Prêt à dépenser

Implémentez un modèle de scoring

Note Méthodologique

PROJET 7 : OpenClassRoom – Data Scientist

Étudiant : Aymen TLILI

IMPLEMENTEZ UN MODELE DE SCORING

Introduction

Dans le cadre de notre collaboration avec Prêt à dépenser, nous développons un outil de scoring crédit pour évaluer la solvabilité des clients. L'entreprise nous a fourni des données détaillées, incluant les montants des crédits et les revenus annuels, qui seront essentielles pour classifier les demandes de crédit.

Contexte et Objectifs

Prêt à dépenser développe un modèle de scoring pour évaluer et justifier la solvabilité des clients, augmentant ainsi la transparence dans les décisions de crédit.

Dans cette optique, nous avons conçu et mis en place un dashboard interactif avec Streamlit, facilitant pour les chargés de relation client la communication des décisions de crédit. Ce dashboard permet également aux clients de consulter et d'explorer facilement leurs informations personnelles, renforçant ainsi la transparence et la confiance.

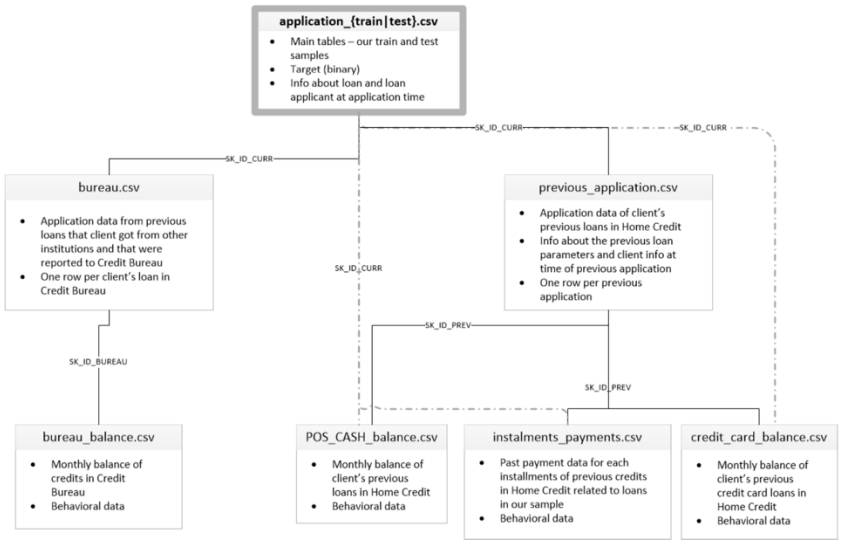
Méthodologie : Nos techniques avancées de data science ont préparé et optimisé les données pour le scoring. Pour explorer nos méthodes et résultats en détail, consultez notre page ***GitHub*** et accédez à notre dashboard interactif sur ***Streamlit*** Cloud ou ***Heroku***.

Table of Contents

[PARTIE 1] — METHODOLOGIE D'ENTRAINEMENT DU MODELE

A. Organisation des Données

Les données sont réparties en 7 fichiers distincts, interconnectés comme indiqué dans le schéma ci-joint.



La table principale, « application », est scindée en deux jeux de données :

- "application train" - comprenant 307 511 clients avec des données décisionnelles connues de "Prêt à Dépenser".
- "application test" - où la décision de crédit reste inconnue.

B. Compléments des Données

- Tables 'bureau' et 'balance_bureau' : Contiennent des informations sur les crédits antérieurs pris dans d'autres institutions financières.
- Table 'Previous_application' : Détaille les antécédents de crédit des clients chez "Prêt à Dépenser".

C. Importance du Preprocessing

- Le succès de notre algorithme dépend largement du traitement préalable des données, souvent non adaptées ou corrompues par des erreurs humaines ou techniques. Ce preprocessing est essentiel pour corriger les données incomplètes, erronées, ou bruitées.

