IMPLÉMENTATION D'UN MODÈLE DE SCORING

30/08/2024

Khoty WOLIE

SOMMAIRE

1. Problématique

Contexte :
Une société spécialisée dans les crédits à la consommation, y compris pour les personnes ayant peu ou pas d'historique de prêt, cherche à développer un outil de scoring. Cet outil doit permettre de déterminer la probabilité qu'un client rembourse son crédit. En outre, la société souhaite garantir la transparence du processus de scoring afin que les clients puissent comprendre les critères utilisés pour évaluer leur solvabilité.

Objectif :
Développer et mettre en place un outil de scoring crédit transparent qui évalue la probabilité de remboursement des prêts, en tenant compte des clients ayant peu ou pas d'historique de crédit.

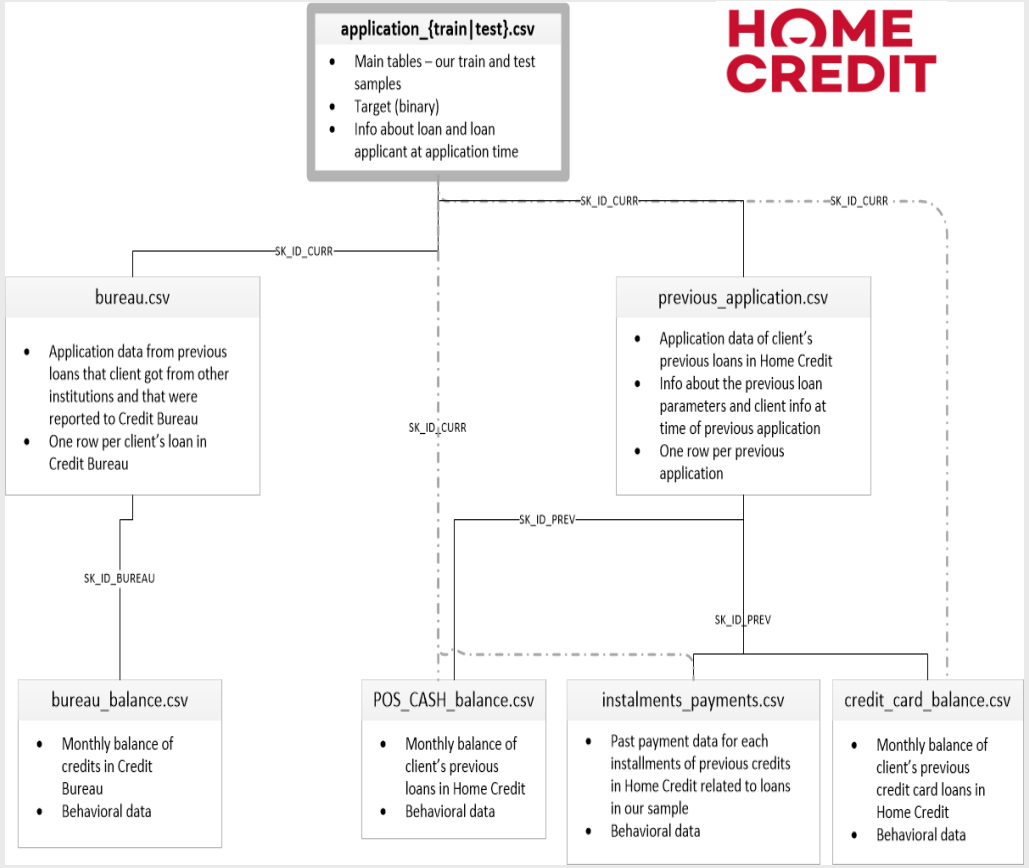
Mission :
Construire un modèle de scoring pour prédire automatiquement la probabilité de faillite d'un client. Analyser les features les plus contributives, tant au niveau global que local, pour garantir la transparence du score. Mettre en production le modèle via une API avec une interface de test, et adopter une approche MLOps complète, incluant le suivi des expérimentations et l'analyse du data drift en production.

2. Présentation des données

Source :
Les données proviennent de la compétition Kaggle "Home Credit Default Risk" (https://www.kaggle.com/c/home-credit-default-risk/data). Le jeu de données se compose de 8 fichiers tabulaires.

Description des données sources :

- **Jeu de données d'entraînement** (**application_train**) **:** Comprend 307 000 demandes de crédit avec l'issue correspondante (variable binaire "TARGET").
- **Jeu de données de test** (**application_test**) : Comprend 49 000 demandes de crédit sans l'issue connue.
- **219 variables :** Contiennent des informations détaillées (descriptions) sur les clients, telles que l'emploi, le cadre de vie, l'historique de crédit, la gestion des comptes bancaires, etc.



3. Préparation des données et Feature engineering

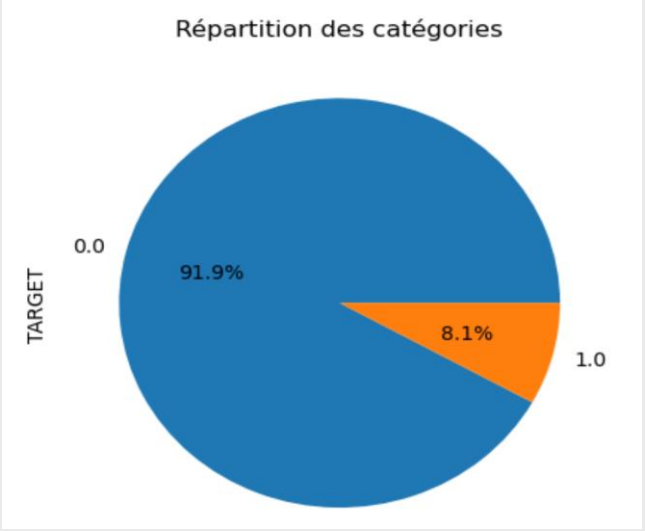
Basé sur le kernel Kaggle de jsaguiar :
Disponible à l'adresse suivante : https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script
Ce kernel exploite l'ensemble des tables de données, nettoie les valeurs aberrantes, et ajoute 226 nouvelles variables par features engineering, en transformant les principales caractéristiques métier, telles que **la vitesse de remboursement du crédit, le taux d'endettement**, etc.
Création et modification des variables (par calculs, **dummisation, factorisation**) à l'aide du code de Kaggle,
avec **jointure des fichiers**, réduction de la taille des fichiers pour GitHub (maximum 25 Mo),
et récupération des nouveaux clients pour l'application.

Dimensions du DataFrame utilisé pour la modélisation :
307 506 lignes et 796 colonnes

Valeurs manquantes :
25% des données présentent des valeurs manquantes.

Problème majeur :
Déséquilibre du jeu de données pour la variable binaire prédite.

Un ensemble de données aux catégories déséquilibrées



4. Développement d'un modèle prédictif :

4.1- Sélection de métrique et prise en main du modèle

Utilisation de **MLFlow** pour suivre et comparer les différentes exécutions, et création de deux métriques pour accorder plus d'importance aux faux négatifs.

- **Métrique 1:** Mesure personnalisée (calculate_custom_metric)

Le déséquilibre du coût métier entre un faux négatif (FN - mauvais client prédit bon client : donc crédit accordé et perte en capital) et un faux positif (FP - bon client prédit mauvais : donc refus crédit et manque à gagner en marge).Nous pourrons supposer, par exemple, que le coût d'un FN est dix fois supérieur au coût d'un FP. Nous créerons un score "métier" (minimisation du coût d'erreur de prédiction des FN et FP)

