Projet 7 - Implémentez un modèle de scoring

PRET A DEPENSER

Note Méthodologique

Soutenance P7 – Data Scientist – OpenClassrooms 2024

Table des matières

1. Méthodologie d'entraînement du modèle

▪ Prétraitement des données

L'imputation des valeurs manquantes s'effectue avec la médiane (**SimpleImputer**), suivie d'une normalisation à l'aide du **MinMaxScaler**. Cela garantit que toutes les fonctionnalités sont à la même échelle, facilitant l'apprentissage du modèle.

▪ Sélection des caractéristiques

L'algorithme de sélection récursive des caractéristiques (**RFE**) est utilisé pour réduire la dimensionnalité des données tout en conservant les caractéristiques les plus pertinentes. Il élimine de manière itérative les caractéristiques les moins significatives, en s'assurant de ne retenir que 30% des caractéristiques initiales, celles qui apportent le plus de valeur aux prédictions du modèle.

▪ Entraînement

Les données sont divisées en ensembles d'entraînement et de test, puis utilisées pour entraîner plusieurs modèles de classification, notamment la **LogisticRegression** et **LGBMClassifier**. Pour améliorer la robustesse, l'entraînement est optimisé à l'aide de la classe **StratifiedKFold** pour la validation croisée, garantissant un échantillonnage équilibré des classes dans chaque pli.

▪ Optimisation

L'optimisation des hyperparamètres se fait par validation croisée avec **GridSearchCV**. La classe **TunedThresholdClassifierCV** est utilisée pour ajuster les seuils de décision et optimiser les performances du modèle en utilisant les meilleures combinaisons de paramètres.

2. Traitement du déséquilibre des classes

Les modèles utilisent des poids de classe équilibrés, spécifiés dans les hyperparamètres par « **class_weight='balanced'** ». Cela permet de compenser le déséquilibre entre les classes en donnant plus d'importance à la classe minoritaire.

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

▪ Fonction coût métier

Cette fonction est définie sous le nom de **custom_score**. Elle pénalise plus sévèrement les **F**aux **N**égatifs (10x plus) que les **F**aux **P**ositifs. Cela reflète la priorité de minimiser les **FN** dans les décisions de crédit. En effet, un **FN** indique que le modèle a prédit que le client remboursera le prêt alors qu'il ne le fera pas, entraînant une perte financière plus importante pour l'entreprise.

- ### Algorithme d'optimisation

La fonction **custom_score** introduit un biais fort contre les **FN**, ce qui signifie que les hyperparamètres et le seuil de décision optimisé via **TunedThresholdClassifierCV** visent à ajuster la propension du modèle à faire des prédictions positives ou négatives.

- ### Métriques d'évaluation

F1-score : Le F1-score mesure l'équilibre entre la précision et le rappel, et peut montrer si la réduction des **FP** se fait au détriment du rappel, il est particulièrement utile lorsque les classes sont déséquilibrées. Bien que le modèle soit optimisé pour minimiser les **FN** (grâce à **custom_score**), le F1-score permet d'évaluer si cette optimisation impacte négativement le rappel (c'est-à-dire la capacité à détecter correctement les positifs).

