

Introduction

Le projet vise à développer un modèle de scoring pour prédire la probabilité de défaut des clients demandant un crédit. Ce modèle servira à faciliter la prise de décision des conseillers, en l'associant à un tableau de bord interactif permettant d'expliquer de manière transparente les décisions d'octroi de crédit.

Collecte des données

Les données proviennent de plusieurs sources internes à l'entreprise, regroupées dans dix fichiers distincts, dont les principaux sont `application_train.csv` pour l'entraînement et `application_test.csv` pour le test.

D'autres fichiers fournissent des informations supplémentaires sur les crédits passés, les paiements des cartes de crédit et les informations démographiques des clients.

Prétraitement des données

Le prétraitement a consisté à :

- **Gestion des valeurs manquantes** : Certaines colonnes contenaient un pourcentage significatif de valeurs manquantes, que nous avons traité en remplaçant par 0 ou en imputant par la moyenne.
- **Nettoyage des données** : Certaines colonnes contenaient des valeurs aberrantes, telles que `DAYS_EMPLOYED` avec des valeurs correspondant à des milliers de jours (indiquant des erreurs de saisie). Ces anomalies n'ont pas été corrigées ou supprimées.
- **Transformation des données** : Les variables catégorielles ont été encodées, et certaines colonnes numériques ont été transformées pour obtenir des données plus homogènes.

Séparation des données

Les données ont été divisées en trois ensembles : un ensemble d'entraînement (60 %), et un ensemble de test (40 %). La répartition a été faite de manière à maintenir la distribution originale de la cible (`TARGET`) dans chaque ensemble.

Méthodologie d'entraînement du modèle

1. Standardisation des données d'entraînement (StandardScaler)

Le StandardScaler transforme les caractéristiques de telle sorte à ce que leur moyenne soit nulle et que leur écart type soit égal à 1.

Standardiser les données avant d'entraîner les modèles donne plusieurs avantages :

- 1) Convergence plus rapide : Pour des algorithmes basés sur la descente de gradient (comme la régression logistique), elle peut aider à accélérer la convergence.
- 2) Évite la dominance des caractéristiques : La mise à l'échelle garantit que toutes les caractéristiques ont le même poids initial.
- 3) Importance des caractéristiques : Si les caractéristiques sont mises à l'échelle, leurs poids peuvent être comparés plus facilement et directement.

2. Validation croisée (StratifiedKFold) – Nombre de folds = 5

Dans des problèmes de classification où il y a un important déséquilibre de classe sur la target (c'est le cas dans ce projet), il est essentiel que lorsqu'une validation croisée s'effectue, chaque fold soit représentatif du dataset complet.

C'est pour ça qu'il a été décidé de réarranger les données de manière à ce que chaque fold conserve les proportions similaires d'échantillons par rapport à l'ensemble du dataset. Cette technique s'appelle la stratification et c'est pour ça que la méthode "StratifiedKFold" de scikit-learn a été utilisée.

Utilisation de SMOTE avec une pipeline.

Éviter la fuite de données : Lorsque on applique SMOTE avant de diviser vos données en ensembles d'entraînement et de test, on risque d'introduire une fuite de données. Cela signifie que des informations du jeu de test peuvent influencer le modèle pendant l'entraînement, ce qui biaise les résultats.

. En utilisant une pipeline, on s'assure que SMOTE est appliqué uniquement sur les données d'entraînement pendant la validation croisée.

Pipeline imblearn

L'utilisation de la pipeline imblearn pour Gestion des déséquilibres de classes : imblearn est spécialement conçu pour traiter les ensembles de données avec des classes déséquilibrées. Il permet d'appliquer des techniques de rééchantillonnage comme le suréchantillonnage des classes minoritaires et le sous-échantillonnage des classes majoritaires, ce qui améliore la performance des modèles de machine learning.

Cross-validation

Cross-validation : La pipeline permet de réaliser une validation croisée de l'ensemble des étapes, ce qui permet de régler les paramètres des différentes étapes de manière cohérente et d'améliorer la performance globale du modèle.

Intégration de l'hyperparamètre seuil (threshold)

Les modèles de classification donnent en sortie une probabilité permettant de conclure si oui ou non, l'accord d'un crédit à un client peut s'effectuer. En général, la valeur par défaut du seuil est de 0,5.

Cependant, il peut arriver que cette valeur puisse ne pas être optimale. En décidant d'ajuster ce seuil, il est possible d'optimiser le nombre de faux positifs et de faux négatifs.

