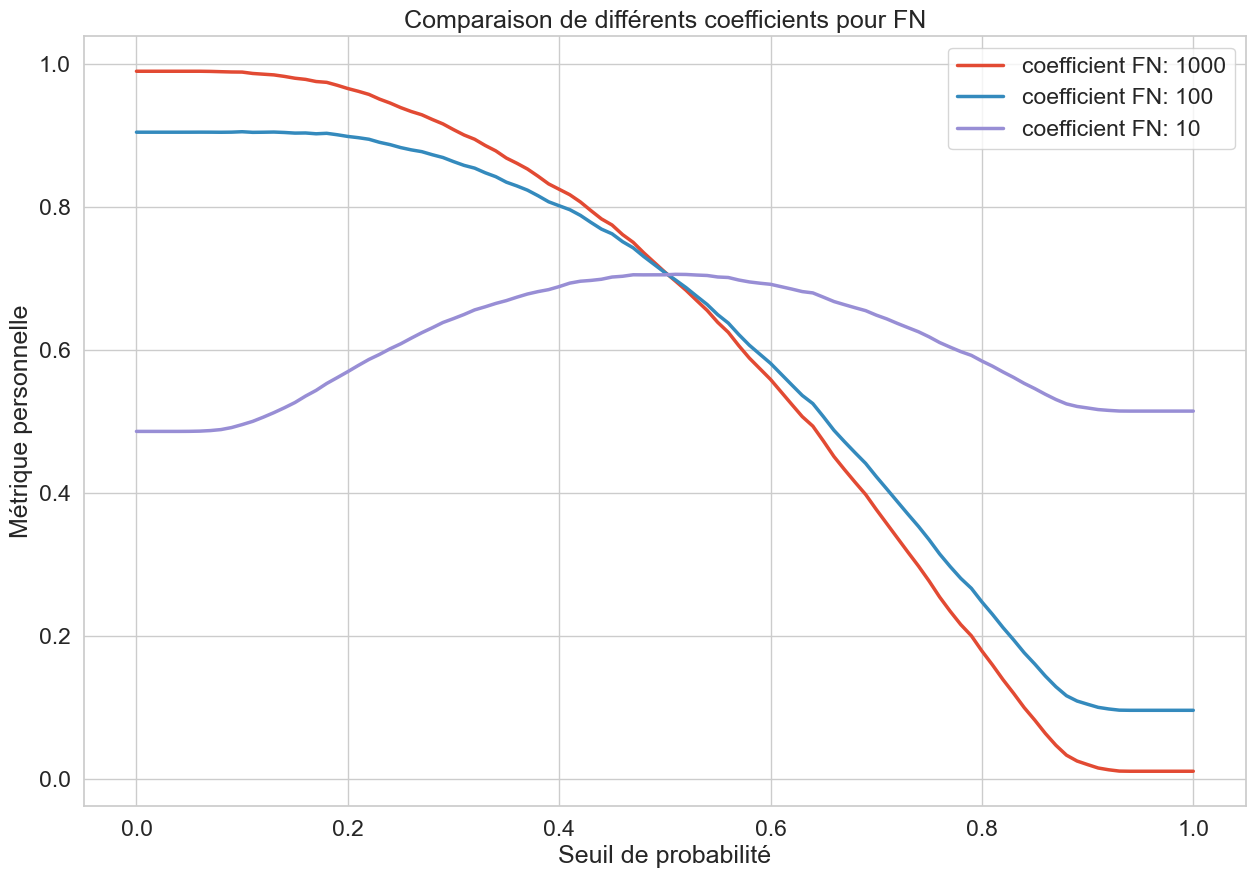
\* L’AUC (l’aire sous la courbe ROC). Plus l’AUC est élevée, plus la performance du modèle à distinguer les classes positives et négatives est bonne.

