Note Méthodologique

Table des matières

[TL;DR : Résumé de la méthodologie de scoring de crédit 2](#_Toc168490165)

[Introduction 4](#_Toc168490166)

[I. Méthodologie 5](#_Toc168490167)

[I.1. Préparation des données 5](#_Toc168490168)

[I.2. Développement des modèles de scoring 6](#_Toc168490169)

[I.3. Evaluation des modèles 7](#_Toc168490170)

[I.3.1. Coût Métier Total (CMT) 7](#_Toc168490171)

[I.4. Sélection du modèle final 9](#_Toc168490172)

[I.4.1. CMT vs F1\_Score\_1 vs AUC ROC 9](#_Toc168490173)

[I.4.2. Precision\_1 vs Recall\_1 vs AUC ROC 9](#_Toc168490174)

[II.4.3. Synthèse et comparaison des meilleurs modèles 10](#_Toc168490175)

[II. Interprétabilité globale et locale du modèle (Analyse de Shap) 11](#_Toc168490176)

[II.1. Interprétabilité Globale 11](#_Toc168490177)

[II.2. Interprétabilité Locale 12](#_Toc168490178)

[III. Analyse du Data Drift avec Evidently AI 13](#_Toc168490179)

[IV. Gestion du code et des versions (MLflow, Dagshub, Git/Github) 15](#_Toc168490180)

[V. Déploiement de l'API (Streamlit Cloud) 17](#_Toc168490181)

[VI. Tests Unitaires 20](#_Toc168490182)

[Conclusion : Développement d'un Modèle de Scoring de Crédit Performant et Interprétable 21](#_Toc168490183)

# TL;DR : Résumé de la méthodologie de scoring de crédit

**Problème :** Développer un modèle de scoring de crédit performant pour évaluer la solvabilité des clients et minimiser les coûts d'erreurs de prédiction.

**Méthodologie :**

1. **Préparation des données :**
   * Collecte et nettoyage des données historiques de demandes de crédit.
   * Exploration et analyse des données pour identifier les distributions des variables, les relations entre elles et les potentiels biais.
   * Gestion des déséquilibres de clients : SMOTE (suréchantillonnage synthétique des exemples minoritaires) et ajustement des poids des classes.
2. **Développement des modèles de scoring :**
   * Sélection de modèles d'apprentissage automatique adaptés à la classification binaire (XGBOOST, Random Forest, Logistic Regression, etc.).
   * Apprentissage des modèles sur l'ensemble de données préparé.
   * Hyperparamètre tuning pour optimiser les performances du meilleur modèle.
3. **Evaluation des modèles :**
   * Métriques d'évaluation : Coût Métier Total (CMT), F1\_Score\_1 et AUC\_ROC.
   * Comparaison des modèles en se basant sur les métriques d'évaluation.
   * Analyse du CMT pour identifier le seuil de décision optimal.
4. **Sélection du modèle final :**
   * Choix du modèle avec les meilleures performances :
     + Minimisation du CMT
     + Maximisation du F1\_Score\_1 et de l'AUC\_ROC
     + Temps d'entrainement raisonnable
   * Le modèle XGBoost sans feature engineering, sans SMOTE et avec hyperparamètre tuning a été sélectionné.
5. **Interprétabilité du modèle :**
   * Analyse de SHAP globale pour identifier les features les plus importantes pour la prédiction de la solvabilité.
   * Analyse de SHAP locale pour expliquer les prédictions du modèle pour des clients individuels.
6. **Analyse du Data Drift :**
   * Utilisation de Evidently AI pour analyser le Data Drift entre les données d'apprentissage et les données de production.
   * Faible Data Drift détecté, suggérant que le modèle devrait continuer à fonctionner sur les nouveaux clients.
7. **Gestion du code et des versions :**
   * MLflow pour centraliser le suivi des expérimentations, des modèles et des artefacts.
   * Dagshub pour accéder aux modèles stockés dans MLflow depuis le dashboard.
   * Git/Github pour stocker le code source, les scripts d'entraînement et les notebooks d'analyse.
8. **Déploiement de l'API :**
   * Utilisation de Streamlit Cloud pour créer une interface web interactive pour le modèle de scoring.
   * L'interface permet de saisir les informations d'un client et d'obtenir une prédiction de solvabilité avec les features les plus impactantes.
9. **Tests unitaires :**
   * Création de tests unitaires pour la fonction de chargement de modèle et la fonction de chargement des données de tests.

