NOTE METHODOLOGIQUE

Introduction

Le présent document détaille la méthodologie employée pour former le modèle de scoring de crédit dans le cadre du projet en cours (https://github.com/Emeline2104/Projet-7-Models-API). La mission consiste à élaborer un algorithme de classification capable de prédire la probabilité de défaut de paiement des clients de "Prêt à Dépenser". L'objectif de ce document est de fournir un aperçu clair de l'approche adoptée, en mettant l'accent sur des aspects tels que:

- l'entraînement du modèle ;
- la gestion du déséquilibre des classes ;
- l'optimisation du seuil de probabilité et la fonction coût métier ;
- la synthèse des résultats ;
- l'interprétabilité globale et locale du modèle ;
- les limites et améliorations possibles ;
- l'analyse du data drift.

Entrainement du modèle

L'entraînement du modèle comprend plusieurs étapes majeures dont le chargement des données, l'analyse exploratoire, la préparation des données et l'entraînement du modèle. Ces étapes sont détaillées ci-dessous.

Chargement des données

Les données ont été extraites de Kaggle (https://www.kaggle.com/c/home-credit-default-risk/data) et sauvegardées sur un serveur S3 d'AWS, réparties en dix tables distinctes (voir schémas ci-dessous).

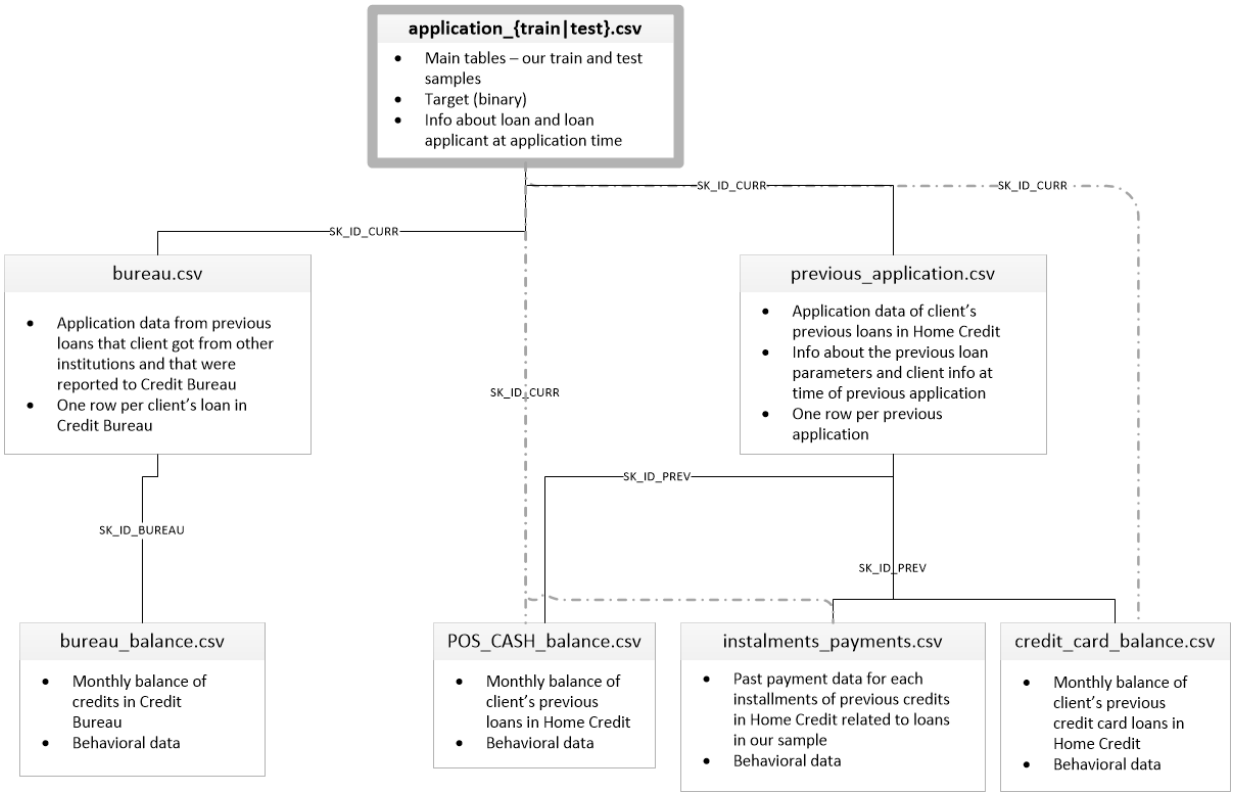


Figure 1 : Architecture des bases de données

Analyse exploratoire

L'analyse exploratoire englobe diverses informations, notamment les détails personnels des clients tels que l'âge, l'emploi, le parcours professionnel et scolaire, la situation familiale, ainsi que les biens et possessions. Elle comprend également des informations détaillées sur la demande de prêt, les demandes précédentes au sein de Prêt à Dépenser, ainsi que les informations remontées par d'autres institutions financières via le bureau de crédit, avec un détail mensuel.

Le jeu de données test compte 48 744 demandes, tandis que le jeu de données d'entraînement comprend 307 511 demandes.

Lors de l'analyse des variables, il est apparu que certaines d'entre elles semblent être particulièrement informatives pour évaluer le risque de défaut. Ces variables incluent des aspects divers tels que le type d'accompagnement des clients, le genre, le statut familial et le nombre d'enfants, le type de revenu, la nature du travail, le niveau d'éducation, le salaire, l'âge, le temps de

travail, le prix du bien, le statut du crédit des clients, le type de crédit, la durée du crédit, l'objectif du prêt, ainsi que le statut des contrats précédents.