Le modèle utilisé pour l’outil doit maximiser l’AUC, le score F1, le Recall. Une banque cherche à ne pas accorder de prêt à des client qui ne peuvent pas le rembourser (TP à maximiser). La banque ne veut pas accorder un prêt à un client qui ne peut pas le rembourser (FN à minimiser).

Chaque modèle doit donc maximiser ce gain et remplir les contraintes des métriques d’évaluation. Toutes ces conditions sont validées par le modèle **Light Gradient Boosting Machine** (modèle robuste qui repose sur le principe d’arbre de décisions) avec la méthode Under-Sampling pour pallier le déséquilibre qui donc ici le modèle optimal.

3. Coefficient Optimal.

Le seuil de probabilité est la valeur de prédiction à partir de laquelle le modèle considère qu’un client est solvable ou non, compte tenu de la fonction gain utilisée. Des coefficients arbitraires ont précédemment été définis, mais sont-ils optimums ? Le coefficient de FN est celui qui a la plus grande importance dans notre métrique. Mais a-t-il le bon ordre de grandeur ? Pour vérifier cela, le score gain a été recalculé avec le modèle optimal pour différent seuil de probabilité en variant l’ordre de grandeur du coefficient FN :



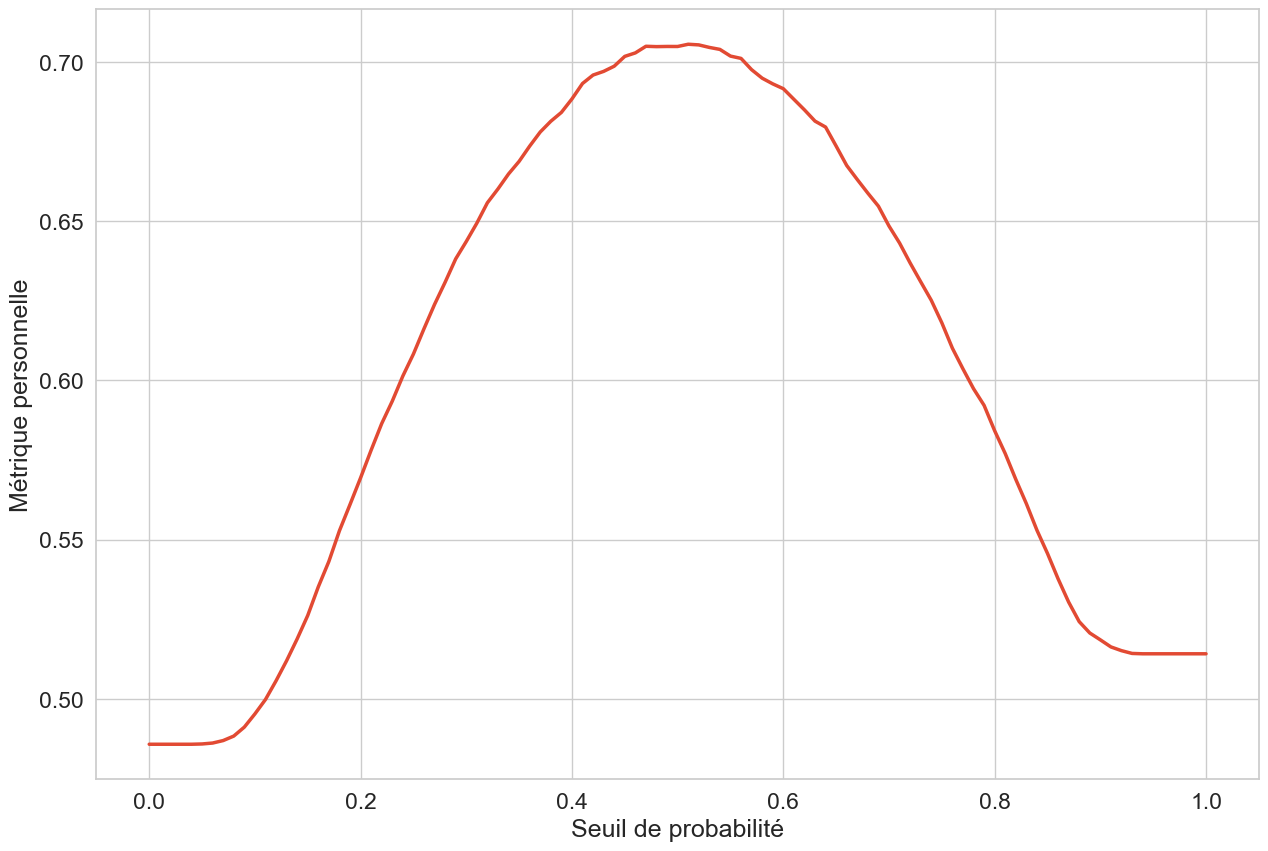
Lorsque le coefficient est 1000 pour les FN, le seuil optimal pour que la banque ait le maximum de gain d'argent se situe aux alentours d'une probabilité de 0,15 à partir de laquelle on peut voir que la courbe des gains décroit rapidement. Ce coefficient est trop strict car seuls les clients avec une probabilité de défaut de paiement < 0,15% se voient accepter leur prêt ce qui correspond à peu de personnes.