D. Data Cleaning

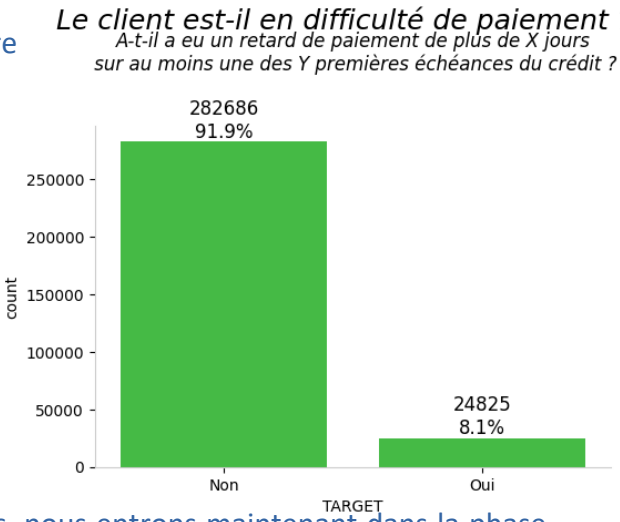
- Chaque table a été nettoyée séparément : Suppression des variables avec plus de 0,1% de valeurs manquantes.
- Remplacement des données manquantes inférieures à 0,1% par la médiane pour les variables numériques et par le mode pour les catégorielles.

E. Data Reduction

- La sélection des variables a été effectuée pour réduire le nombre de variables d'entrée à celles essentiellement importantes pour le modèle, en utilisant des méthodes de filtrage, d'encapsulation, et intégrées Et une étape de features sélection ou on passe de 48 à 22 variables.

F. Data Transformation : Les modifications structurelles des données incluent :

- La discrétisation des variables continues pour réduire les modalités et éliminer les valeurs aberrantes.
- La normalisation et la standardisation pour ajuster les échelles des données numériques et optimiser les performances du modèle.

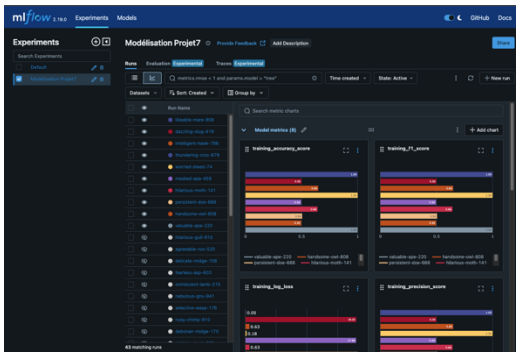


G. Résultat Final

- Le jeu de données d'entraînement finalisé compte 307 511 clients avec 22 variables descriptives.

F. Transition vers la Modélisation

- Avec la finalisation de notre préparation des données, nous entrons maintenant dans la phase de modélisation, où nous appliquerons une classification binaire pour distinguer les clients à risque de ceux sans risque. Pour cela, nous sélectionnons cinq modèles principaux : Régression Logistique, Dummy Classifier, Random Forest, XGBoost, et LIGHTGBM, choisis pour leur efficacité dans des scénarios similaires.



- Nous intégrons MLflow et ngrok dans notre environnement de développement sous Google Colab, pour un suivi précis et une gestion efficace de nos expériences. MLflow nous aide à documenter et comparer les performances de nos modèles, tandis que ngrok nous permet d'accéder à notre interface de modélisation à distance. Cela facilite le partage et la collaboration, crucial dans notre processus de développement agile.
- Les modèles sont ensuite sauvegardés sous format 'pickle' pour assurer leur réutilisation et leur déploiement facile dans divers environnements, maximisant ainsi l'efficacité et la portabilité de notre solution de scoring de crédit.

```
import pickle
save_path = '/content/drive/My Drive/Projets Master/Projet 7/model_streamlit.pkl'
# Sauvegarde du modèle
with open(save_path, 'wb') as file:
    pickle.dump(model, file)
print(f"Le modèle a été sauvegardé dans : {save_path}")
```

[PARTIE 2] — TRAITEMENT DU DESEQUILIBRE DE CLASSE

A. Première Phase de Modélisation et Défi du Déséquilibre des Classes

- Suite à une première phase de modélisation avec nos cinq modèles, comprenant des techniques de normalisation et de standardisation, nous avons observé des résultats

prometteurs mais également un problème majeur : un déséquilibre significatif des classes dans notre dataset. Pour y remédier, nous avons opté pour trois méthodes de rééquilibrage des données :