L'impact de ces variables sur le risque de défaut peut être exploré en détail, offrant ainsi une vision approfondie des facteurs qui influent sur la fiabilité du remboursement. L'examen de ces caractéristiques variées offre une opportunité de mieux comprendre les dynamiques sous-jacentes et d'optimiser la modélisation du scoring de crédit pour une prise de décision plus précise.

Préparation des Données

La première étape de la méthodologie consiste en une préparation des données. L'analyse de la qualité de données mets en lumière la présence de valeurs manquantes dans le dataset, l'absence de doublons, et l'identification d'un déséquilibre dans la variable cible, où seulement 8% des prêts ont la valeur 1, indiquant un défaut de remboursement.
La gestion de la qualité des données est réalisée en fonction du modèle sélectionné. Par exemple, pour les modèles ne prenant pas en charge les données manquantes (comme la régression logistique) ou pour les techniques spécifiques au déséquilibre de classes (comme *SMOTE*), des approches adaptées de sélection des caractéristiques avec un taux de remplissage minimum puis une surpression des lignes avec des valeurs manquantes est appliquées.
L'agrégation des données se fait à l'aide des clés de jointures entre les tables (voir Figure 1). Pendant cette phase, un encodage one-hot est appliqué aux variables catégorielles, et dans le cas de la présence de seulement deux catégories, un encodage catégoriel est effectué.

Entraînement du Modèle

Le modèle a été entrainé en suivant des étapes adaptées à chaque type de modèle (régression logistique, forêt aléatoire, LightGBM) :

- Tout d'abord, le jeu de données a été séparé en jeu de test et en jeu d'entraînement. Ensuite, un pipeline de modèle a été créé, comprenant la standardisation (uniquement dans le cas de la régression logistique) et le classificateur.
- En raison de la taille significative du jeu de données, la recherche des hyperparamètres a été initialement réalisée sur un sous-ensemble de données à l'aide de *GridSearchCV* et d'un *StratifiedKFold*, en tenant compte de la gestion du déséquilibre des classes (voir la section sur le traitement du déséquilibre des classes).
- La sélection des caractéristiques a été effectuée pour réduire la complexité du modèle tout en préservant ses performances, en excluant les variables dont l'importance dépasse un seuil prédéfini.
- Le modèle a ensuite été entrainé sur l'ensemble du jeu de données avec des hyperparamètres optimisés et des variables sélectionnées.
- Enfin, l'évaluation du modèle a été réalisée (voir la section sur l'optimisation du seuil de probabilité et la fonction de coût métier).