De même, lorsque l'on donne un coefficient de 100 pour les FN, on peut dire que la banque demeure rigide dans le choix des clients pour l'accord d'un prêt, car le seuil optimal pour que la banque ait le maximum de gain d'argent se situe aux alentours d'une probabilité de 0,2 et tous les clients avec une probabilité de défaut de paiement < 0,2% se verraient accepter leur prêt ce qui correspond toujours à peu de personnes.

Un coefficient de 10 pour les FN est mieux adapté, car le seuil optimal se situerait aux alentours de 0,51 ce qui est correct pour déterminer si oui ou non le prêt est accepté.

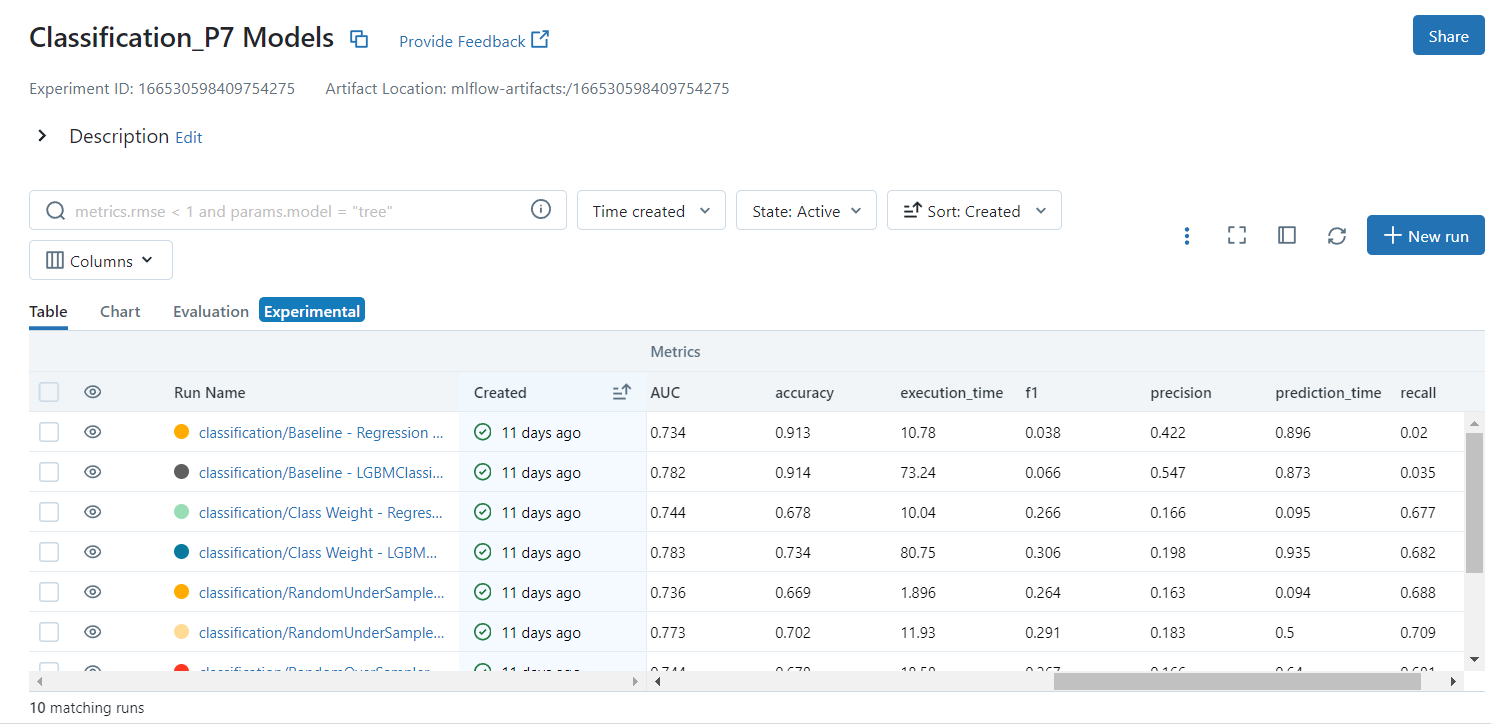
4. Seuil de Probabilité.