Enregistrement des résultats avec MLflow

MLflow est une plateforme qui gère le cycle de nos modèles. Cela permet de suivre et de comparer les différents essais et modèles, facilitant la reproductibilité et le déploiement. Dans ce projet,

MLflow enregistre les hyperparamètres, les sorties ainsi que les modèles sous forme de format pickle.

Analyse du déséquilibre

Le jeu de données présente un fort déséquilibre avec environ 92 % des prêts remboursés à temps (classe 0) et seulement 8 % de défauts de paiement (classe 1). Cela nécessite un traitement spécifique pour garantir que le modèle ne favorise pas la classe majoritaire.

Techniques utilisées

Pour remédier à ce déséquilibre, plusieurs techniques ont été testées :

- + **Imbalanced-Learn** est une bibliothèque Python open source conçue pour traiter les problèmes de déséquilibre des classes dans les ensembles de données de Machine Learning.

- + **Suréchantillonnage** : Nous avons utilisé la technique de SMOTE (Synthetic Minority Over-sampling Technique) pour générer artificiellement des exemples de la classe minoritaire.

- +F-beta

Le F-beta score est une métrique utilisée pour évaluer la performance des modèles de classification, en particulier dans les cas où il y a un déséquilibre entre les classes. Il s'agit de la moyenne harmonique pondérée de la précision (precision) et du rappel (recall), atteignant sa valeur optimale à 1 et sa pire valeur à 0.

La formule du F-beta score est la suivante :

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

F-beta Score avec beta=10

Lorsque $\beta = 10$, le F-beta score accorde beaucoup plus d'importance au rappel qu'à la précision. Cela signifie que le modèle est évalué principalement sur sa capacité à identifier correctement les exemples positifs, même si cela se fait au détriment de la précision.

Fonction coût métier

La fonction coût utilisée repose sur l'importance de minimiser les erreurs de classification pour la classe minoritaire (clients en défaut). Dans le cadre de cette problématique, les erreurs de faux négatifs (clients risqués classés comme sûrs) ont un coût plus élevé que les faux positifs.

Le score F-beta est une métrique utilisée pour évaluer la performance des modèles de classification en combinant la précision et le rappel. Lorsque $\beta=10$, cela signifie que l'on accorde beaucoup plus d'importance au rappel par rapport à la précision

Algorithme d'optimisation

Régression logistique

Simplicité et Interprétabilité : Facile à comprendre et à interpréter, ce qui en fait un bon choix pour les problèmes de classification binaire.

Efficacité pour la Classification Binaire : Particulièrement efficace pour les problèmes de classification binaire.

Probabilités Prédictives : Fournit des probabilités prédictives pour chaque classe, permettant une meilleure prise de décision.

LightGBM

Performance et Rapidité : Très rapide et efficace, même sur de grands ensembles de données.

Scalabilité : Peut gérer de grands volumes de données et s'adapte bien aux tâches complexes.

Flexibilité : Offre de nombreuses fonctionnalités avancées pour améliorer les performances.

XGBoost

Vitesse d'Exécution : Connu pour son exécution rapide et efficace.

Régularisation Intégrée : Aide à prévenir le surajustement grâce à des techniques de régularisation intégrées.

Gestion des Valeurs Manquantes : Capable de gérer les valeurs manquantes et les valeurs aberrantes, rendant robuste dans diverses situations.

Métrique d'évaluation

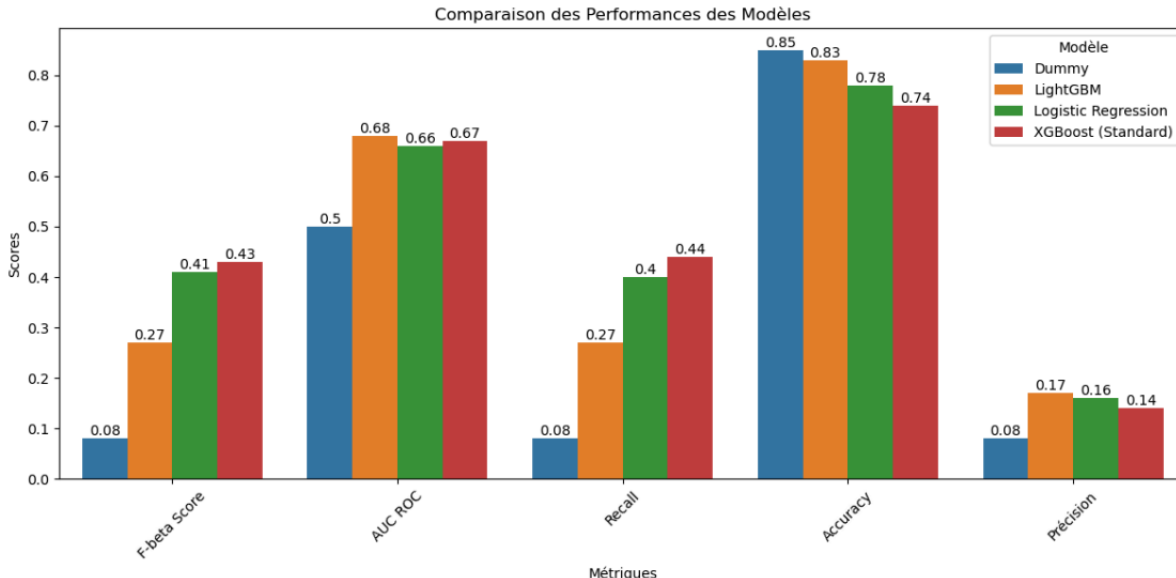
Les principales métriques utilisées sont