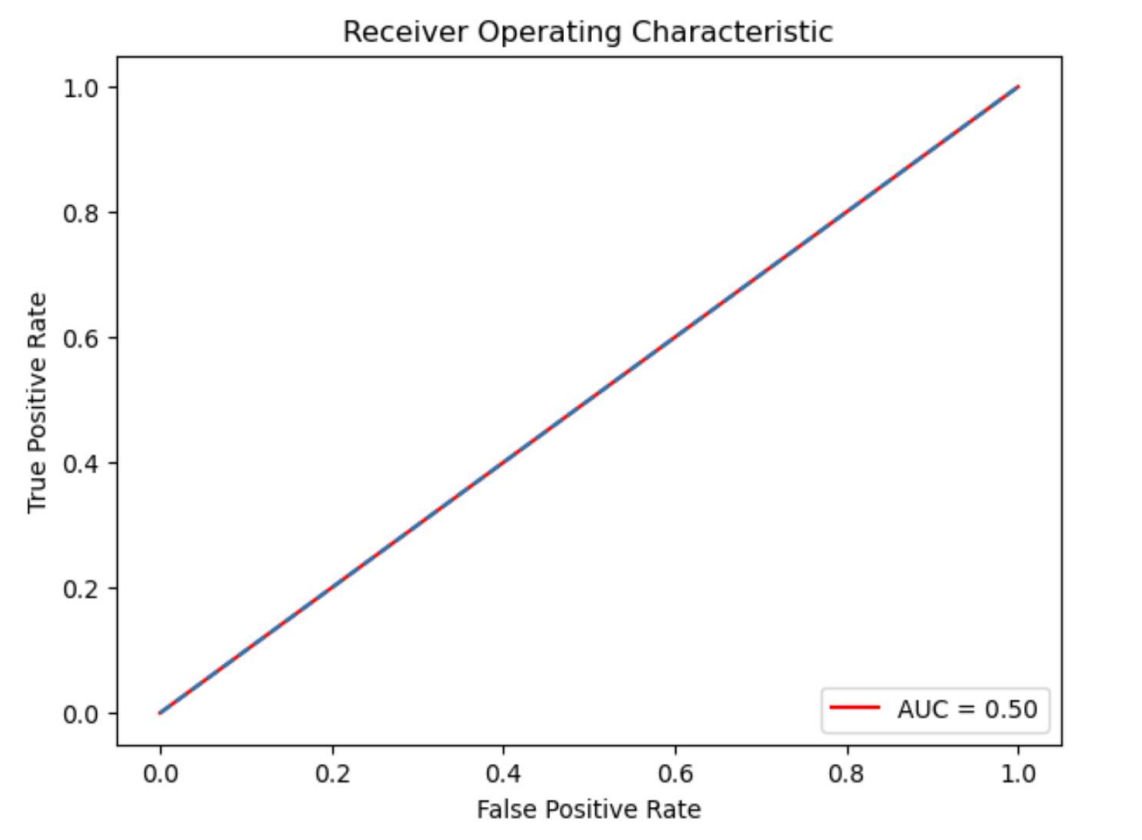
Etape	Seuil de probabilité	Génération des étiquettes	Matrice de confusion	Pénalisation des faux négatifs	Sélection du meilleur seuil
Description	Itération sur les seuils de probabilité de 0,30 à 1,00.	Génération des étiquettes binaires en comparant les probabilités prédites au seuil actuel.	Calcul de la matrice de confusion pour obtenir les VP, VN, FP, FN.	Les faux négatifs sont multipliés par 10 pour leur donner un poids plus important.	Identification du seuil qui minimise le score de pénalité.

❑ **Métrique 2:** la meilleure probabilité de seuil (best_threshold_probability)

Etape	Détermination du meilleur seuil	Pénalisation des faux négatifs	Retour du meilleur
Description	Itération sur les seuils de probabilité pour minimiser la métrique personnalisée.	Les faux négatifs sont pénalisés pour orienter le modèle à mieux identifier les clients à risque.	Retour du seuil de probabilité optimal qui minimise le score de pénalité.

❖ Ces deux métriques sont créées pour ajuster la prédiction du modèle afin de minimiser le risque d'un faux négatif. Dans notre contexte, cela signifie réduire les chances que le modèle rate l'identification d'un client à haut risque de défaut de paiement, ce qui est crucial pour la gestion des risques dans les institutions financières.

- **La Baseline** dans un modèle supervisé sert de référence pour mesurer les performances minimales et comparer l'efficacité des modèles plus complexes. J'ai utilisé le **DummyClassifier** pour établir cette Baseline, qui montre le score qu'un algorithme simple pourrait obtenir en faisant des prédictions. Cette Baseline sera utilisée pour évaluer les performances de mon modèle.



DummyClassifier

```
DummyClassifier(strategy='most_frequent')
```

Pourquoi utiliser un classifieur naïf ?

Valider que la forme des données est compatible avec une modélisation.

4.2- Analyse du déséquilibre des données et sélection d'un modèle de machine learning

❑ Comparaison des différentes méthodes de gestion du déséquilibre des données.

En raison de limitations en puissance de calcul, je commence par entraîner des modèles simples, avant d'optimiser les hyperparamètres pour le modèle le plus performant.

La méthode Class_weight

Class_weight est une technique utilisée pour gérer le déséquilibre des données dans les modèles de machine learning supervisés. Lorsqu'une classe est sous-représentée dans les données, cela peut entraîner un biais du modèle en faveur de la classe majoritaire. Cette méthode attribue des poids différents à chaque classe, donnant plus d'importance aux classes minoritaires. Ainsi, le modèle est encouragé à accorder une attention proportionnée à toutes les classes, ce qui améliore les performances de classification pour les classes sous-représentées.

	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
0	DummyClassifier	0	795	0.852	0.852	0.499	0.500	0.490	0.30
1	LogisticRegression	16	179	0.519	0.520	0.569	0.571	0.467	0.66
2	RandomForestClassifier	170	795	1.000	0.919	1.000	0.725	0.000	0.30
3	GradientBoostingClassifier	127	179	0.919	0.919	0.684	0.673	0.466	0.30
4	LGBMClassifier	39	795	0.734	0.722	0.838	0.784	0.348	0.58

La méthode Undersampling

Undersampling est une méthode de gestion du déséquilibre des données qui consiste à réduire la taille de la classe majoritaire en supprimant aléatoirement des échantillons pour équilibrer la distribution des classes. Cela permet de rendre les classes majoritaires et minoritaires plus équilibrées, ce qui peut aider à éviter que le modèle ne soit biaisé en faveur de la classe dominante. Cependant, cette technique peut également entraîner une perte d'information, car elle élimine des données potentiellement utiles.

	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
0	DummyClassifier	0	795	0.496	0.500	0.496	0.500	0.848	0.30
1	LogisticRegression	3	182	0.529	0.533	0.555	0.567	0.500	0.30
2	RandomForestClassifier	20	795	1.000	0.679	1.000	0.745	0.000	0.39
3	GradientBoostingClassifier	26	182	0.642	0.639	0.694	0.690	0.507	0.30
4	LGBMClassifier	11	795	0.596	0.564	0.886	0.776	0.307	0.76

La méthode Oversampling

L’ **oversampling** est une méthode de gestion du déséquilibre des données qui consiste à augmenter artificiellement la taille de la classe minoritaire en dupliquant des échantillons existants ou en générant de nouveaux échantillons similaires. Cette approche permet de rendre les classes plus équilibrées, aidant ainsi le modèle à apprendre de manière plus équitable sur toutes les classes. L'oversampling peut améliorer la performance du modèle sur la classe minoritaire, mais il présente aussi le risque de surapprentissage (overfitting) si les échantillons sont simplement dupliqués.

	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
0	DummyClassifier	1	179	0.500	0.503	0.497	0.508	0.634	0.30
1	LogisticRegression	17	179	0.585	0.586	0.571	0.574	0.467	0.65
2	RandomForestClassifier	65	179	1.000	0.919	1.000	0.646	0.000	0.30
3	GradientBoostingClassifier	214	179	0.638	0.635	0.685	0.673	0.446	0.61
4	LGBMClassifier	5	179	0.666	0.657	0.731	0.677	0.429	0.61

La méthode SMOTE

Le **SMOTE (Synthetic Minority Over-sampling Technique)** est une méthode d'oversampling utilisée pour gérer le déséquilibre des données en créant de nouveaux échantillons synthétiques pour la classe minoritaire. Plutôt que de simplement dupliquer des échantillons existants, SMOTE génère de nouvelles instances en interpolant entre les échantillons minoritaires existants. Cela permet de créer une classe minoritaire plus diversifiée et équilibrée par rapport à la classe majoritaire, ce qui aide le modèle à mieux généraliser et à éviter les biais en faveur de la classe dominante. SMOTE est particulièrement efficace pour améliorer la performance des modèles sur des ensembles de données déséquilibrés sans risquer de surapprentissage excessif.