B. Techniques de Rééquilibrage des Données

- SMOTE (Synthetic Minority Over-sampling Technique) est une méthode de sur-échantillonnage qui génère des exemples synthétiques de la classe minoritaire en interpolant entre des échantillons existants. Cette technique augmente la diversité sans répéter les données, aidant à éviter le surapprentissage tout en améliorant la représentativité de la classe minoritaire.
- Random Over Sampling (ROS) Description : Cette approche consiste à dupliquer aléatoirement des échantillons de la classe minoritaire jusqu'à équilibrer les proportions avec la classe majoritaire. Simple et rapide, elle présente toutefois un risque de surapprentissage en augmentant la redondance dans les données.
- Random Under Sampling (RUS) Description : En contraste avec le ROS, le RUS élimine aléatoirement des échantillons de la classe majoritaire pour équilibrer le dataset. Cette méthode peut réduire les coûts de traitement et le temps d'apprentissage, mais risque de perdre des informations précieuses si les échantillons supprimés contiennent des caractéristiques uniques importantes.

C. Conclusion sur le Choix des Techniques

- Chaque méthode de rééquilibrage présente des avantages et inconvénients spécifiques. Leur sélection doit être judicieusement effectuée en fonction des caractéristiques du dataset et des objectifs de la modélisation. Une évaluation rigoureuse est nécessaire pour assurer que le choix final maximise l'efficacité du modèle de scoring, tout en préservant l'intégrité et la richesse des données.

[PARTIE 3] — FONCTION COUT ET OPTIMISATION DES HYPERPARAMETRES

A. Sélection des Métriques de Performance

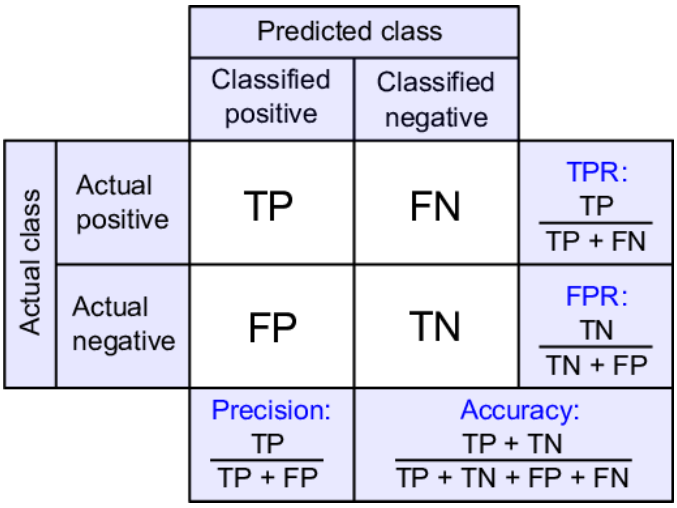
- Afin d'évaluer efficacement la performance de nos modèles de classification, nous utilisons des métriques précises qui reflètent différents aspects de l'accuracy et de la capacité de généralisation :

a. **Accuracy :** Mesure la proportion d'observations correctement classées par le modèle. Calculée comme le ratio des observations correctement classées sur le nombre total d'observations.

b. **Recall :** Indique le pourcentage de positifs réels correctement identifiés par le modèle (nombre de vrais positifs / nombre total de vrais positifs et faux négatifs).

c. **Precision :** Détaille le nombre de prédictions positives qui sont effectivement correctes (nombre de vrais positifs / nombre total de positifs prédits).

d. **F1 Score :** Combine la précision et le rappel en une seule métrique, la moyenne harmonique des deux, fournissant un équilibre entre précision et rappel. Cette métrique est particulièrement utile lorsque les classes sont déséquilibrées.



e. **F2 Score :** Similaire au F1 Score mais donne plus de poids au rappel qu'à la précision, utile lorsque la minimisation des faux négatifs est prioritaire.

f. **Matrice de Confusion:** Cet outil est fondamental pour évaluer la performance de notre modèle. Elle illustre les vrais positifs, vrais négatifs, faux positifs et faux négatifs. La figure jointe explique comment lire cette matrice, mettant en lumière l'importance du rappel dans notre contexte de classification