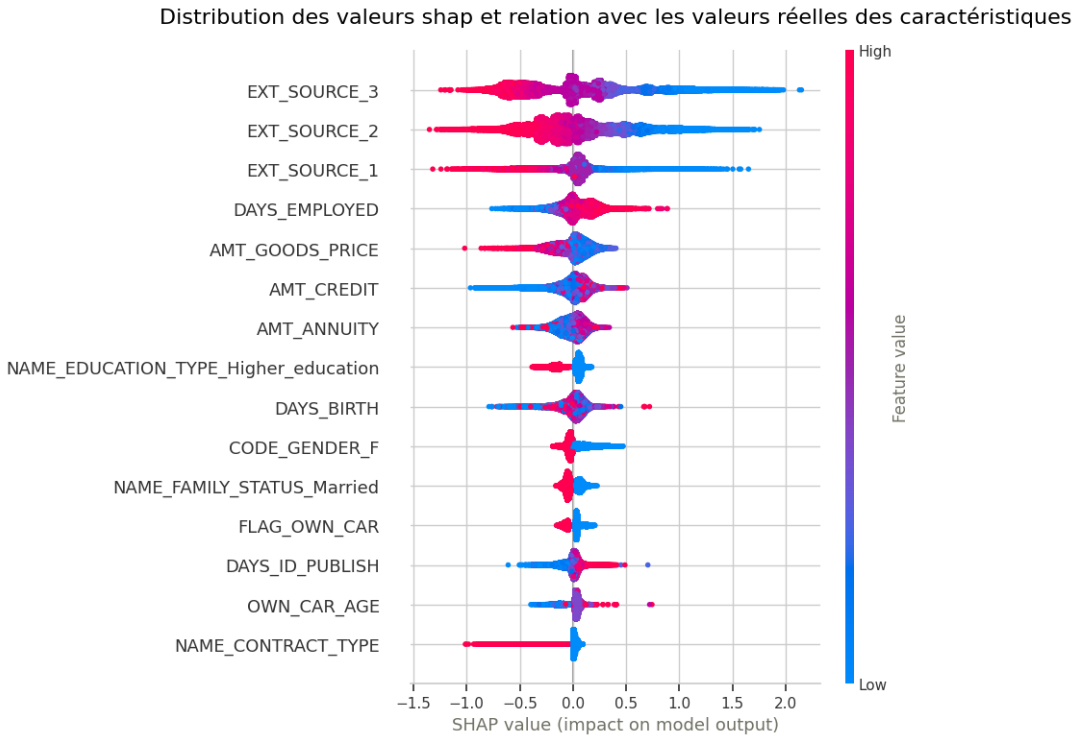
AUC (Area Under the Curve) : L'AUC mesure la capacité du modèle à discriminer entre les classes, ce qui est particulièrement utile dans le cas de classes déséquilibrées. Plutôt que de produire des prédictions binaires (0 ou 1), le modèle génère des probabilités entre 0 et 1, permettant d'évaluer la performance sur différents seuils de décision. Un AUC proche de 1 indique un modèle performant, capable de bien différencier les classes. En revanche, un AUC de 0,5 signifie que le modèle n'est pas meilleur qu'un choix aléatoire.

4. Tableau de synthèse des résultats

Id	Creation Time	Name	AUC	F1	Unite_de_cout
OC-146	2024-10-29	LGBM_rawdata_rfe	75.509	26.942	-32909.25
OC-144	2024-10-29	LGBM_rawdata_rfe	75.509	26.942	-32909.25
OC-143	2024-10-29	LGBM_rawdata_rfe	75.509	26.942	-32909.25
OC-141	2024-10-29	LogisticRegression_rawdata_rfe	74.426	25.82	-33788.25
OC-136	2024-10-25	LGBM_rawdata_rfe	73.121	0.0	-4969.0
OC-135	2024-10-25	LogisticRegression_rawdata_rfe	73.573	0.06	-4975.0

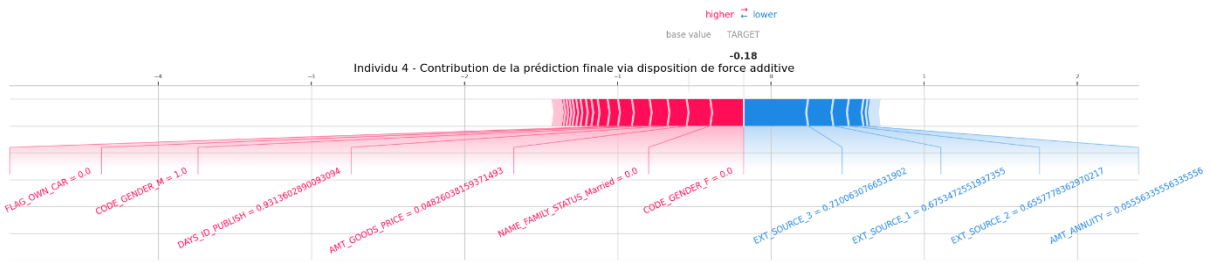
5. Interprétabilité globale et locale du modèle