**Conclusion :**

Le document présente une méthodologie rigoureuse et complète pour le développement d'un modèle de scoring de crédit performant. Le modèle développé permet d'évaluer la solvabilité des clients avec précision et de minimiser les coûts d'erreurs de prédiction. L'utilisation de techniques d'interprétabilité, d'analyse du Data Drift et de gestion du code garantit la fiabilité et la pérennité du modèle.

**Points clés :**

* Le Coût Métier Total (CMT) est une métrique métier importante pour l'évaluation des modèles de scoring de crédit.
* L'interprétabilité du modèle est cruciale pour comprendre son fonctionnement et éviter les biais.
* L'analyse du Data Drift permet de garantir que le modèle continue à fonctionner sur les nouvelles données.
* La gestion du code et des versions assure la traçabilité et la reproductibilité du projet.
* Le déploiement de l'API permet une utilisation facile du modèle par les parties prenantes.

# Introduction

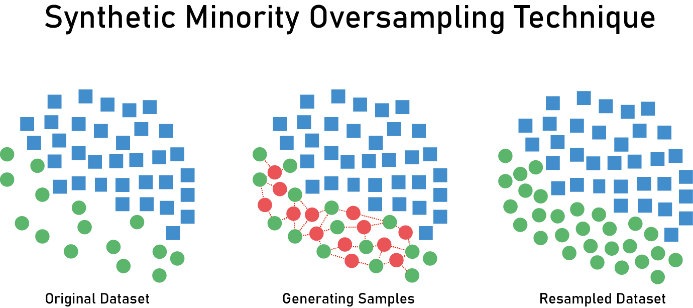
Ce document présente la méthodologie employée pour développer un modèle de scoring permettant d'évaluer la probabilité de remboursement d'un crédit et de classifier les demandes en& crédits accordés ou refusés. Le projet vise à adresser deux problématiques majeures :

1. **Gérer les déséquilibres de clients :** La population de données clients peut présenter un déséquilibre important entre les clients solvables (~92%) et les clients insolvables (~8%), ce qui peut affecter la performance des modèles d'apprentissage automatique.
2. **Minimiser les coûts associés aux erreurs de prédiction :** Les erreurs de prédiction, qu'elles soient des faux négatifs (octroi de crédit à un client insolvable) ou des faux positifs (refus de crédit à un client solvable), engendrent des coûts importants pour l'institution financière.

# I. Méthodologie

## I.1. Préparation des données

* **Collecte et nettoyage des données :** Rassembler les données historiques sur les demandes de crédit, incluant les caractéristiques des clients, les informations sur le crédit et les informations sur le remboursement. Nettoyer les données pour éliminer les valeurs manquantes, les incohérences et les erreurs. Pour accélérer le processus, l’utilisation de kernels Kaggle déjà existants a été utilisées :
  + <https://www.kaggle.com/code/willkoehrsen/start-here-a-gentle-introduction/notebook>
  + <https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script>).
* **Exploration et analyse des données :** Explorer les données pour identifier les distributions des variables, les relations entre les variables et les potentiels biais présents.
* **Gestion des déséquilibres de clients :** Utiliser deux techniques principales pour gérer les déséquilibres de clients :
* **SMOTE (Synthetic Minority Oversampling Technique) :** Générer synthétiquement de nouveaux exemples pour la classe minoritaire (clients insolvables) afin de créer un ensemble de données plus équilibré.



* **Ajustement du poids des classes :** Ajuster les poids attribués aux exemples de chaque classe lors de l'apprentissage du modèle pour tenir compte de leur importance relative.