+ **AUC** a été privilégiée car elle permet d'évaluer la capacité du modèle à discriminer les deux classes (remboursé/défaut) sans être affectée par le déséquilibre des classes.

+ F-beta score accorde beaucoup plus d'importance au rappel qu'à la précision. Cela signifie que le modèle est évalué principalement sur sa capacité à identifier correctement les exemples positifs, même si cela se fait au détriment de la précision

Tableau de synthèse des résultats

Présentation des résultats

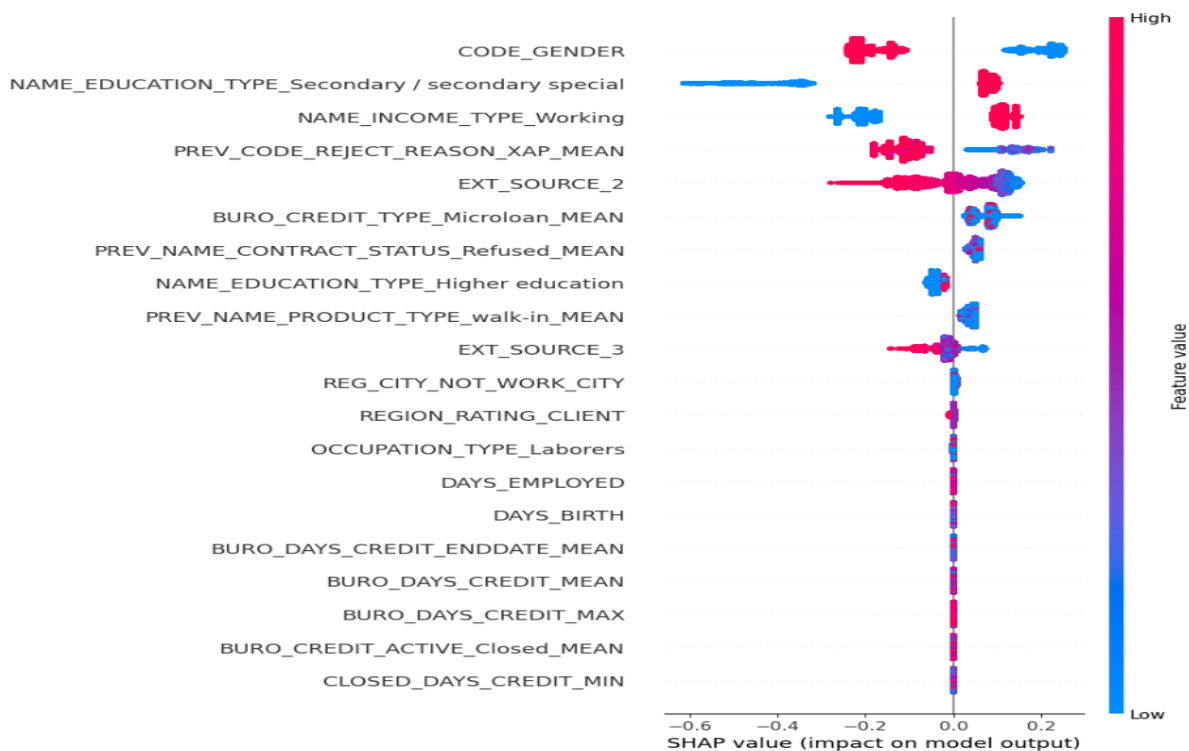


Interprétation des résultats

Les résultats montrent que le modèle XGBoost surpasse les autres modèles en termes de capacité à discriminer les classes (AUC). Bien que la précision soit légèrement inférieure, l'AUC plus élevée indique que le modèle est mieux adapté pour prédire les clients en défaut.