	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
0	DummyClassifier	5	179	0.500	0.503	0.497	0.508	0.634	0.30
1	LogisticRegression	21	179	0.579	0.579	0.569	0.572	0.467	0.67
2	RandomForestClassifier	61	179	1.000	0.919	1.000	0.616	0.000	0.30
3	GradientBoostingClassifier	414	179	0.919	0.919	0.649	0.646	0.465	0.30
4	LGBMClassifier	13	179	0.919	0.919	0.710	0.677	0.464	0.30

❑ Comparaison des approches Class_Weight, Undersampling, Oversampling, et SMOTE pour le traitement des données déséquilibrées

Je vais analyser les résultats en me basant sur trois métriques principales : l'**Accuracy**, l'**AUC (Area Under the ROC Curve),** et la **Métrique personnalisée (Metrique_custom_Train).**

Le **LGBMClassifier** avec la méthode **CLASS_WEIGHT** se distingue comme le modèle le plus performant, offrant les meilleurs résultats en termes **d'AUC (0,784)** et de **métrique personnalisée (0,348)** tout en évitant l'overfitting. Bien que le RandomForestClassifier avec Undersampling obtienne une meilleure Accuracy, son AUC est inférieure. SMOTE et Oversampling, bien que compétitifs, restent légèrement en retrait. Ainsi, je recommande l'utilisation du LGBMClassifier avec CLASS_WEIGHT pour gérer le déséquilibre des classes dans cet ensemble de données.

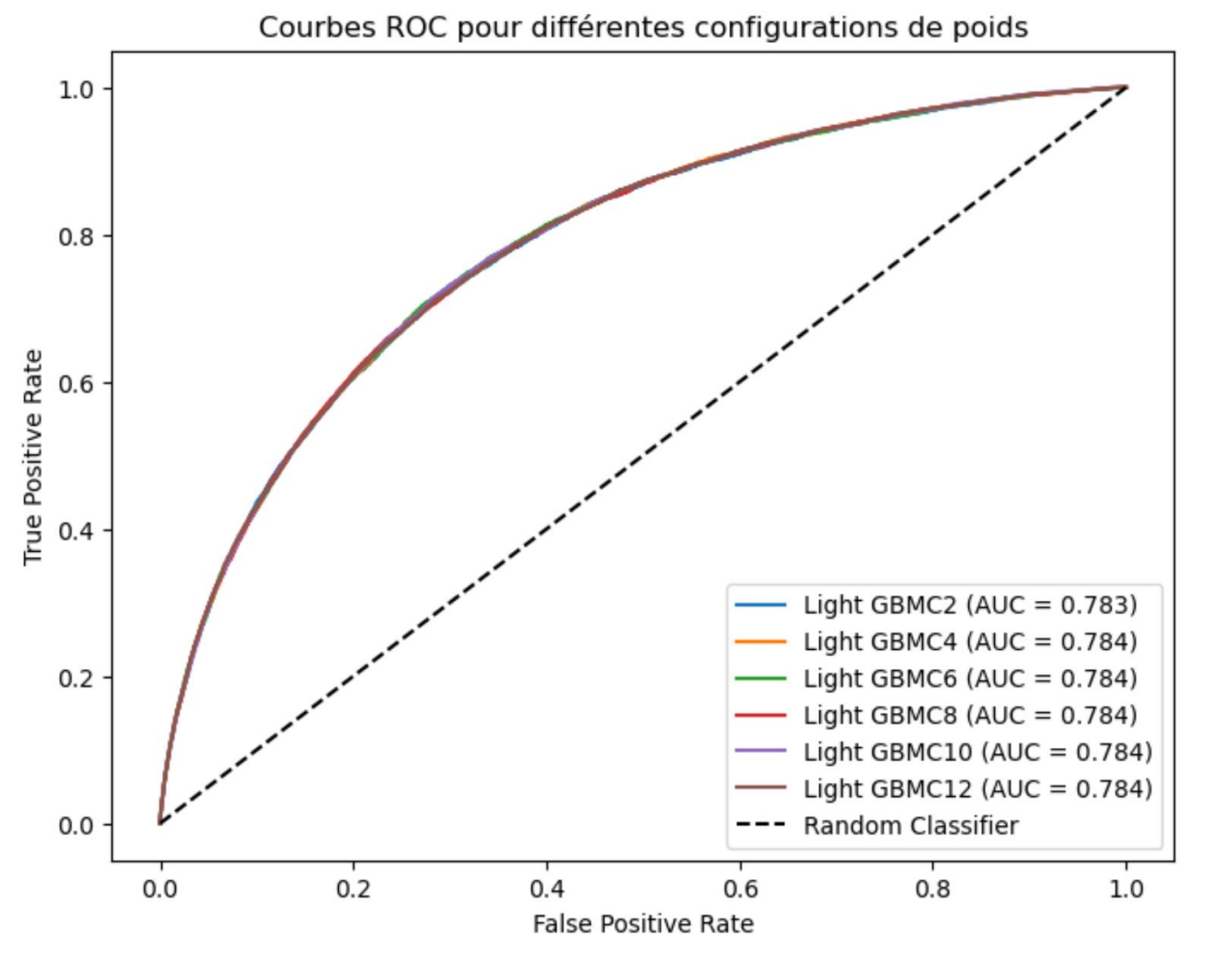
	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
4	WEIGHT	39	795	0.734	0.722	0.838	0.784	0.348	0.58
4	UNDERSAMPLING	11	795	0.596	0.564	0.886	0.776	0.307	0.76
4	OVERSAMPLING	5	179	0.666	0.657	0.731	0.677	0.429	0.61
4	SMOTE	13	179	0.919	0.919	0.710	0.677	0.464	0.30

❑ Détermination du poids du modèle sélectionné

Meilleure Accuracy : Le ratio WEIGHT_1:2 avec une accuracy de **0.917** est le meilleur.
Meilleure AUC : Les ratios WEIGHT_1:4 à WEIGHT_1:12 offrent une AUC stable de **0.784**.
Métrique personnalisée : Les ratios WEIGHT_1:4 à WEIGHT_1:8 montrent de bonnes performances avec une métrique personnalisée autour de **0.345** à **0.347**.
Seuil de probabilité : Les modèles avec des poids plus élevés (WEIGHT_1:10 et WEIGHT_1:12) ont des seuils plus élevés, ce qui peut être utilisé pour ajuster la sensibilité du modèle.
Modèle Globalement Optimal : Le modèle avec le ratio WEIGHT_1:4 semble offrir le meilleur compromis entre accuracy, AUC, et la métrique personnalisée. Ce ratio devrait être notre premier choix, surtout si nous voulons équilibrer la précision globale avec une bonne gestion des faux négatifs.
Ajustement du Seuil : Si nous souhaitons réduire les faux positifs, nous pourrions envisager d'ajuster

	nom_modele	Run_Duree_Model	nb_variables_utilisees	Accuracy_Train	Accuracy_Test	Auc_Train	Auc_Test	Metrique_custom_Train	Best_pourcentage_seui_proba
0	WEIGHT_1:2	33	795	0.922	0.917	0.84	0.783	0.36	0.3
1	WEIGHT_1:4	34	795	0.898	0.889	0.838	0.784	0.345	0.31
2	WEIGHT_1:6	35	795	0.859	0.849	0.837	0.784	0.347	0.41
3	WEIGHT_1:8	35	795	0.817	0.805	0.837	0.784	0.347	0.48
4	WEIGHT_1:10	34	795	0.774	0.763	0.838	0.784	0.348	0.54
5	WEIGHT_1:12	35	795	0.734	0.722	0.838	0.784	0.348	0.58

❑ Visualisation des Résultats



La courbe ROC indique que tous les modèles ont des performances très similaires en termes de séparation des classes. Cela signifie que, pour notre cas d'utilisation, d'autres critères de performance devraient être pris en compte pour choisir la configuration optimale tel que du temps de calcul (Run_Duree_Model), ou de la complexité du modèle (nb_variables_utilisees)...

5. Optimisation du modèle en fonction d'une fonction de coût spécifique au métier et des hyperparamètre

❑ Hypothèses pour les candidats au crédit

Objet	Coût par client	Classe (positif = client en défaut)
Attribution d'un crédit à un client qui fait défaut	100	Faux négatif (FN)
Attribution d'un crédit à un client qui ne fait pas défaut	-10	Vrai négatif (TN)
Refus de crédit à un client qui aurait fait défaut	0	Vrai positif (TP)
Refus de crédit à un client qui n'aurait pas fait défaut	0	Faux positif (FP)
Frais généraux par client	1	-

$$\text{Coût} = \frac{100 \times FN - 10 \times TN + 1 \times (TP + TN + FP + FN)}{TP + TN + FP + FN}$$

Cette fonction évalue les coûts des erreurs et des frais généraux par client, en soulignant particulièrement le coût élevé des faux négatifs (clients en défaut de paiement).

❑ Modèle choisi : Light Gradient Boosting Machine (LightGBM)

LightGBM est un algorithme de boosting sur des forêts aléatoires, optimisé pour la performance et l'efficacité.