La métrique personnelle gain doit être maximisée pour que la banque gagne le plus d’argent et ainsi permettre de déterminer à partir de quel seuil de probabilité un client est considéré comme solvable ou non. Généralement le seuil de probabilité choisi par les modèles pour classer les individus est de 0.5 mais au vu de la courbe du score gain (avec un coefficient de 10), CE seuil est-il optimal pour que la banque maximise ses gains ?



Les résultats montrent un seuil de 0.51. Prédiction >= 0.51 => client non solvable. Prédiction < 0.51 => Client solvable.

1. Analyse des résultats obtenus : Tableau de synthèse de résultats.



**G. Interprétabilité globale et locale du modèle**

La méthode SHAP a été utilisée afin de comprendre comment chacune des variables du client ont influencé la prise de décision finale.

**G1. Variable locale :**

L’interprétabilité locale permet de connaître quelles variables ont influencé la prédiction d’un client en particulier. Par example, dans le cas d’un client donné, les variables en rose ont contribué à ne pas accorder le crédit (donc à augmenter le score). Exemple:EXT\_SOURCE\_3

Les variables en bleu ont contribué à accorder le crédit (donc à diminuer le score) Exemple: EXT\_SOURCE\_1.

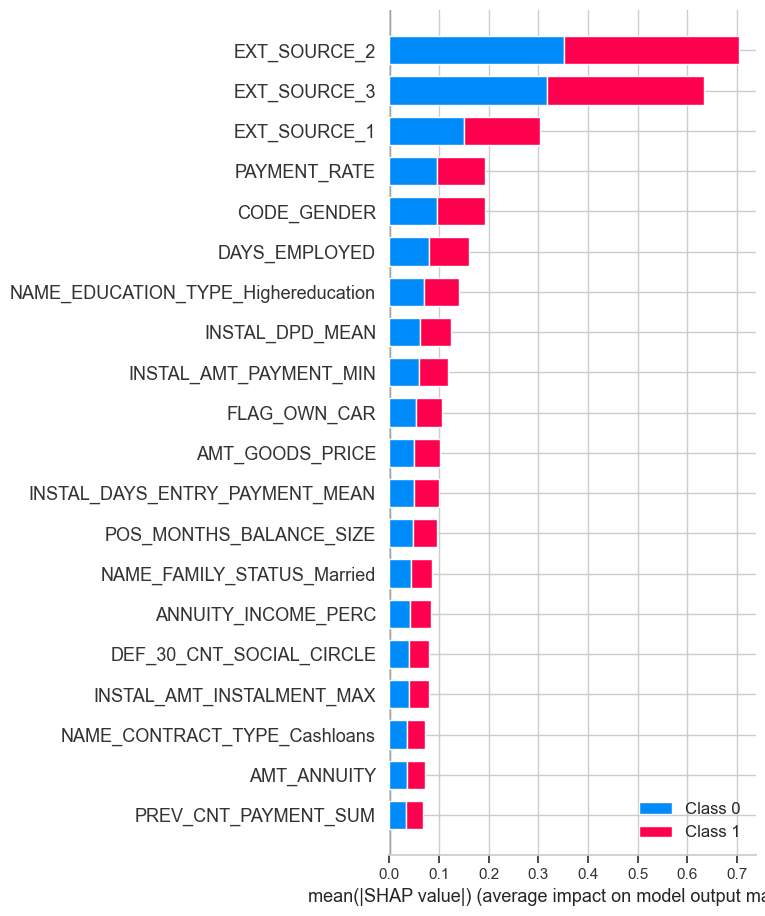
Une image contenant Site web

Description générée automatiquement

**G2. Variable globale :**

L’interprétation Globale permet de savoir quelles variables ont le plus d’importance sur les prédictions pour l’ensemble des clients.

Nous pouvons voir ici que les variables *EXT\_SOURCE\_2*, *EXT\_SOURCE\_3*, *EXT\_SOURCE\_1* et *PAYMENT\_RATE* sont les quatre variables les plus importantes dans la prédiction.



H. API & DASHBOARD

H1.API

Pour ce projet, une api de prédiction a été déployée sur Heroku. Lorsque nous choisissons un identifiant de client, l’API va chercher dans un fichier de données toutes les valeurs des variables qui lui sont associées. L’API va ensuite prédire et afficher le score associé au client à l’aide du meilleur modèle que nous lui avons également donné de façon automatique.