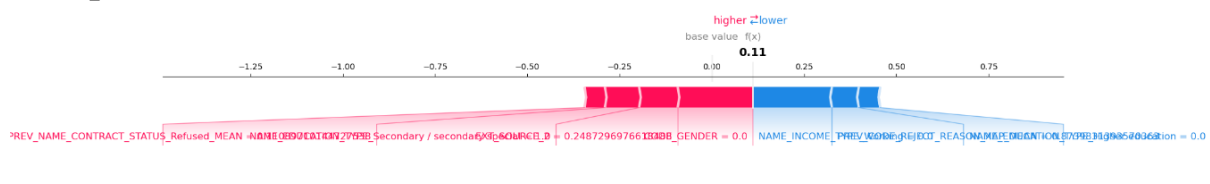
Interprétabilité globale et locale du modèle

Interprétabilité globale



L'interprétabilité globale du modèle a été analysée à l'aide des valeurs SHAP (SHapley Additive exPlanations), qui permettent d'identifier les features les plus influentes dans la décision du modèle. Les variables **EXT_SOURCE_3**, **DAYS_BIRTH** et **AMT_CREDIT** se sont révélées être parmi les plus importantes pour la prédiction du modèle.

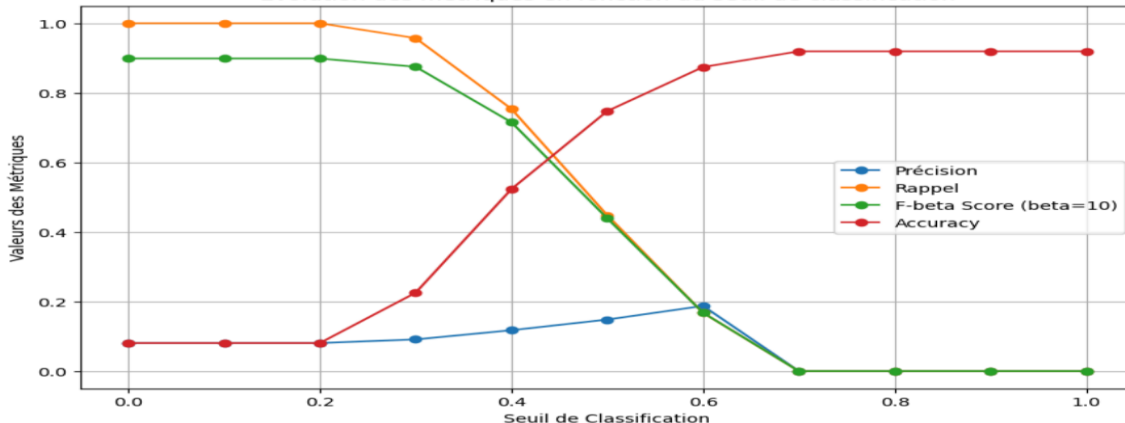
Interprétabilité locale



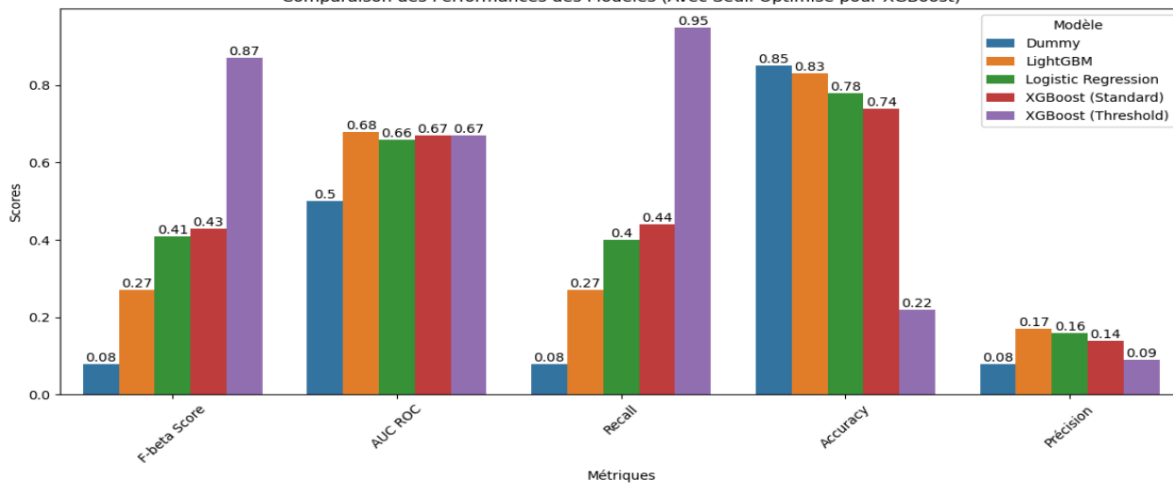
L'interprétabilité locale a été réalisée avec des visualisations SHAP, permettant d'expliquer les prédictions au niveau individuel. Ces visualisations offrent une transparence sur les raisons derrière une prédiction donnée, en identifiant les features spécifiques qui ont contribué à la décision.