Particularités :

- Regroupement des variables continues en classes (binning), d'où l'appellation « Light ».
- Croissance des arbres par feuille plutôt que par profondeur, ce qui permet une convergence rapide.
- Gestion intelligente des valeurs manquantes (NaN) : elles sont ignorées lors du split et allouées de manière optimale après.

LightGBM a été créé en 2017 par une équipe de chercheurs et d'ingénieurs de Microsoft. Le projet a été développé au sein de Microsoft Research en tant qu'algorithme de machine learning pour améliorer les performances des modèles de boosting de gradient sur les grands ensembles de données. La motivation principale était de concevoir un algorithme plus rapide et plus efficace que les alternatives existantes, comme XGBoost, tout en maintenant ou en améliorant la précision du modèle.

❑ Choix des hyperparamètres

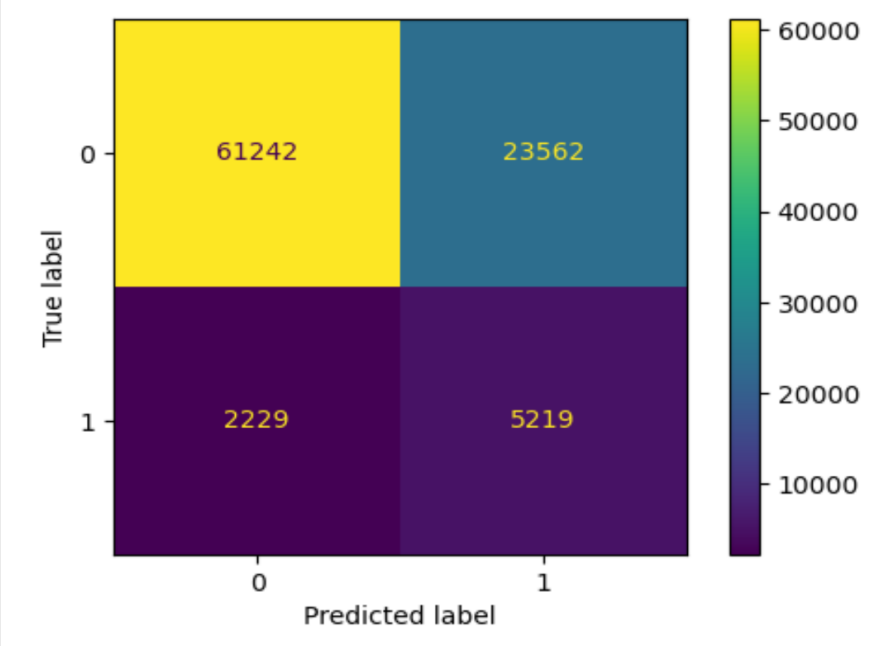
Les **hyperparamètres** sont des paramètres configurés avant l'entraînement d'un modèle de machine learning, et leur ajustement est crucial pour optimiser les performances du modèle.
En raison de limitations en puissance de calcul, je me limite aux hyperparamètres sélectionnés.

Hyperparamètre optimisé	Valeur optimum	Description
n_estimators	1000	Nombre d'arbres dans le modèle, influence la complexité et la performance.
learning_rate	0.01	Taux d'apprentissage, contrôle la contribution de chaque arbre au modèle global.
max_depth	7	Profondeur maximale des arbres, limite la capacité de modélisation pour éviter le surapprentissage.
min_child_weight	1	Poids minimum des feuilles, réduit la variance en imposant une contrainte sur la taille des feuilles.
reg_alpha	0.2	Régularisation L1 (Lasso) : Pénalise les poids des caractéristiques pour encourager la parcimonie.

Réduction du nombre de variables : Sélection des variables présentant moins de 70% de valeurs manquantes (NaNs) afin de minimiser les biais potentiels et d'améliorer la qualité des données utilisées dans le modèle.

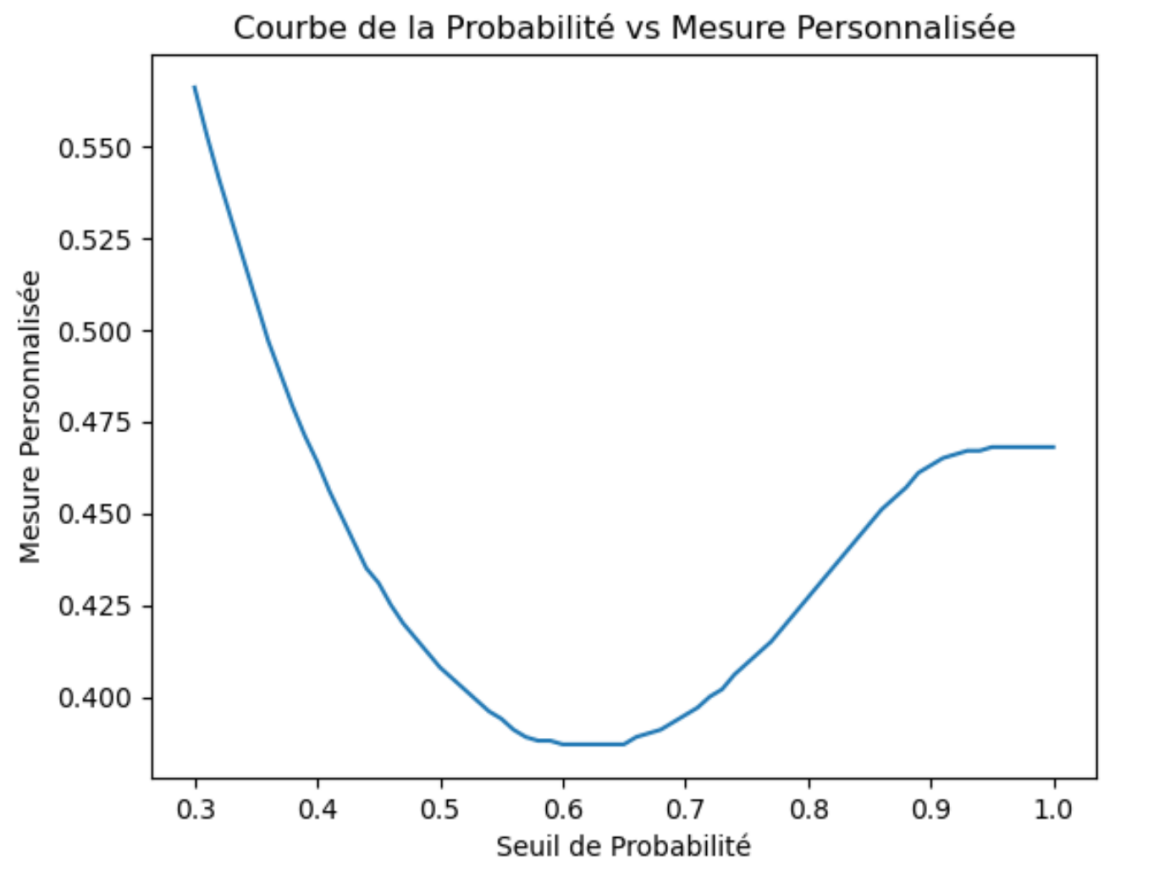
Optimisation du modèle LGBMClassifier : Mise en œuvre d'une optimisation en trois étapes du modèle LGBMClassifier en utilisant la technique de GridSearchCV, pour identifier les meilleures combinaisons d'hyperparamètres et maximiser les performances du modèle.

Analyse des performances : Évaluation approfondie des résultats du modèle à l'aide de la matrice de confusion, permettant d'identifier les erreurs de classification, notamment les faux positifs et faux négatifs, et d'ajuster les stratégies de modélisation en conséquence.



- **Taux de Vrai Négatif (61 242)** est élevé, ce qui indique que le modèle est bon pour identifier les clients qui ne feront pas défaut.
- **Nombre de Faux Positifs (23 562)** est relativement élevé, ce qui pourrait indiquer un besoin de réajuster le seuil de décision si les faux positifs sont trop coûteux pour l'entreprise.
- **Nombre de Faux Négatifs (2 229),** bien que plus faible que les faux positifs, est critique car ces erreurs sont souvent les plus coûteuses pour une entreprise qui accorde des crédits.
- **Taux de Vrai Positif (5 219)** est significatif mais pourrait être amélioré, surtout en réduisant les faux négatifs.

