

Juillet 2025

PROJET N° 7

Implémentez un modèle de scoring



Sabri ATTAL

Parcours :
Data Scientist

Sommaire

- 1. Contexte et objectifs du projet**
- 2. Exploration des données (EDA)**
- 3. Feature Engineering**
- 4. Stratégie de modélisation et gestion du déséquilibre**
- 5. Evaluation des modèles**
- 6. Optimisation du modèle retenu**
- 7. Optimisation du Seuil de Prediction**
- 8. Optimisation du modèle avec un Score Metier**
- 9. Feature Importance Global et Local (SHAP)**
- 10. API et mise en production CI/CD**
- 11. Workflow CI/CD**
- 12. Présentation du Dashbord**
- 13. Détection du Data Drift (Evidently)**
- 14. Conclusion**

1. Contexte et objectifs du projet

Contexte

L'entreprise Prêt à Dépenser propose des crédits à la consommation pour des personnes ayant peu ou pas d'historique bancaire.

Dans ce contexte, elle souhaite automatiser l'évaluation du risque client à l'aide d'un système de scoring de crédit basé sur des données internes et externes.

Objectif

Le projet consiste à développer un modèle de classification capable d'estimer la probabilité de défaut de remboursement d'un client.

Ce modèle doit être déployé dans un environnement de production complet, en suivant une démarche MLOps

Attentes MLOps

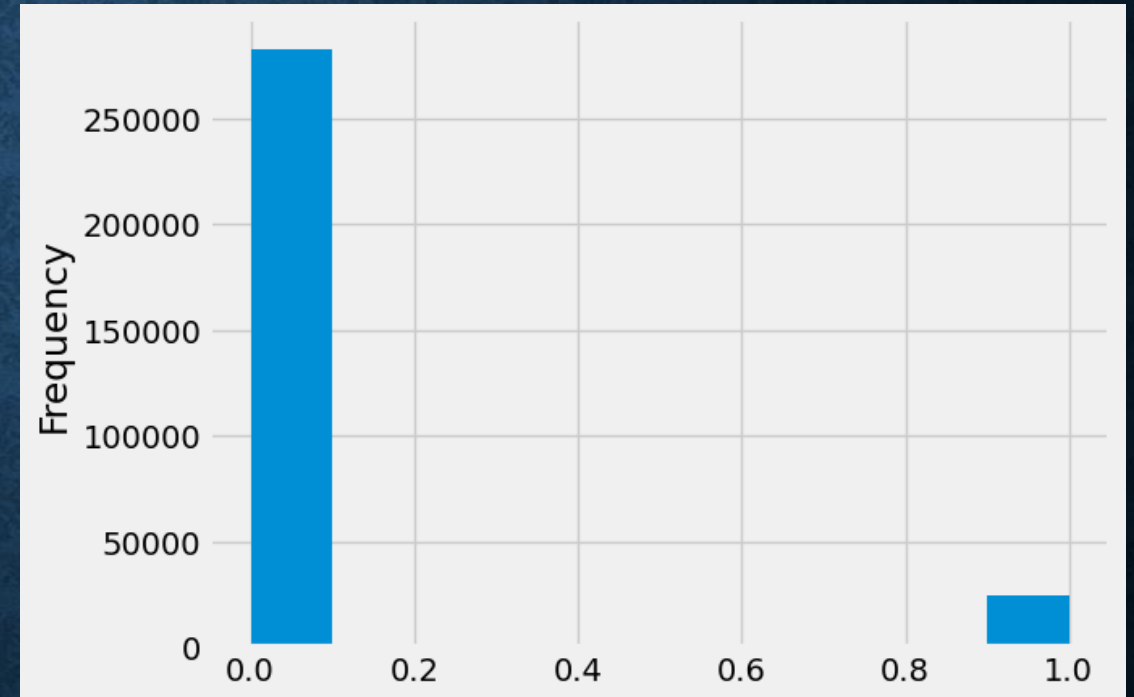
- Modèle ML optimisé
- Tracking Mlflow
- Versionning Git/Github
- API de prédiction
- Interface Streamlit
- CI/CD (GitHub Actions)
- Tests unitaires automatisés
- Détection de data drift



2. Exploration des données (EDA)

- Utilisation d'un notebook Kaggle de référence pour structurer l'EDA
- Analyse des variables : types, valeurs manquantes, outliers, corrélations

- 7 fichiers de données
- 307 000 clients
- 121 features
- Target (0 = bon, 1 = défaut)
- Dataset déséquilibré
 - 90% de bon client (0)
 - 10% de mauvais client (1)



3. Feature Engineering

- Utilisation d'un notebook Kaggle riche en feature engineering.
- Bureau + Bureau Balance :
 - Agrégation des crédits passés/actuels (min, max, mean, etc.).
 - Création de features séparées pour les crédits actifs/fermés.
- Previous Applications :
 - Agrégation des anciennes demandes de crédit par client.
 - Création de features pour les demandes approuvées/refusées.
- POS CASH balance :
 - Agrégation des crédits renouvelables (POS / Cash loans) par client.
 - Comptage du nombre total de comptes POS.
- Installments Payments :
 - Calcul des retards, des paiements anticipés et des ratios payé/dû.
 - Agrégation par client pour obtenir des statistiques globales.
- Credit Card Balance :
 - Agrégation des historiques mensuels des cartes de crédit.
 - Calcul du nombre total de lignes carte de crédit par client.