Traitement du déséquilibre des classes

L'importance de la gestion du déséquilibre des classes dans la variable cible est cruciale, particulièrement dans ce contexte où la répartition est de 8% pour une catégorie et 92% pour l'autre. Cette disparité peut biaiser le modèle en faveur de la classe majoritaire, affectant ainsi sa capacité à détecter les exemples de la classe minoritaire.

Différentes approches ont été évaluées pour remédier à ce déséquilibre, notamment l'absence de traitement, l'utilisation de la pondération de classe (*class_weight*), et l'application de la technique *SMOTE*. Chacune de ces approches vise à améliorer la capacité du modèle à bien généraliser à la classe minoritaire.

- **Absence de traitement :** C'est la méthode de base où aucune correction n'est apportée au déséquilibre. Cependant, cette approche peut entraîner une mauvaise performance du modèle sur la classe minoritaire.
- **Pondération de classe (class_weight) :** Cette approche ajuste les poids des classes lors de l'entraînement du modèle, donnant plus de poids à la classe minoritaire. Cette méthode s'est avérée plus efficace, surtout avant l'optimisation du seuil de classification avec la méthode predict_proba.
- **SMOTE :** La technique SMOTE vise à synthétiser des exemples supplémentaires de la classe minoritaire pour équilibrer la distribution. Cependant, elle nécessite la suppression des valeurs manquantes, ce qui peut poser des problèmes, notamment la perte d'informations importantes.

Une fois l'optimisation du seuil de probabilité effectuée en amont, la méthode la plus efficace s'est révélée être l'absence de rééchantillonnage (None). Cette approche a été retenue en raison de sa capacité à mieux gérer le déséquilibre des classes tout en maintenant la qualité des données et en évitant la suppression d'informations cruciales associées aux valeurs manquantes.

Fonction coût métier, algorithme d'optimisation et métrique d'évaluation

Fonction coût métier

Pour évaluer le modèle de classification, les métriques habituelles, telles que l'AUC et l'accuracy, ont été utilisées. Cependant, dans le cas d'un déséquilibre de classe, la précision seule peut être trompeuse en raison du biais vers la classe majoritaire. En particulier, lorsque les coûts d'erreurs de faux négatifs (FN) et de faux positifs (FP) diffèrent, l'accuracy seule peut ne pas refléter correctement l'impact métier.

Pour surmonter cette limitation, un score personnalisé a été mis en place, prenant en compte le déséquilibre du coût métier entre FN et FP. La formule de ce score personnalisé, normalisé par le nombre total d'individus, est définie comme suit :

$$\text{score\_personalized} = \frac{10 \times \text{fn} + \text{fp}}{\text{num\_individuals}}$$

Où :

- fn représente le nombre de faux négatifs (mauvais client prédit comme bon client).
- fp représente le nombre de faux positifs (bon client prédit comme mauvais client).
- num_individuals est le nombre total d'individus dans l'échantillon.

Ce score personnalisé tient compte de l'écart de coût métier entre FN et FP. Un score plus bas indique une meilleure performance du modèle, car cela signifie une minimisation du coût d'erreur prédiction.

Algorithme d'optimisation du seuil

Parallèlement à l'optimisation des hyperparamètres, le seuil de classification a été ajusté pour déterminer la classe 0 ou 1 à partir d'une probabilité donnée par la fonction *predict_proba*. Ce seuil est optimisé après la recherche des hyperparamètres en utilisant la fonction *predict_proba*, en recherchant le seuil pour lequel le score personnalisé est minimal. La valeur optimale du seuil varie en fonction du type de gestion du déséquilibre des classes, environ 0,45 avec gestion et environ 0,1 sans gestion. Cette approche garantit que le modèle est ajusté de manière optimale pour les besoins spécifiques de l'application.