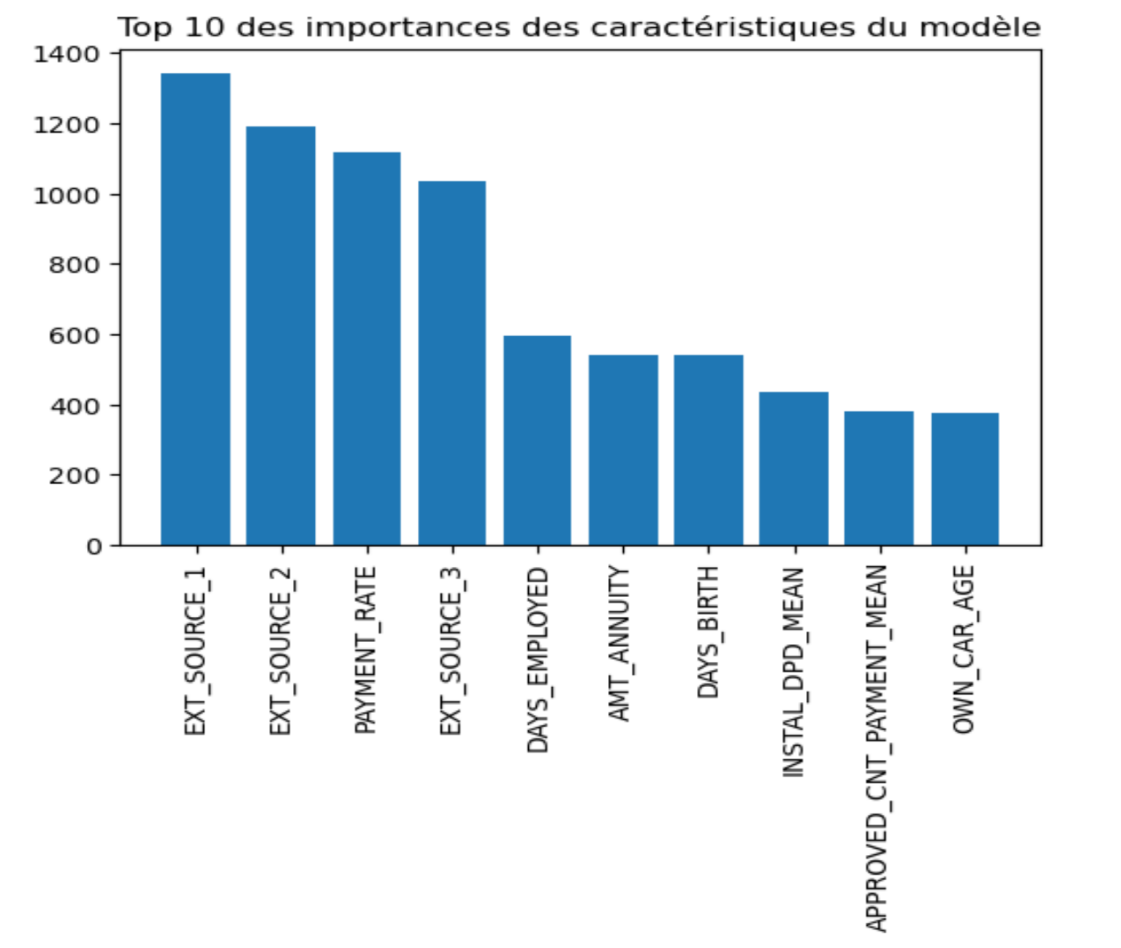
❑ Analyse de l'ajustement du seuil de probabilité pour optimiser la métrique personnalisée.



- **Seuil Optimal** : Le seuil optimal pour ce modèle semble être autour de 0.6, où la mesure personnalisée est la plus basse, ce qui suggère un bon équilibre entre faux négatifs et faux positifs, et donc un compromis optimal entre la sensibilité et la spécificité.
- **Impact des Ajustements de Seuil** : Des seuils de probabilité plus bas ou plus élevés que ce point optimal entraînent une augmentation de la mesure personnalisée, signalant une détérioration des performances du modèle en termes de gestion des coûts liés aux erreurs.

- Exportation du modèle final au format **joblib** et **pickle** à des fins pédagogiques, pour utilisation dans mon API Flask.

6. Importance des Features à l'échelle globale et locale



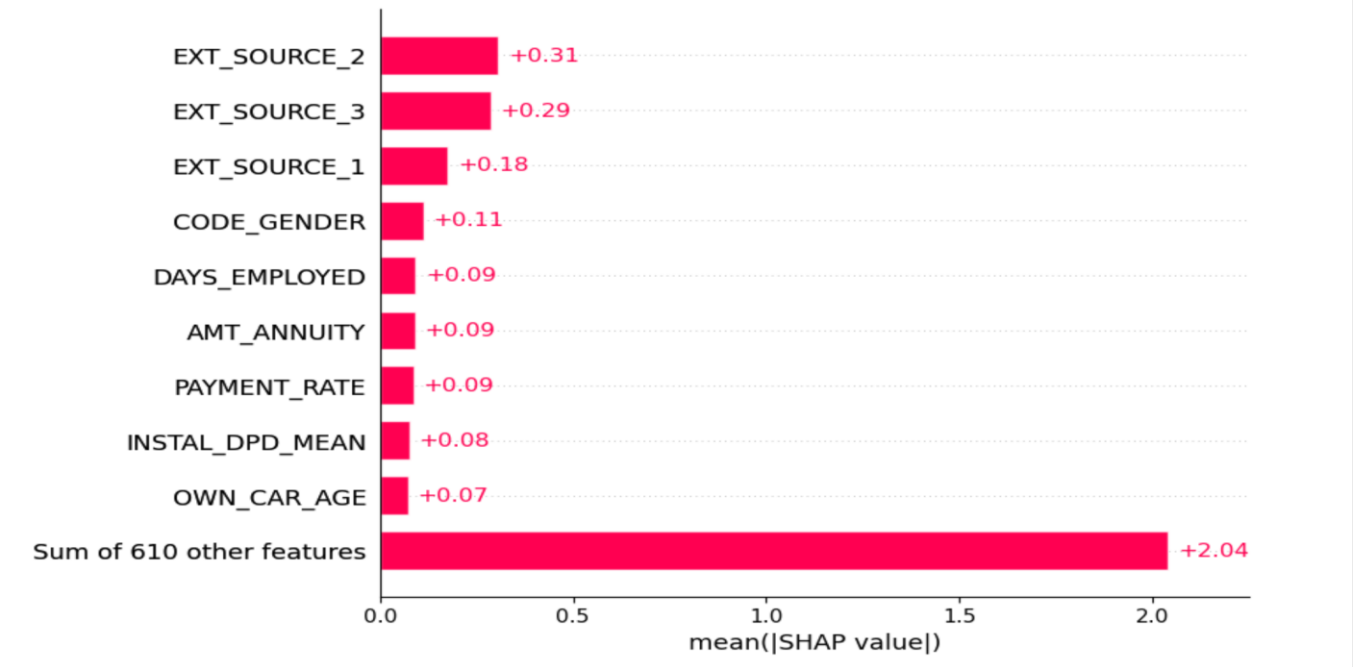
- Le graphique montre les **10 caractéristiques les plus influentes dans le modèle**. Les sources de données externes (**EXT_SOURCE_1, 2, 3**) dominent, suivies par des indicateurs financiers tels que le taux de paiement et l'annuité. La durée d'emploi et l'âge des clients sont également des facteurs clés dans la prédiction du modèle. Ces variables sont essentielles pour évaluer le risque de crédit des clients.

❑ SHAP globale

Ce graphique montre les 10 principales caractéristiques influençant le modèle, classées selon leur importance SHAP (SHapley Additive exPlanations).