## I.2. Développement des modèles de scoring

* **Sélection des modèles :** Choisir une variété de modèles d'apprentissage automatique adaptés à la classification binaire, tels que XGBOOST, Random Forest, Logistic Regression, Gradient Boosting Classifier, LightGBM, CatBoost, Voting Classifier, Stacking Classifier, Bagging Classifier.
* **Apprentissage des modèles :** Entraîner chaque modèle sur l'ensemble de données préparé. Cinq tests ont été utilisés :

## I.3. Evaluation des modèles

* **Définition des métriques d'évaluation :** Utiliser trois métriques principales pour évaluer la performance des modèles :
  + **Coût Métier Total (CMT) :** Une métrique définie pour mesurer le coût global des erreurs de prédiction, en tenant compte des coûts des faux positifs et des faux négatifs.
  + **F1\_Score\_1 :** Mesure de la performance globale du modèle, prenant en compte à la fois la précision et le rappel pour la classe des clients insolvables.
  + **AUC\_ROC (Area Under the ROC Curve) :** Mesure de la capacité du modèle à distinguer entre les clients solvables et les clients insolvables sur l'ensemble des seuils de probabilité possibles.
* **Comparaison des modèles :** Comparer les performances des différents modèles en se basant sur les métriques d'évaluation définies.

### I.3.1. Coût Métier Total (CMT)

Le **Coût Métier Total (CMT)** est une métrique métier conçue pour évaluer l'impact financier des erreurs de prédiction d'un modèle de scoring de crédit. Il prend en compte les coûts engendrés par les faux négatifs (octroi de crédit à un client insolvable) et les faux positifs (refus de crédit à un client solvable).

La formule du CMT est la suivante :

Avec :

- FP : Nb Faux Positifs  
 - FN : Nb Faux Négatifs  
 - VP : Nb Vrai Positifs  
 - VN : Nb Vrai Négatifs  
 - N**b de crédits :** Nombre total de demandes de crédit dans l'ensemble de données

**Pondération des erreurs :**

La formule du CMT attribue un poids plus important aux faux négatifs (FN) qu'aux faux positifs (FP). Cela reflète le fait que les conséquences d'un refus de crédit à un client solvable (coût de renonciation à un client potentiel) peuvent être moins importantes que les conséquences d'un octroi de crédit à un client insolvable (perte financière due au défaut de paiement).

**Seuil optimal**

Le seuil de décision est une valeur de probabilité de crédit utilisée pour classifier une demande de crédit comme "accordée" ou "refusée".

Le seuil optimal peut être déterminé en utilisant la courbe de ROC (Receiver Operating Characteristic Curve). La courbe de ROC montre la relation entre la sensibilité (taux de vrais positifs) et la spécificité (taux de vrais négatifs) du modèle pour différents seuils de décision.

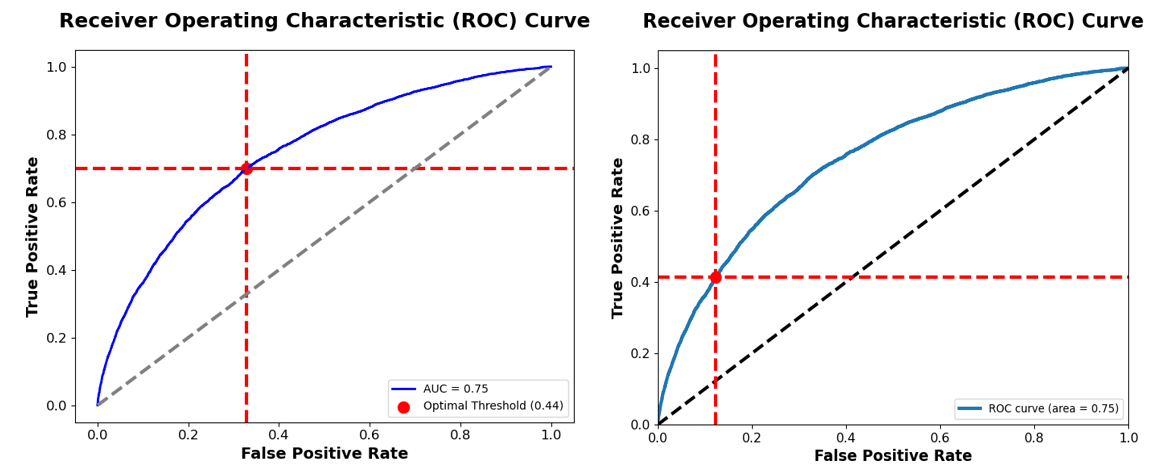


Figure 1: Courbes de ROC indiquant le choix de 2 seuils différents (fonction d'optimisation) :  
A gauche : Maximisation du TPR et minimisation du FPR.   
A droite  : Minimisation du F1\_Score\_1.