Métriques d'évaluation

Les critères d'évaluation du modèle comprennent l'AUC, l'accuracy, et le score personnalisé, comme mentionné précédemment. En plus de ces métriques, les temps d'entraînement et d'inférence sont également pris en considération. Ces derniers aspects jouent un rôle crucial dans l'évaluation globale du modèle, contribuant à une compréhension complète de ses performances en termes de précision, de coût métier, et d'efficacité temporelle. Toutes ces métriques sont rigoureusement suivies grâce aux outils de MLOps, en particulier MLflow, assurant ainsi une gestion transparente et efficace du cycle de vie du modèle, de son développement initial à son déploiement opérationnel.

Synthèse des résultats

Modèle	Gestion déséquilibre classes	AUC Test	Accuracy Test	Score personnalisé Test	Temps entrainement (s)	Temps inférence (s)
Baseline (DummyClassifier)	NAN	0,49	0,85	0,83	1	0,003
Regression Logistique	None	0,51	0,48	0,85	17	0,2
	Class_weight	0,52	0,24	0,87	25	0,2
	SMOTE	0,51	0,44	0,85	45	0,8
Forêt Aléatoire	None	0,63	0,67	0,63	1975	1,0
	Class_weight	0,66	0,68	0,59	2391	0,8
	SMOTE	0,64	0,65	0,61	4151	3,8
Light GBM	None	0,71	0,76	0,50	3757	5,0
	Class_weight	0,71	0,74	0,50	2909	8,9
	SMOTE	0,68	0,72	0,54	1272	1,7

Le Light GBM émerge comme le meilleur modèle, et en ce qui concerne le déséquilibre de classe, l'application initiale de SMOTE et class_weight améliore les performances. Initialement, le choix est de retenir class_weight, car SMOTE nécessite la gestion des valeurs manquantes.

Dans une étape ultérieure, l'optimisation du seuil de classification est réalisée, conduisant à des performances comparables sans la gestion explicite du déséquilibre de classe, tout en réduisant le temps d'inférence. En conséquence, le modèle final retenu est le **Light GBM**, **sans gestion explicite du déséquilibre de classe,** mais avec un seuil de classification optimisé fixé à 0,09.

Interprétabilité globale et locale du modèle

Analyse globale

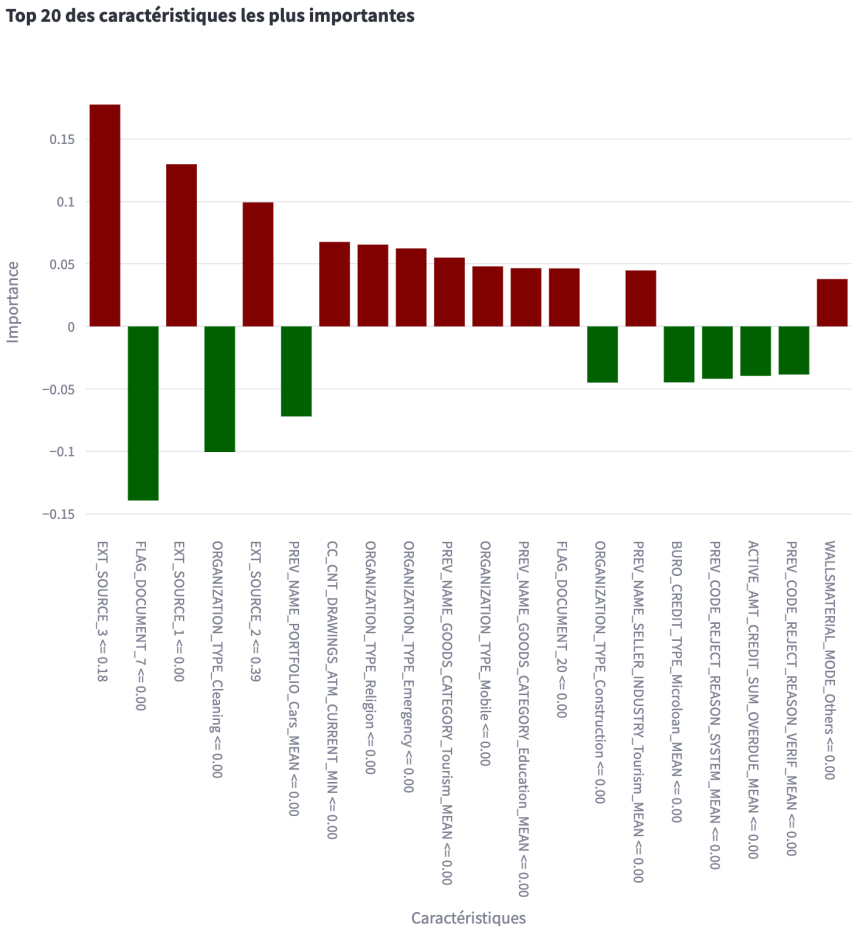
Comprendre le fonctionnement global du modèle est essentiel. Extraire les caractéristiques les plus influentes grâce à des techniques telles que l'extraction des importances des caractéristiques offre une vision claire des facteurs déterminants dans le processus de prise de décision du modèle.