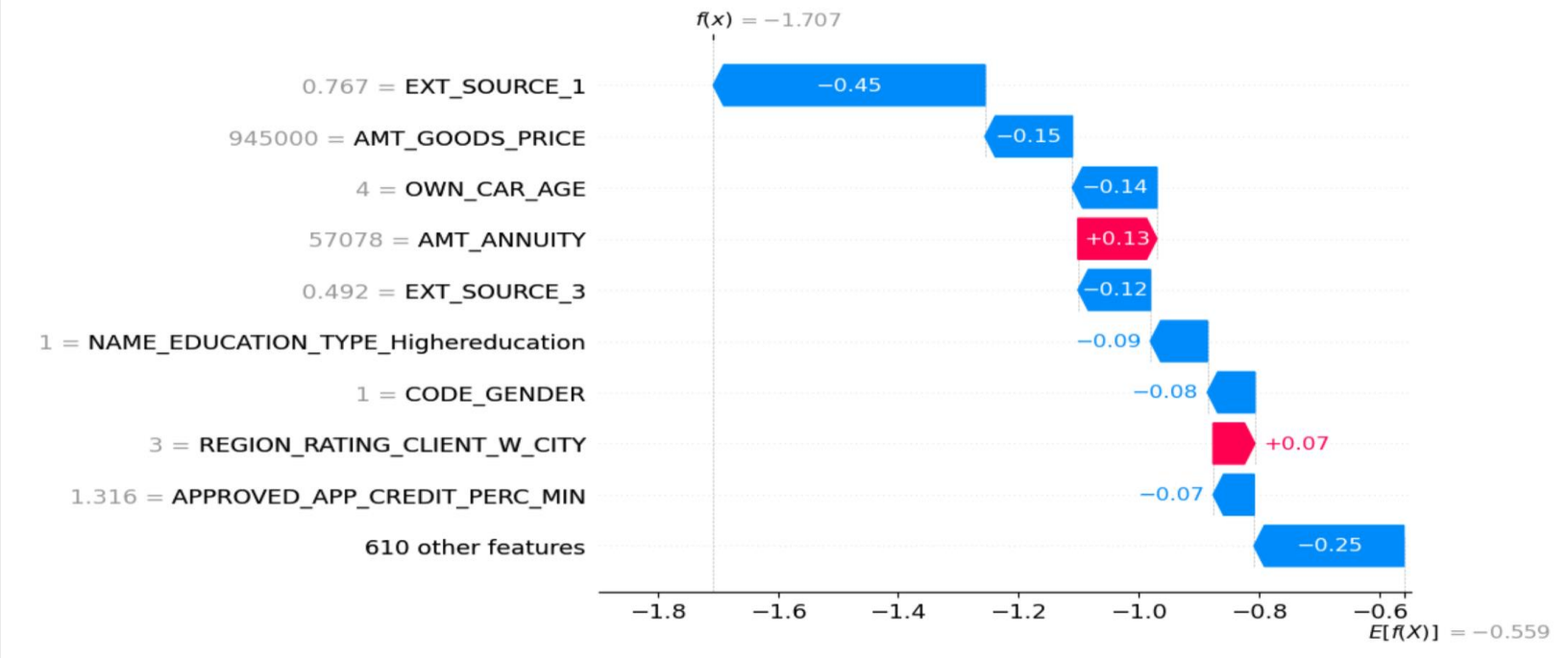
- **EXT_SOURCE_2**, **EXT_SOURCE_3**, et **EXT_SOURCE_1** : Ces sources externes sont les plus déterminantes, ayant la plus grande influence sur les prédictions du modèle.
- **CODE_GENDER**, **DAYS_EMPLOYED**, et **AMT_ANNUITY** : Ces caractéristiques démographiques et financières jouent également un rôle clé dans la décision du modèle.
- **PAYMENT_RATE**, **INSTAL_DPD_MEAN**, et **OWN_CAR_AGE** : Ces variables liées aux comportements de paiement et à l'âge des biens sont aussi significatives mais à un degré moindre.

Contribution Cumulative des Autres Variables : Les 610 autres caractéristiques combinées ont une influence importante mais sont individuellement moins déterminantes.



❑ SHAP locale

- **EXT_SOURCE_1** et **AMT_GOODS_PRICE** ont des effets négatifs importants, ce qui signifie que ces caractéristiques réduisent la probabilité que ce client soit classé dans une certaine catégorie (par exemple, être approuvé pour un prêt).
- **OWN_CAR_AGE**, **REGION_RATING_CLIENT_W_CITY** et d'autres caractéristiques rouges ont des effets positifs, ce qui signifie qu'elles augmentent la probabilité de cette classification.
- **610 autres caractéristiques** : Cette mention indique que le modèle utilise de nombreuses autres caractéristiques qui, combinées, ont une contribution totale non négligeable. Cependant, elles ne sont pas listées individuellement pour simplifier la visualisation.

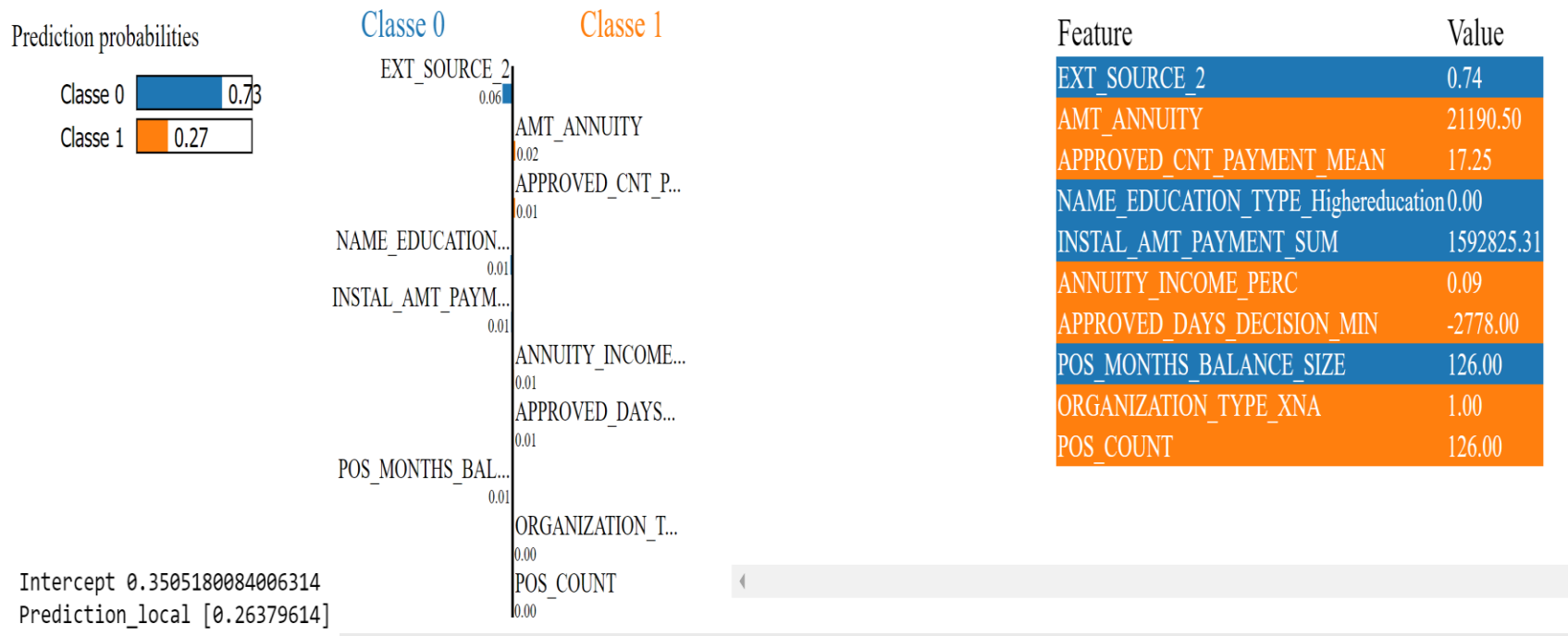


❑ Avec LIME

Le graphique explique la prédiction locale du modèle.