**Conclusion sur le CMT**

Un CMT plus faible indique que le modèle génère moins d'erreurs coûteuses et qu'il est donc plus performant d'un point de vue métier.

Le CMT présente plusieurs avantages en tant que métrique métier pour l'évaluation des modèles de scoring de crédit :

* **Prise en compte des coûts réels des erreurs :** Le CMT intègre les coûts financiers des faux positifs et des faux négatifs, ce qui permet d'évaluer l'impact réel du modèle sur l'institution financière.
* **Facilité d'interprétation :** Le CMT est une métrique simple et intuitive qui peut être facilement comprise par les parties prenantes non techniques.
* **Adaptabilité à différents scénarios :** La pondération des erreurs dans la formule du CMT peut être ajustée pour refléter les priorités spécifiques de l'institution financière.

**Le Coût Métier Total (CMT) est une métrique métier précieuse pour l'évaluation des modèles de scoring de crédit. Il permet de quantifier l'impact financier des erreurs de prédiction et de sélectionner le modèle qui optimise les performances d'un point de vue métier.**

## I.4. Sélection du modèle final

* **Sélectionner le modèle avec les meilleures performances :** Choisir le modèle qui 1. Minimise le CMT, 2. maximise le F1\_Score\_1 et l'AUC\_ROC et 3. Avec un temps d’entrainement faible.

### I.4.1. CMT vs F1\_Score\_1 vs AUC ROC

Une image contenant capture d’écran, Logiciel multimédia, texte, logiciel

Description générée automatiquement

Figure 2: Ensemble des modèles dans l'espace de représentation des métriques d'évaluations : CMT vs F1\_Score\_1 vs AUC ROC  
Note : Les modèles a favoriser sont ceux dans le coin supérieur gauche.

### I.4.2. Precision\_1 vs Recall\_1 vs AUC ROC

Une image contenant capture d’écran, Logiciel multimédia, logiciel, Logiciel de graphisme

Description générée automatiquement

Figure 3: Ensemble des modèles dans l'espace de représentation des métriques d'évaluations : Precision\_1 vs Recall\_1 vs AUC ROC  
Note : Les modèles à favoriser sont ceux dans le coin supérieur droit.

### II.4.3. Synthèse et comparaison des meilleurs modèles

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Conclusion sur le choix du modèle final**

Peu importe l’espace de représentation des métriques d’évaluations, les trois meilleurs modèles (1. Voting Classifier, 2. XGBOOST TUNED et 3. CatBoost) se situent dans les zones à favoriser.

Le modèle **Voting Classifier** (qui est basé sur les prédictions de XGBOOST, RANDOM FOREST, Logistic Regression, Gradient Boosting, CatBoost et LightGBM offre les meilleures performances en général pour l’AUC ROC, le CMT et le F1\_Score\_1. Cependant, ce modèle est relativement long à entrainer par rapport à d’autres modèles dont les performances sont similaires.

Pour cette raison, je me permets de prendre un modèle légèrement moins bon mais qui s’entraine plus vite : **le modèle** **XGBoost, sans feature engineering, sans SMOTE, et avec hyperparameter tuning.**

# II. Interprétabilité globale et locale du modèle (Analyse de Shap)

L'interprétabilité des modèles d'apprentissage automatique est cruciale pour comprendre leur fonctionnement et pour s'assurer qu'ils ne prennent pas de décisions discriminatoires ou biaisées. Dans le cadre de ce projet, deux approches d'interprétabilité ont été utilisées.

## II.1. Interprétabilité Globale

L'interprétabilité globale permet d'identifier les features qui ont le plus d'impact sur les prédictions du modèle.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Figure 4: Analyse de SHAP Globale

Dans ce projet, l'analyse de l'importance des features a révélé que les features les plus importantes pour la prédiction de la solvabilité des clients sont :

* **EXT\_SOURCE\_1**
* **EXT\_SOURCE\_2**
* **EXT\_SOURCE\_3**
* **AMT\_GOODS\_PRICE**
* **AMT\_CREDIT**

Ces features sont toutes liées à la capacité financière du client et à sa capacité à rembourser le crédit.