B. Optimisation de la Fonction Coût et des Hyperparamètres

a. Fonction Coût avec l'Indice de Fowlkes-Mallows

- Dans le cadre de notre projet, l'optimisation de la fonction coût est cruciale pour aligner les performances du modèle avec les objectifs métier. Nous intégrons l'Indice de Fowlkes-Mallows (FMI) comme partie de notre fonction coût. Le FMI est une mesure qui combine la précision et le rappel pour évaluer la qualité de la classification binaire. Il est particulièrement adapté aux contextes où l'équilibre entre la sensibilité (rappel) et la spécificité (précision) est essentiel. Le FMI est calculé en prenant la racine carrée du produit du taux de vrais positifs et de la précision positive, offrant ainsi un compromis équilibré entre ces deux métriques.

b. Optimisation des Hyperparamètres

- L'ajustement des hyperparamètres est réalisé à travers la méthode de Grid Search, une technique de validation croisée qui évalue systématiquement différentes combinaisons d'hyperparamètres pour identifier celle qui optimise le mieux la fonction coût. Cette approche permet de tester un large éventail de valeurs pour chaque hyperparamètre et de sélectionner la combinaison qui maximise le FMI, assurant ainsi une optimisation précise et une performance améliorée du modèle.

[PARTIE 4] — SYNTHESE DES RESULTATS

- Le tableau ci-dessous illustre l'impact des différentes méthodes de prétraitement, de standardisation et de rééquilibrage des classes sur la performance du modèle Random Forest. Nous avons exploré plusieurs techniques, MinMaxScaler et StandardScaler, ainsi que des méthodes de rééquilibrage des classes telles que ROS (Random Over Sampling), RUS (Random Under Sampling), et SMOTE (Synthetic Minority Over-sampling Technique). Chaque approche a été évaluée sur plusieurs métriques clés que nous avons définies précédemment.

Modèles/ Métriques	Accuracy	Precision	Recall	F1 Score	F2 Score	ROCAUC	FMI
Régression Logistique							
StandardScaler	0.918	0.5	0.000638	0.00127	0.0008	0.5	0.0179
ROS	0.657	0.141	0.628	0.23	0.37	0.644	0.297
RUS	0.659	0.141	0.625	0.23	0.37	0.644	0.297
Smote	0.658	0.141	0.628	0.23	0.37	0.644	0.298
DummyClassifier							
MinMaxScaler	0.851	0.0802	0.0788	0.0795	0.079	0.499	0.0795
StandardScaler	0.851	0.083	0.0823	0.0826	0.082	0.501	0.0826
ROS	0.498	0.0803	0.493	0.138	0.24	0.496	0.199
RUS	0.501	0.0808	0.492	0.139	0.24	0.497	0.199
Smote	0.498	0.0801	0.492	0.138	0.24	0.495	0.198
RandomForest							
MinMaxScaler	0.918	0.5	0.00159	0.00318	0.002	0.501	0.0282
StandardScaler	0.918	0.471	0.00128	0.00254	0.0016	0.501	0.0245
ROS	0.918	0.35	0.00685	0.0134	0.0085	0.503	0.049
RUS	0.667	0.146	0.634	0.237	0.38	0.652	0.304
Smote	0.916	0.34	0.0249	0.0463	0.031	0.51	0.0919
XGBoost							
MinMaxScaler	0.918	0.429	0.0179	0.0343	0.022	0.508	0.0875
StandardScaler	0.918	0.429	0.0179	0.0343	0.022	0.508	0.0875
ROS	0.717	0.156	0.562	0.245	0.37	0.646	0.297
RUS	0.65	0.141	0.645	0.231	0.38	0.648	0.301
Smote	0.918	0.432	0.0188	0.0361	0.023	0.508	0.0902
LightGBM							
MinMaxScaler	0.919	0.561	0.0059	0.0117	0.0074	0.503	0.0575
StandardScaler	0.919	0.576	0.00606	0.012	0.0076	0.503	0.0591
ROS	0.685	0.154	0.637	0.249	0.39	0.664	0.314
RUS	0.664	0.148	0.655	0.242	0.39	0.66	0.311
Smote	0.918	0.455	0.00813	0.016	0.01	0.504	0.0608