Les points clés relatif à la décision d'octroi de crédit sont les suivants :

- **Sources Externes de Notation** (EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3) : Scores cruciaux pour évaluer la fiabilité financière des clients.
- **Taux de Paiement** (PAYMENT_RATE) : Indicateur déterminant de la capacité de remboursement, reflétant la manière dont les paiements sont effectués.
- **Âge du Client** (DAYS_BIRTH) : L'âge joue un rôle crucial, les clients plus jeunes pouvant être considérés comme plus risqués.
- **Mensualité du Prêt** (AMT_ANNUITY) : Importance des mensualités par rapport au revenu, indiquant le niveau de risque.

Analyse locale

Pour une compréhension précise des prédictions pour une observation spécifique, LIME est utilisé. Ce principe implique la création de modèles locaux simples autour de points de données spécifiques, offrant ainsi une interprétation claire de la décision du modèle pour une instance particulière. Un exemple de cette analyse est pour le client *10002* (dont le prêt a été refusé).



Ce graphique explique la prédiction de défaut de crédit concernant le client spécifié.

Plus la barre est haute, plus la caractéristique est importante, et la direction de la barre indique si cette caractéristique a eu un impact positif ou négatif sur la probabilité de refus de crédit. Par exemple pour ce client la valeur de *EXT_SOURCE_3, EXT_SOURCE_1, EXT_SOURCE_2* semble être la raison principale pour lequel le client n'a pas obtenu son prêt.

Les limites et les améliorations possibles

Les différentes limites et améliorations associées sont les suivantes :