## II.2. Interprétabilité Locale

L'interprétabilité locale permet d'expliquer les prédictions du modèle pour un individu particulier. Dans ce projet, l'analyse de SHAP (SHapley Additive exPlanations) a été utilisée pour identifier les features qui ont le plus contribué à la prédiction de la solvabilité d'un client individuel.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 5: Analyse de SHAP Locale pour le client ayant l’index 5 dans le dataset

L'analyse de SHAP a montré que pour l'individu sélectionné, les features négatives qui ont le plus influencé la prédiction sont :

* **AMT\_CREDIT :** Le montant élevé du crédit demandé peut avoir conduit le modèle à prédire un risque de défaut de paiement.
* **AMT\_ANNUITY :** Le montant élevé de l'annuité du crédit peut également avoir contribué à la prédiction d'un risque de défaut de paiement.

En revanche, les principales features positives qui ont influencé la prédiction sont :

* **EXT\_SOURCE\_3 :** Une information positive provenant d'une source externe 3 peut avoir atténué l'impact des features négatives.
* **AMT\_GOODS\_PRICE :** Le prix élevé des biens pour lesquels le crédit est demandé peut indiquer une capacité financière suffisante pour rembourser le crédit.
* **DAYS\_EMPLOYED :** Un nombre élevé d'années d'ancienneté dans l'emploi peut également être un indicateur de stabilité financière et de capacité de remboursement.

# III. Analyse du Data Drift avec Evidently AI

L'analyse du Data Drift est cruciale pour garantir que les performances d'un modèle d'apprentissage automatique restent stables dans le temps. En effet, la distribution des données peut évoluer au fil du temps, ce qui peut affecter la performance du modèle s'il n'est pas mis à jour en conséquence.

Dans ce projet, la librairie Evidently AI a été utilisée pour analyser le Data Drift entre les données d'apprentissage (dataset "application\_train") et les données de production (dataset "application\_test").

Les hypothèses de départ sont :

* ***dataset “application\_train”*** représente les datas pour l’entrainement du modèle
* ***dataset “application\_test”*** représente les datas de nouveaux clients une fois le modèle en production.

Les résultats de l’analyse sont :

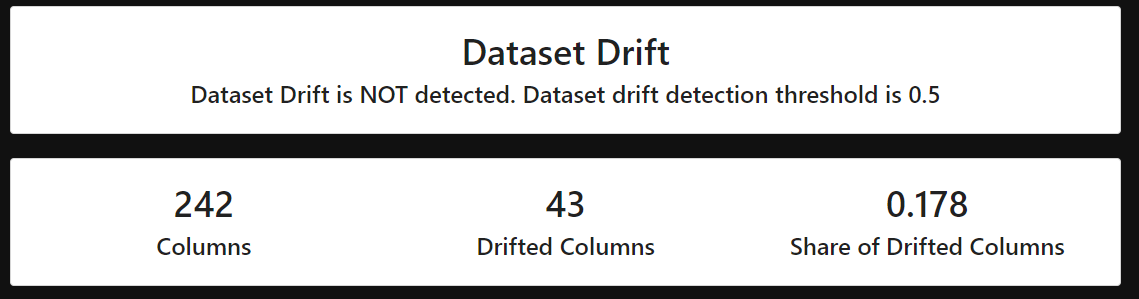


Figure 6: Résultats généraux de l'analyse du Data Drift

**Interprétation des résultats :**

Le fait qu'il n'y ait qu'un faible pourcentage de colonnes avec du drift suggère que les données d'apprentissage et les données de production sont relativement similaires. Cela signifie que le modèle devrait continuer à fonctionner de manière satisfaisante sur les nouveaux clients.

Il est important de noter que l'analyse du Data Drift ne garantit pas l'absence de problèmes de performance du modèle. D'autres facteurs, tels que des changements dans les caractéristiques des clients ou dans l'environnement économique, peuvent également affecter la performance du modèle. Il est donc important de surveiller les performances du modèle de manière continue et de le mettre à jour si nécessaire.

Ci-dessous une comparaison des distributions de train et test pour la feature AMT\_CREDIT. On remarque que la distribution est très similaire.