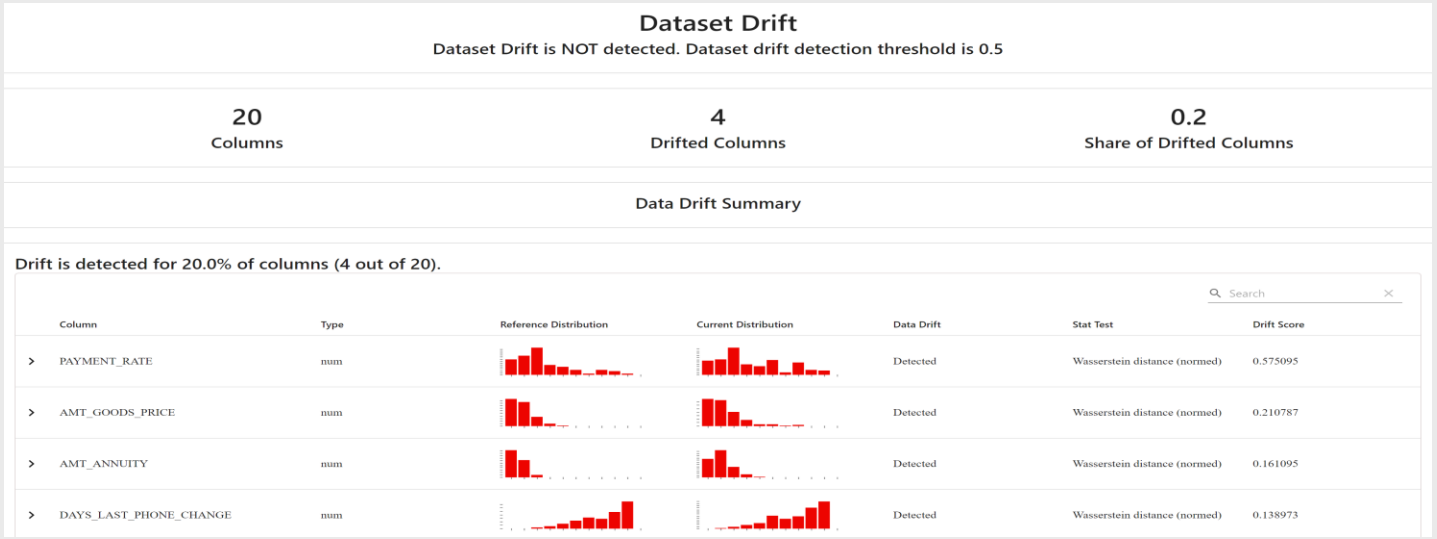
- Intercept (0.3505) : Point de départ de la prédiction, avant l'ajout des contributions des caractéristiques.
- Prédiction Locale (0.2637) : Valeur ajustée de la prédiction après prise en compte des caractéristiques spécifiques.
- Influence des Caractéristiques :
    - Classe 0 (Bleu, 73%) : Les caractéristiques comme EXT_SOURCE_2 (0.74) poussent la prédiction vers la classe 0.
    - Classe 1 (Orange, 27%) : Des caractéristiques comme APPROVED_CNT_PAYMENT_MEAN (17.25) poussent la prédiction vers la classe 1.

Conclusion : Le modèle estime une probabilité plus élevée (73%) pour la classe 0, influencée principalement par EXT_SOURCE_2. Ce type de visualisation est essentiel pour comprendre les raisons derrière une décision du modèle, comme l'approbation ou le rejet d'un crédit.



7. Déploiement d'une API de prédiction et d'un Dashboard interactif

❑ Analyse du DataDrift réalisée sur les 20 variables les plus influentes issues des features_importances.



Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

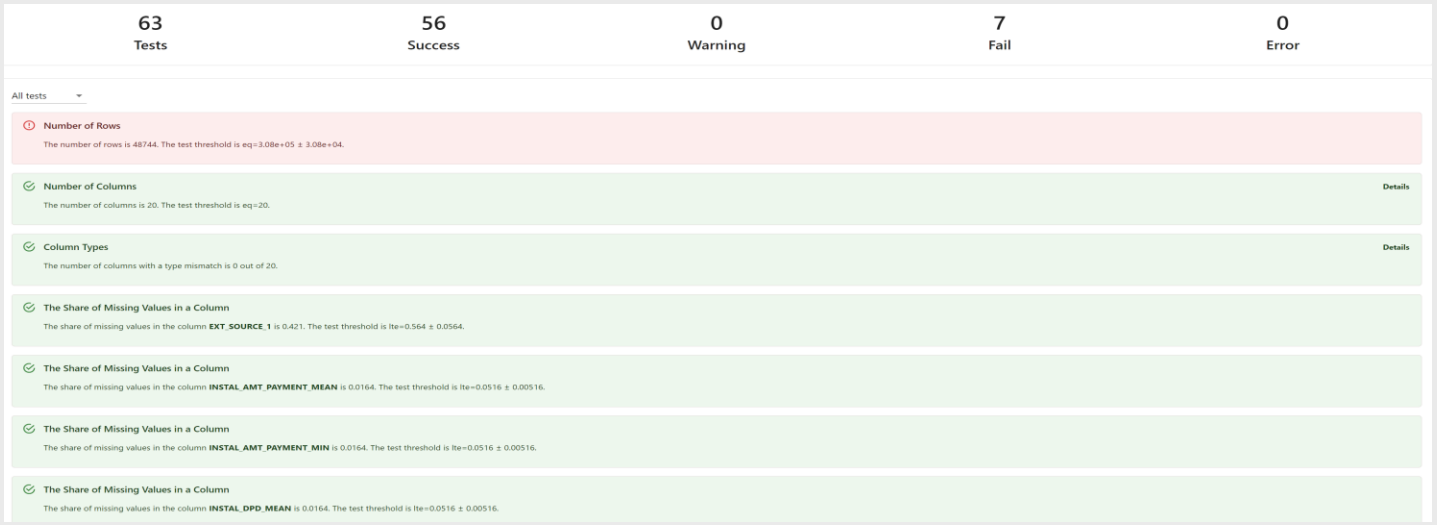
20	4	0.2
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 20.0% of columns (4 out of 20).

Search

	Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
>	PAYMENT_RATE	num			Detected	Wasserstein distance (normed)	0.575095
>	AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210787
>	AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161095
>	DAYS_LAST_PHONE_CHANGE	num			Detected	Wasserstein distance (normed)	0.138973



63	56	0	7	0
Tests	Success	Warning	Fail	Error

All tests

- Number of Rows
  The number of rows is 48744. The test threshold is eq=3.08e+05 ± 3.08e+04.
- Number of Columns — Details
  The number of columns is 20. The test threshold is eq=20.
- Column Types — Details
  The number of columns with a type mismatch is 0 out of 20.
- The Share of Missing Values in a Column
  The share of missing values in the column **EXT_SOURCE_1** is 0.421. The test threshold is lte=0.564 ± 0.0564.
- The Share of Missing Values in a Column
  The share of missing values in the column **INSTAL_AMT_PAYMENT_MEAN** is 0.0164. The test threshold is lte=0.0516 ± 0.00516.
- The Share of Missing Values in a Column
  The share of missing values in the column **INSTAL_AMT_PAYMENT_MIN** is 0.0164. The test threshold is lte=0.0516 ± 0.00516.
- The Share of Missing Values in a Column
  The share of missing values in the column **INSTAL_DPD_MEAN** is 0.0164. The test threshold is lte=0.0516 ± 0.00516.

- Le **DataDrift** a été détecté dans 4 des 20 colonnes analysées (20%). Cela indique une différence notable entre les distributions des données de référence et actuelles, ce qui peut affecter la performance du modèle. Une surveillance continue est nécessaire pour évaluer l'impact et, si besoin, réentraîner ou ajuster le modèle.

- Sur **63 tests de stabilité** des données, 56 ont réussi, montrant une bonne cohérence générale. Les 7 échecs, principalement liés au nombre de lignes, suggèrent une variabilité dans la taille des échantillons. Les colonnes et types de données sont conformes, et les valeurs manquantes restent dans les limites acceptables. Les échecs méritent une attention pour assurer la stabilité globale.

❑ Différents tests de prédiction effectués avec MLFlow, ainsi qu'à travers l'API Flask en local .

```
C:\WINDOWS\system32>cd C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook

C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook>python server_local.py
C:\Python39\lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator LabelEncoder from version 1.5.0 when using version 1.5.1. This might lead to breaking code
or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'server_local'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:3000
 * Running on http://192.168.1.85:3000
Press CTRL+C to quit
127.0.0.1 - - [27/Aug/2024 21:05:07] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:05:23] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:05:36] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:07:52] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:08:17] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:08:26] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:09:10] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:09:14] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 21:59:26] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:21:27] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:21:50] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:21:55] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:29:29] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:29:44] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:29:53] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:00] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:06] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:14] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:22] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:27] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:33] "POST /api/ HTTP/1.1" 200 -
127.0.0.1 - - [27/Aug/2024 22:30:41] "POST /api/ HTTP/1.1" 200 -
```


```
Réponse brute du serveur : [[0.63895802188395,0.36104197811605]]

Prédiction : 36
```

- **Réponse Brute du Serveur** : `[[0.63895802188395, 0.36104197811605]]`
- **Classe 0 (probabilité = 63.89%)** : Selon notre modèle, il y a 63.89% de chances que l'observation appartienne à la classe 0.
- **Classe 1 (probabilité = 36.10%)** : Il y a 36.10% de chances que l'observation appartienne à la classe 1.