- Dans le cadre de l'**optimisation des hyperparamètres**, la méthodologie actuelle repose sur l'utilisation de GridSearchCV. La méthodologie actuelle utilise GridSearchCV, mais cette approche peut être intensive en termes de ressources computationnelles et de temps. Cependant, d'autres méthodes plus avancées telles que l'optimisation bayésienne pourraient être utilisées. Cette approche vise à garantir un ajustement optimal du modèle en explorant de manière intelligente l'espace des hyperparamètres, crucial dans le contexte de prédiction du risque de défaut de client pour des prêts.
- Actuellement, l'approche pour traiter le **déséquilibre** repose sur l'utilisation de class_weight ou la technique de suréchantillonnage SMOTE, qui ne sont pas toujours optimales car dépendant de la distribution des données. Cependant, il serait opportun d'examiner d'autres approches plus adaptées à la spécificité de notre ensemble de données. Une option à considérer serait l'utilisation de la méthode ADASYN (Adaptive Synthetic Sampling), une variante du suréchantillonnage, qui génère des exemples synthétiques en accord avec la densité locale des zones minoritaires. Cette approche peut offrir une meilleure adaptation aux caractéristiques particulières du jeu de données, contribuant ainsi à une gestion plus efficace du déséquilibre.
- Pour une meilleure **sélection de variables** dans le contexte de prédiction de risque avec un grand nombre de données, des approches telles que l'Analyse en Composantes Principales (ACP), l'utilisation d'algorithmes de machine learning (RFE), des tests statistiques avancés (Kolmogorov-Smirnov), des techniques basées sur la stabilité, et l'importance des variables dans des modèles spécifiques comme le Gradient Boosting pourraient être envisagés. Ces méthodes offrent une sélection plus précise et adaptée aux nuances de l'ensemble de données.
- Une piste d'amélioration importante serait d'explorer la création d'un **ensemble de modèles**. Combiner les prédictions de plusieurs approches peut renforcer la robustesse du modèle global, un aspect crucial dans le projet de prédiction du risque de défaut de client pour des prêts.
- La **gestion des données manquantes** pourrait être approfondie en étudiant des méthodes plus avancées, telles que l'imputation basée sur des modèles. Cette démarche peut contribuer à une gestion plus nuancée des données manquantes, minimisant ainsi le risque de biais dans le modèle résultant de la suppression de lignes.
- Afin de renforcer la **compréhension du modèle** de prédiction du risque de défaut de client pour des prêts, l'intégration de techniques d'interprétabilité supplémentaires pourraient être implémentée. Par exemple, l'utilisation de SHAP (SHapley Additive exPlanations) peut offrir des insights complémentaires en évaluant l'impact de chaque variable sur les prédictions. Cette approche enrichit l'interprétation globale et locale du modèle, fournissant ainsi une vision plus détaillée de ses mécanismes sous-jacents.

L'analyse du Data Drift

Le data drift se réfère à l'évolution des données au fil du temps, pouvant impacter la performance des modèles. L'analyse de data drift vise à détecter les changements dans la distribution des données entre le dataset d'entraînement et celui de test. Cela s'effectue en mesurant les distances entre les distributions, utilisant des métriques telles que la distance de Wasserstein et la distance de Jensen-Shannon.

Principaux Résultats - Dataset Drift

Le dataset drift n'est pas détecté, avec un seuil de détection fixé à 0.5. Cependant, une analyse des colonnes révèle que 9.091% des colonnes (11 sur 121) montrent des signes de drift. Des caractéristiques clés, telles que AMT_REQ_CREDIT_BUREAU_MON, présentent une Wasserstein distance significative de 2.32, indiquant un changement notable. Voici une analyse détaillée de certaines caractéristiques spécifiques :

Caractéristiques avec une Forte Distorsion	
AMT_REQ_CREDIT_BUREAU_MON (Numerique)	Wasserstein distance (normed) : 2.33
AMT_REQ_CREDIT_BUREAU_WEEK (Numerique)	Wasserstein distance (normed) : 0.58
AMT_REQ_CREDIT_BUREAU_QRT (Numerique)	Wasserstein distance (normed) : 0.41
NAME_CONTRACT_TYPE (Catégorique)	Jensen-Shannon distance : 0.14
Caractéristiques avec une Distorsion Modérée	
AMT_GOODS_PRICE, AMT_CREDIT (Numeriques)	Wasserstein distances (normed) : 0.23
AMT_ANNUITY (Numerique)	Wasserstein distance (normed) : 0.15
DAYS_LAST_PHONE_CHANGE (Numerique)	Wasserstein distance (normed) : 0.13
FLAG_EMAIL (Numerique)	Jensen-Shannon distance : 0.12
AMT_REQ_CREDIT_BUREAU_DAY (Numerique)	Wasserstein distance (normed) : 0.11

Actions à Mettre en Place :

Les actions à mettre en place sont:

- Surveiller de près les caractéristiques avec une distorsion significative.
- Adapter les modèles pour tenir compte des changements détectés.
- Réévaluer la qualité des prédictions et, si nécessaire, re-entrainer le modèle.
- Mettre en œuvre un processus de suivi continu du data drift pour assurer la robustesse du modèle sur le long terme.