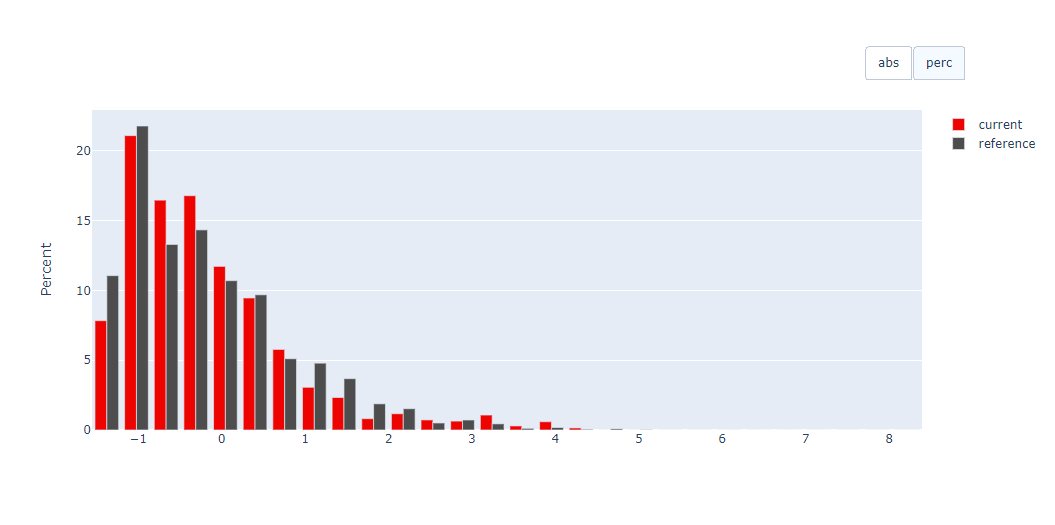


Figure 7: Comparaison des distributions de current (test) et reference (train) pour   
la feature AMT\_CREDIT (en %)  
Pas de data Drift détecté (score de 0.098).

# IV. Gestion du code et des versions (MLflow, Dagshub, Git/Github)

* **MLflow :** Implémenter MLflow pour centraliser le suivi des expérimentations, des modèles et des artefacts générés pendant le développement du modèle. Cela permet de reproduire facilement les résultats, de comparer les différentes versions du modèle et de suivre l'évolution des performances au fil du temps.

Une image contenant capture d’écran, texte, Appareils électroniques, logiciel

Description générée automatiquement

Figure 8: MLflow - Tableau des expérimentations.   
Pour chaque modèle, on y retrouve le dataset et les paramètres utilisés pour entrainer le modèle. On y retrouve aussi ses performances avec les métriques d'évaluation.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Figure 9: MLflow :Artifacts et structure d'un modèle stocké.  
Pour re-utiliser le modèle stocké dans MLflow, nous avons besoin de l’ensemble des fichiers dans la partie « model ».

* **Dagshub :** Utiliser Dagshub pour accéder aux modèles stockées dans MLflow dans mon code dashboard.
* **Git/Github :** Stocker le code source du projet, les scripts d'entraînement et les notebooks d'analyse dans un dépôt Git hébergé sur Github. Cela permet de versionner le code, de faciliter la collaboration et de garantir la traçabilité des modifications.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Figure 10: Screenshot de quelques commits pour le code d'entrainement des modèles

# V. Déploiement de l'API (Streamlit Cloud)

**Streamlit Cloud :** Déployer le modèle de scoring sur Streamlit Cloud pour créer une interface web interactive.

**Cas 1 : Le client existe, le crédit est approuvé (20% < 63%)**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Cas 2 : Le client n’existe pas dans le dataset Test**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Cas 3 : Le client existe, le crédit est refusé (64.30% > 63.64%)**

Le TOP10 des features ayant un impact positif ou négatif sont affichées.   
J’offre la possibilité de modifier les valeurs de ces features et re-prédire pour identifier des features clefs pour l’obtention du crédit.

Dans l’exemple ci-dessous, la feature ayant l’impact négatif le plus grand est : « DAYS EMPLOYED ». La modification de celle-ci permet de passer d’un refus à une acceptation.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, menu

Description générée automatiquement

# VI. Tests Unitaires

Création de deux tests unitaires pour 1. la fonction de chargement de modèle et 2. la fonction de chargement des données de tests.