- **Développement d'une API Flask en Python, hébergée sur Heroku, pour la prédiction des demandes de prêts.**

Les différentes démarches

Créations d’un Bucket S3 pour héberger le modèle sur AWS

```
C:\WINDOWS\system32>aws configure
AWS Access Key ID [None]: AKIAST6S7IRG2LV3ZVOI
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json

C:\WINDOWS\system32>aws s3 ls

C:\WINDOWS\system32>aws s3 mb s3://mon-bucket-test
make_bucket failed: s3://mon-bucket-test An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The requested bucket name is not available. The bucket namespace is shared by
 all users of the system. Please select a different name and try again.

C:\WINDOWS\system32>aws configure
AWS Access Key ID [**************ZVOI]:
AWS Secret Access Key [**************xf6O]:
Default region name [us-east-1]: eu-west-3
Default output format [json]:

C:\WINDOWS\system32>aws s3 ls

C:\WINDOWS\system32>aws s3 mb s3://votre-nom-de-bucket --region eu-west-3
make_bucket: votre-nom-de-bucket

C:\WINDOWS\system32>aws s3 ls
2024-08-28 23:47:06 votre-nom-de-bucket

C:\WINDOWS\system32>aws s3 cp "C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifact
s\model" s3://votre-nom-de-bucket/model/ --recursive
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\MLmodel to s
3://votre-nom-de-bucket/model/MLmodel
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\python_env.y
aml to s3://votre-nom-de-bucket/model/python_env.yaml
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\conda.yaml t
o s3://votre-nom-de-bucket/model/conda.yaml
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\registered_m
odel_meta to s3://votre-nom-de-bucket/model/registered_model_meta
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\requirements
.txt to s3://votre-nom-de-bucket/model/requirements.txt
upload: ..\..\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku\mlflow_runs\290555362347125930\1e46374402274ffe9572106d93203ef9\artifacts\model\model.pkl to
 s3://votre-nom-de-bucket/model/model.pkl
```

```
C:\WINDOWS\system32>aws s3 ls s3://votre-nom-de-bucket/model/
2024-08-28 23:52:42        549 MLmodel
2024-08-28 23:52:42        459 conda.yaml
2024-08-28 23:52:42    3654221 model.pkl
2024-08-28 23:52:42        129 python_env.yaml
2024-08-28 23:52:42         62 registered_model_meta
2024-08-28 23:52:42        274 requirements.txt

C:\WINDOWS\system32>aws s3 ls
2024-08-28 23:47:06 votre-nom-de-bucket

C:\WINDOWS\system32>aws s3 mb s3://mon-projet-ml --region eu-west-3
make_bucket: mon-projet-ml

C:\WINDOWS\system32>aws s3 cp s3://votre-nom-de-bucket/ s3://mon-projet-ml/ --recursive
copy: s3://votre-nom-de-bucket/model/conda.yaml to s3://mon-projet-ml/model/conda.yaml
copy: s3://votre-nom-de-bucket/model/python_env.yaml to s3://mon-projet-ml/model/python_env.yaml
copy: s3://votre-nom-de-bucket/model/MLmodel to s3://mon-projet-ml/model/MLmodel
copy: s3://votre-nom-de-bucket/model/registered_model_meta to s3://mon-projet-ml/model/registered_model_meta
copy: s3://votre-nom-de-bucket/model/requirements.txt to s3://mon-projet-ml/model/requirements.txt
copy: s3://votre-nom-de-bucket/model/model.pkl to s3://mon-projet-ml/model/model.pkl

C:\WINDOWS\system32>aws s3 ls s3://mon-projet-ml/
                           PRE model/

C:\WINDOWS\system32>aws s3 rb s3://votre-nom-de-bucket --force
delete: s3://votre-nom-de-bucket/model/conda.yaml
delete: s3://votre-nom-de-bucket/model/python_env.yaml
delete: s3://votre-nom-de-bucket/model/model.pkl
delete: s3://votre-nom-de-bucket/model/registered_model_meta
delete: s3://votre-nom-de-bucket/model/MLmodel
delete: s3://votre-nom-de-bucket/model/requirements.txt
remove_bucket: votre-nom-de-bucket

C:\WINDOWS\system32>aws s3 ls
2024-08-29 00:03:02 mon-projet-ml

C:\WINDOWS\system32>
```

Création d'une nouvelle application sur Heroku

```
C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/69ce3647-24e5-4272-ae61-bb8155b41946?requestor=SFMyNTY.g2gDbQAAAA05NC4yMzkuMTEuMTU0bgYACOAbm5EBYgABUYA.7YG8Ndbj5SmudtZmTkvpzh4ESDv4CHmQblxqTi-2i-U
Logging in... done
Logged in as angekhoty2@gmail.com

C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku>heroku create predictions-app-projet7
Creating ⬚ predictions-app-projet7... done
https://predictions-app-projet7-f3b0b3d90518.herokuapp.com/ | https://git.heroku.com/predictions-app-projet7.git

C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku>cd C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook

C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook>python server_distant.py
C:\Python39\lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator LabelEncoder from version 1.5.0 when using version 1.5.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'server_distant'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.85:5000
Press CTRL+C to quit
```

```
Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git init
Reinitialized existing Git repository in C:/Users/Infogene/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku/.git/

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git add mlflow_runs/

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git add server_distant.py

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git add Procfile

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git add requirements.txt

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git commit -m "Updated server_distant with S3 model loading"
[master e3c38c5] Updated server_distant with S3 model loading
 3 files changed, 16 insertions(+), 9 deletions(-)
```

HEROKU

git bash

```
Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git remote remove heroku

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git remote add heroku https://git.heroku.com/predictions-app-projet7.git

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ git push heroku master
Enumerating objects: 674, done.
Counting objects: 100% (674/674), done.
Delta compression using up to 4 threads
Compressing objects: 100% (372/372), done.
Writing objects: 100% (674/674), 1.24 MiB | 7.89 MiB/s, done.
Total 674 (delta 143), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (143/143), done.
remote: Updated 1253 paths from 79cb5f9
remote: Compressing source files... done.
remote: Building source:
remote:
```

```
Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ heroku logs --tail --app predictions-app-projet7
2024-08-28T23:04:26.576708+00:00 app[web.1]: ^^^^^^^^^^^^^^^^^
2024-08-28T23:04:26.576708+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.12/site-packages/botocore/signers.py", line 105, in handler
2024-08-28T23:04:26.576708+00:00 app[web.1]: return self.sign(operation_name, re
```

```
Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION
DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ heroku config:set AWS_ACCESS_KEY_ID=AKIAST6S7IRG2LV3ZVOI
Setting AWS_ACCESS_KEY_ID and restarting predictions-app-projet7... done, v4
AWS_ACCESS_KEY_ID: AKIAST6S7IRG2LV3ZVOI

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION
DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ heroku config:set AWS_SECRET_ACCESS_KEY=iE+f2eVmObIJshBODkGO65YtwxPeTvuLavaIxf
6O
Setting AWS_SECRET_ACCESS_KEY and restarting predictions-app-projet7... done, v5
AWS_SECRET_ACCESS_KEY: iE+f2eVmObIJshBODkGO65YtwxPeTvuLavaIxf6O

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION
DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ heroku restart --app predictions-app-projet7
Restarting dynos on predictions-app-projet7... done

Infogene@LAPTOP-2-Mohamed-Ali MINGW64 ~/Documents/Khoty_Privé/DOSSIER FORMATION
DATA SCIENTIST/PROJET 7 ML/API_Heroku (master)
$ heroku logs --tail --app predictions-app-projet7
2024-08-28T23:17:36.759956+00:00 app[web.1]: botocore.exceptions.PartialCredenti
alsError: Partial credentials found in env, missing: AWS_SECRET_ACCESS_KEY
2024-08-28T23:17:36.760056+00:00 app[web.1]: [2024-08-28 23:17:36 +0000] [9] [IN
FO] Worker exiting (pid: 9)
2024-08-28T23:17:36.842762+00:00 app[web.1]: [2024-08-28 23:17:36 +0000] [17] [E
```

git bash

Streamlit

```
C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\API_Heroku>cd C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook

C:\Users\Infogene\Documents\Khoty_Privé\DOSSIER FORMATION DATA SCIENTIST\PROJET 7 ML\Notebook>streamlit run dashboardStreamlit_Flask_distant_heroku.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.85:8501

<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
c:\python39\lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator LabelEncoder from version 1.5.0 when using version 1.5.1. This might lead to breaking code
or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
c:\python39\lib\site-packages\shap\plots\_waterfall.py:315: UserWarning:

FigureCanvasAgg is non-interactive, and thus cannot be shown

<IPython.core.display.HTML object>
c:\python39\lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning:

Trying to unpickle estimator LabelEncoder from version 1.5.0 when using version 1.5.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
```

❑ Conception d'une application Streamlit qui interagit avec l'API Flask : Interface utilisateur de l'application