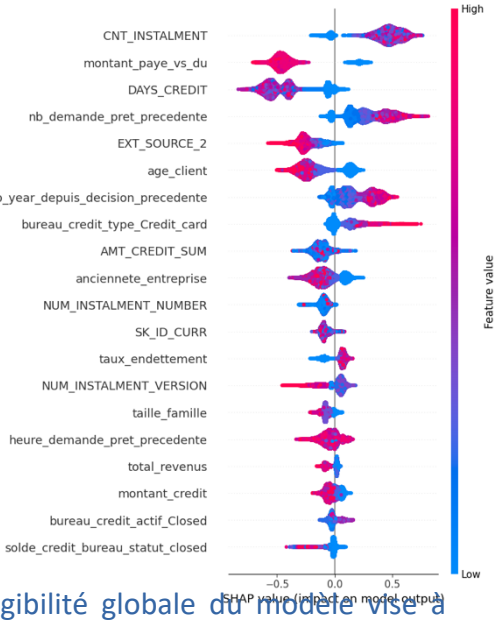
- Les résultats montrent une performance similaire entre les méthodes StandardScaler et MinMaxScaler. En raison de temps de calcul prolongés pour XGBoost et Random Forest, et de performances relativement inférieures avec SMOTE, nous avons choisi de nous concentrer sur l'optimisation de LightGBM avec ROS et RUS, qui offrent une efficacité comparable sans lourdeur computationnelle.

[PARTIE 5]— ANALYSE GLOBALE ET LOCALE DU MODELE

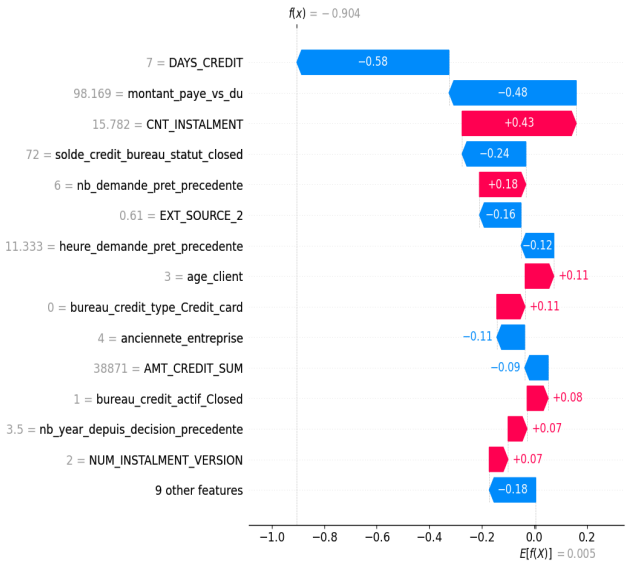
L'Importance des Caractéristiques L'interprétation des modèles de machine learning est cruciale. Identifier les variables pertinentes est essentiel pour comprendre les mécanismes sous-jacents à la prise de décision du modèle. Pour cela, nous utilisons le Notebook 5, dédié à l'analyse de l'importance des caractéristiques. Ce processus nous permet d'évaluer l'impact de chaque variable sur le modèle, en utilisant des méthodes telles que le calcul des valeurs de Shapley et d'autres indicateurs d'erreur de référence.

A. Interprétation Locale

L'intelligibilité locale se concentre sur l'explication spécifique des prédictions pour des individus donnés. Cela permet de comprendre pourquoi une demande de prêt d'un client particulier a été approuvée ou rejetée. L'utilisation des valeurs de Shapley, par exemple, aide à expliquer l'impact de chaque caractéristique sur la décision spécifique, en considérant toutes les permutations possibles de présence des variables pour garantir une comparaison équitable.



B. Interprétation Globale



À l'inverse, L'intelligibilité globale du modèle vise à clarifier quelles variables sont les plus influentes en moyenne, affectant le comportement général du modèle. Par exemple, en analysant les résultats du Notebook 5, nous déterminons quelles caractéristiques sont cruciales pour le modèle d'allocation de prêt. Cela inclut l'évaluation de l'importance des variables par des méthodes statistiques et computationnelles qui mesurent leur impact global sur les performances prédites du modèle.

[PARTIE 6] — ANALYSE DU DATA DRIFT

A. Méthodologie

Dans cette partie nous abordons à présent l'étude du Data Drift entre les datasets *application_train* et *application_test*. Cette transition est essentielle pour évaluer la stabilité et la fiabilité du modèle en conditions de production, une étape clé pour garantir la robustesse des prédictions.