Les limites et les améliorations possibles

Évolution des métriques en fonction du seuil de classification



Comparaison des Performances des Modèles (Avec Seuil Optimisé pour XGBoost)



Le seuil de probabilité est un concept clé dans les modèles de classification, notamment lorsqu'on utilise des métriques comme le score F-beta.

+ Analyse du seuil optimal :

- Rappel élevé : Pour minimiser les FN, le rappel doit être élevé, car il capte la capacité du modèle à identifier correctement tous les positifs. Le rappel est maximal pour un seuil de 0 et diminue rapidement après 0,4.
- Précision et rappel : La précision et le rappel sont généralement en tension. Un seuil bas conduit à un rappel élevé, mais à une précision plus faible (car plus de faux positifs sont prédits).
- F-beta (vert) : Le F-beta score avec $\beta=10$ pondère beaucoup plus le rappel que la précision, ce qui est exactement en ligne avec votre objectif de minimiser les FN. Le F-beta score est maximisé autour du seuil de 0,1 à 0,4.
- Accuracy (rouge) : Cette métrique n'est pas aussi pertinente dans ce cas, car elle ne pondère pas spécifiquement les FN par rapport aux FP.

Dataset Drift

Le Dataset Drift se produit lorsque les statistiques des données changent entre le moment où un modèle est formé et le moment où il est utilisé. les points clés :

- Détection de Drift : Un drift a été détecté pour 56,658% des colonnes .
- Seuil de Détection : Le seuil de détection du drift est fixé à 0,5.
- Colonnes Driftées : 451 colonnes ont été identifiées comme ayant un drift.
- Tests Statistiques : Des tests statistiques comme le test de Kolmogorov-Smirnov (K-S) sont utilisés pour détecter le drift.

Dataset Drift

Dataset Drift is detected. Dataset drift detection threshold is 0.5

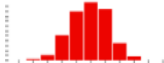
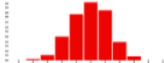
796
Columns

451
Drifted Columns

0.567
Share of Drifted Columns

Data Drift Summary

Drift is detected for 56.658% of columns (451 out of 796).

Q Search X						
Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> REFUSED_HOUR_APPR_PROCESS_START_MEAN_REFUSED	num			Detected	K-S p_value	0.011163

Cela signifie que la distribution des données en entrée du modèle, contre celle des données actuelles a changé au fil du temps. Comme les données catégorielles ont été encodées à l'aide de la méthode OneHot, toutes les features du dataset sont désormais numériques. Le test utilisé est donc la distance de Wasserstein (normalisée) (WD). Il s'agit de la méthode par défaut pour les données tabulaires numériques supérieures à 1000 lignes. WD mesure l'effort nécessaire pour transformer une distribution en une autre. WD normalisée indique le nombre d'écarts types en moyenne qu'il faudrait déplacer pour chaque ID du groupe actuel pour qu'il corresponde au groupe de référence. Le seuil de détection du data drift de 0.1 signifie donc qu'un changement dans la taille des écarts-types de 0.1 est significatif. Pistes à envisager pour gérer le data drift du modèle en production :

- 1) Mettre en place des contrôles sur la qualité et intégrité des données
- 2) S'assurer que les relations entre les features ou entre les features et la target n'ont pas changé
- 3) Réentraîner le modèle lorsque de nouveaux labels seront disponibles
- 4) Recalibrer ou reconstruire le modèle, créer un modèle spécifique pour les segments qui échouent
- 5) Modifier le traitement des valeurs aberrantes
- 6) Si le modèle est jugé peu performant, utiliser plutôt un ensemble de règles de décision moins précises mais plus robuste