▪ Interprétabilité globale



Le graphique met en évidence les principales caractéristiques influençant le modèle, en montrant comment leurs valeurs spécifiques impactent les prédictions. Les scores externes (**EXT_SOURCE**) et la stabilité de l'emploi (**DAYS_EMPLOYED**) apparaissent comme des indicateurs clés pour évaluer le risque de crédit. Cette visualisation permet de comprendre quels facteurs le modèle interprète comme des signaux de risque ou de stabilité.

▪ Interprétabilité locale



Ce graphique montre une interprétation locale de la prédiction de risque pour un individu spécifique. Les caractéristiques en rouge augmentent le risque de crédit, tandis que celles en bleu le réduisent. Pour cet individu, des facteurs comme **CODE_GENDER_F** et **NAME_FAMILY_STATUS_Married** augmentent le risque, tandis que les variables **EXT_SOURCE_1**, **EXT_SOURCE_2**, et **EXT_SOURCE_3** le réduisent.

6. Limites et améliorations possibles

▪ Limites du projet

Déséquilibre des classes : Même avec l'AUC et un seuil optimisé, la gestion du déséquilibre reste complexe, ce qui peut entraîner des biais.

Dépendance aux données externes (EXT_SOURCE) : Une grande part des décisions semble reposer sur ces variables, limitant la transparence si leur source n'est pas bien comprise.

Interprétation des modèles : Les modèles plus complexes peuvent être difficiles à expliquer aux non-experts, malgré l'utilisation des valeurs SHAP et des graphiques locaux/globaux.

Optimisation du seuil de décision : La fonction de coût personnalisée est spécifique au cas d'usage actuel et pourrait ne pas être généralisable à d'autres contextes de crédit.

▪ Améliorations possibles

Optimisation de l'imputation des données : Explorer d'autres techniques de traitement des valeurs manquantes.

Amélioration de la gestion des classes déséquilibrées : Utiliser des techniques comme le suréchantillonnage/sous-échantillonnage ou explorer des méthodes comme SMOTE pour améliorer la performance.

Diversification des variables explicatives : Ajouter d'autres caractéristiques pertinentes pour réduire la dépendance excessive aux variables externes et améliorer la robustesse du modèle.

Diversification des modèles : Tester d'autres modèles de Machine Learning ou de Deep Learning comme les réseaux de neurones.

Amélioration de l'interprétabilité : Intégrer des outils d'explication plus visuels et intuitifs, ou simplifier le modèle en utilisant des techniques comme la distillation de modèles.

Amélioration du MLOPS : Utiliser docker pour effectuer les tests unitaires et s'assurer du bon fonctionnement de l'application avant le déploiement.

Suivi continu des performances : Mettre en place une surveillance plus régulière pour détecter le drift de données et ajuster le modèle ou le seuil de décision en conséquence.

7. Analyse du Data Drift

Le rapport indique qu'**aucune dérive significative n'a été détectée** dans l'ensemble du jeu de données, avec un seuil de détection fixé à 0,5. Cependant, parmi les 240 colonnes, 10 montrent des signes de dérive, représentant 4,167 % des colonnes. Bien que le drift global soit faible, certaines colonnes, comme **DAYS_BIRTH** , présentent des **distances de Wasserstein** normées significatives (**7.356832**), suggérant un changement substantiel dans leur distribution. La distance de Wasserstein mesure le coût minimal nécessaire pour transformer une distribution de probabilité en une autre, fournissant ainsi une évaluation précise des différences entre les distributions d'origine et actuelles.