L'analyse du Data Drift permet de détecter d'éventuelles variations dans la distribution des variables explicatives utilisées par le modèle. En effet, le dataset *application_train* représente les données historiques sur lesquelles le modèle a été entraîné, tandis que *application_test* simule un scénario avec de nouvelles données client, reflétant une situation réelle de production. Cette comparaison est cruciale pour identifier les variables dont les distributions ont significativement changé, ce qui pourrait potentiellement impacter les performances du modèle. Pour mener cette analyse, nous avons employé des techniques statistiques avancées, notamment ***la distance de Wasserstein*** pour les variables numériques et la ***distance de Jensen-Shannon*** pour les variables catégoriques. Le choix de ces méthodes nous permet de quantifier précisément l'écart entre les distributions des deux jeux de données. Le **seuil de détection de drift a été fixé à 0.5**, permettant ainsi de repérer les dérives qui requièrent une attention particulière.

B. Interprétation du résultat sur le drift

Bien que notre analyse ait révélé un drift dans certaines variables, **aucun n'a excédé le seuil critique**, ce qui confirme la stabilité générale du modèle. Cependant, **la présence de drift modéré dans environ 7.5% des variables** nous encourage à recommander une surveillance continue et une **réévaluation périodique** des variables concernées pour prévenir tout impact négatif futur sur la précision des prédictions du modèle. En somme, cette section de notre note méthodologique fait le pont entre l'évaluation des performances initiales et la nécessité d'un monitoring rigoureux du modèle en production, soulignant l'importance de cette étape pour le maintien de la validité et de l'efficacité du modèle dans le temps.

[PARTIE 7] — CONCLUSION ET LIMITES DE L'ETUDE

En conclusion, notre projet a réussi à mettre en œuvre un système de scoring efficace, marquant une avancée significative dans le développement de notre solution analytique. Au cours de cette étude, nous avons mené un travail approfondi sur l'ingénierie des caractéristiques pour optimiser les entrées de notre modèle. Nous avons également appliqué des techniques de modélisation avancées qui nous ont permis de construire un modèle prédictif robuste. L'intégration de

MLFLOW a joué un rôle crucial en nous permettant de documenter les diverses versions de notre modèle et les paramètres de formation, facilitant la reproduction des résultats.

Enfin, le développement d'un tableau de bord pour visualiser les résultats du scoring rend les prédictions de notre modèle accessibles et facilement interprétables, ce qui est essentiel pour une application pratique.

Limites et pistes d'amélioration

Toutefois, cette étude n'est pas sans limites. Premièrement, l'optimisation des hyperparamètres s'est révélée être un processus chronophage, particulièrement pour les modèles complexes tels que XGBoost et LightGBM. Une piste d'amélioration pourrait être l'utilisation de la librairie ***Optuna***, qui propose une méthode plus efficace d'optimisation des hyperparamètres par recherche bayésienne, potentiellement réduisant le temps nécessaire pour ces opérations.

Deuxièmement, le processus de création de nouvelles caractéristiques a introduit un drift significatif, observé lors de tests préliminaires. Ce drift pourrait être dû à des changements dans la distribution des données ou des comportements des utilisateurs non capturés par les modèles antérieurs. Ceci souligne la nécessité de mettre en place un **monitoring continu** des entrées pour détecter et ajuster rapidement à ces changements, assurant ainsi la fiabilité et la précision du modèle sur le long terme.

Enfin, la sécurité de l'application et la gestion des dépendances des librairies représentent des défis supplémentaires. Les dépendances externes, notamment les librairies utilisées pour le traitement des données et la modélisation, peuvent poser des problèmes de compatibilité ou de sécurité qui nécessitent une vigilance constante.

Assurer la maintenance et **la mise à jour régulières** de ces composants est crucial pour éviter des vulnérabilités et garantir la stabilité du système. Ces éléments mettent en lumière les défis à surmonter pour raffiner davantage notre solution de scoring et renforcer sa performance dans des applications futures. En adressant ces limites, nous pourrons améliorer la robustesse et l'efficacité de notre modèle, augmentant ainsi sa valeur et son applicabilité dans des contextes réels.