**Workflow**

Une image contenant texte, logiciel, Logiciel multimédia, Logiciel de graphisme

Description générée automatiquementUne image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Dans l’exemple ci-dessus, les deux tests ont été un succès. Ils ont été réalisés en 1.39s.

# Conclusion : Développement d'un Modèle de Scoring de Crédit Performant et Interprétable

Ce document présente une méthodologie rigoureuse et complète pour le développement d'un modèle de scoring de crédit performant et interprétable. Le modèle développé permet d'évaluer la solvabilité des clients avec précision et de minimiser les coûts d'erreurs de prédiction.

**Points clés du projet :**

* **Gestion des déséquilibres de clients :** Le SMOTE et l'ajustement des poids des classes ont été utilisés pour gérer efficacement le déséquilibre des données entre les clients solvables et insolvables.
* **Sélection de modèles d'apprentissage automatique :** Une variété de modèles d'apprentissage automatique a été évaluée, et le modèle XGBoost a été sélectionné en raison de ses performances optimales en termes de coût métier total (CMT), F1\_Score\_1 et AUC\_ROC.
* **Interprétabilité du modèle :** L'analyse de SHAP a été utilisée pour identifier les features les plus importantes pour la prédiction de la solvabilité des clients, et pour expliquer les prédictions du modèle pour des clients individuels.
* **Analyse du Data Drift :** Evidently AI a été utilisé pour analyser le Data Drift entre les données d'apprentissage et les données de production, et a montré un faible niveau de drift, suggérant que le modèle devrait continuer à fonctionner efficacement sur les nouveaux clients.
* **Gestion du code et des versions :** MLflow, Dagshub et Git/Github ont été utilisés pour centraliser le suivi des expérimentations, des modèles et des artefacts, pour faciliter la collaboration et garantir la traçabilité des modifications.
* **Déploiement de l'API :** Streamlit Cloud a été utilisé pour déployer le modèle de scoring sur une interface web interactive, permettant aux utilisateurs de saisir les informations d'un client et d'obtenir une prédiction de solvabilité avec les features les plus impactantes.
* **Tests unitaires :** Des tests unitaires ont été créés pour la fonction de chargement de modèle et la fonction de chargement des données de tests, garantissant la robustesse du code.

**Impact et avantages du modèle :**

* **Réduction des coûts d'erreurs de prédiction :** Le modèle permet de minimiser les coûts d'erreurs de prédiction en identifiant plus précisément les clients solvables et insolvables.
* **Amélioration de la prise de décision :** Le modèle fournit aux analystes du crédit des informations précieuses pour prendre des décisions éclairées concernant l'octroi de crédit.
* **Augmentation de la satisfaction client :** En réduisant le nombre de refus de crédit erronés, le modèle peut contribuer à améliorer la satisfaction client.
* **Transparence et interprétabilité :** L'utilisation de l'analyse de SHAP permet de comprendre les raisons des prédictions du modèle, favorisant la transparence et l'acceptabilité du modèle.

**Conclusion générale :**

Le projet a abouti au développement d'un modèle de scoring de crédit performant, interprétable et robuste, qui peut apporter une valeur significative à l'institution financière en réduisant les coûts d'erreurs de prédiction, en améliorant la prise de décision et en augmentant la satisfaction client. La méthodologie rigoureuse employée et les outils utilisés garantissent la fiabilité, la maintenabilité et l'évolutivité du modèle.

**Recommandations pour une utilisation future :**

* **Surveillance continue du modèle :** Il est important de surveiller en permanence les performances du modèle sur les nouvelles données et de le mettre à jour si nécessaire en cas de Data Drift significatif ou de changements dans les caractéristiques des clients ou dans l'environnement économique.
* **Exploration d'autres techniques d'apprentissage automatique :** D'autres techniques d'apprentissage automatique, telles que les réseaux de neurones profonds, pourraient être explorées dans de futures recherches pour améliorer encore les performances du modèle.
* **Intégration du modèle dans les processus métier :** Le modèle de scoring peut être intégré aux processus métier existants pour automatiser l'évaluation de la solvabilité des clients et améliorer l'efficacité du processus de décision de crédit.

Ce projet démontre le potentiel des techniques d'apprentissage automatique pour développer des modèles de scoring de crédit précis, interprétables et robustes, qui peuvent apporter des avantages significatifs aux institutions financières.