H2. DASHBOARD

Afin d’avoir une meilleure interpretabilité des prédictions faites par le modèle entraîné, et une meilleure amélioration de la connaisance client des chargés de la clientèle, un dashboard interactif a été construit sous ***Streamlit*** en suite déployé par ***Heroku*** afin de réaliser la mise en disposition au publique.

I. Les limites et les améliorations possibles

La modélisation a été effectué sur la base d’une métrique personnelle créée pour répondre au mieux au besoin de gain d’argent d’une banque. Les coefficients de cette métrique ont été choisis arbitrairement selon le bon sens. L’axe principal d’amélioration serait donc de définir plus précisément ces coefficients associés à chaque combinaison classe prédite/classe réelle car le modèle déterminé ici ne sera pas obligatoirement le meilleur.

Le modèle créé ici tend à être un modèle “éthique“ car il ne tient pas compte de variables telles que le genre du client qui peut être une information discriminante dans l’attribution d’un prêt. Cependant toutes les variables discriminantes n’ont peut-être pas été enlevées. Il serait intéressant de comparer un modèle “éthique“ à un modèle prenant en compte toutes les variables pour voir si le fait de supprimer ces variables discriminantes amène une perte d’information et de précision pour les prédictions ainsi qu’une perte de rentabilité pour la banque.

Le traitement des données a été réalisé de façon superficielle car une seule table de données a été prise en compte alors que les données sur les précédents prêts étaient disponibles. Il y a donc très probablement la possibilité d’améliorer la modélisation en utilisant d’autres features provenant des autres bases de données ainsi qu’en créant de nouvelles features grâce à l’équipe métier.

J. Analyse du data drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5







Le seuil de détection de la dérive de l’ensemble des données est de 0.5.

La dérive est détectée pour 7.5% des colonnes (9 sur 100).