797 features

4. Stratégie de modélisation et gestion du déséquilibre

Déséquilibre **90/10** entre classes → Accuracy inadaptée

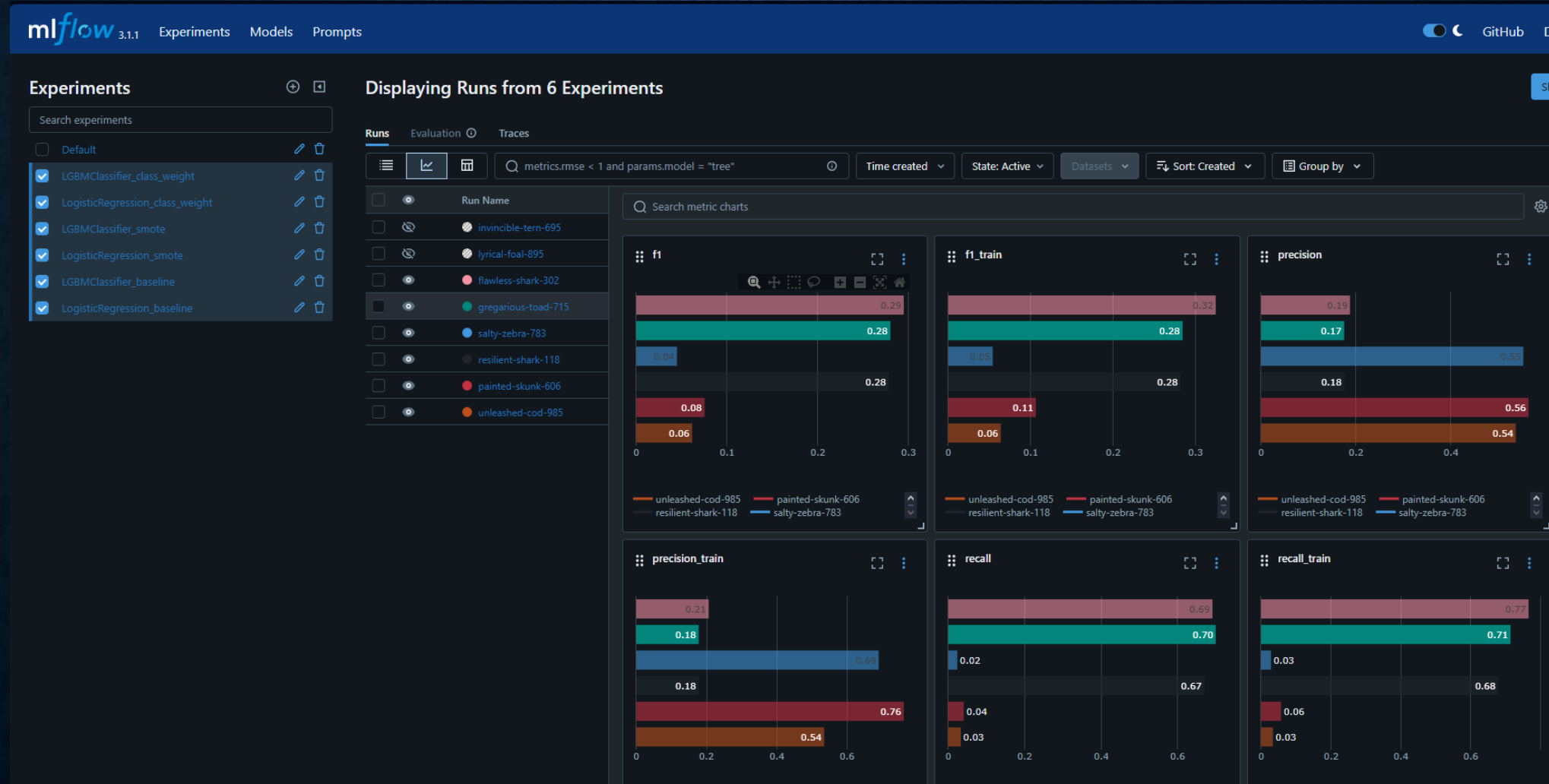
Deux modèles Linéaire/Non linéaire testés sous trois variantes chacun :

- Baseline
- SMOTE
- Class Weight

Modèle	Imputation NaN	StandarScaler	SMOTE	Class Weight
LogisticRegression_Baseline	■	■	×	×
LogisticRegression_SMOTE	■	■	■	×
LogisticRegression_ClassWeight	■	■	×	■
LightGBM_Baseline	×	■	×	×
LightGBM_SMOTE	■	■	■	×
LightGBM_ClassWeight	×	■	×	■

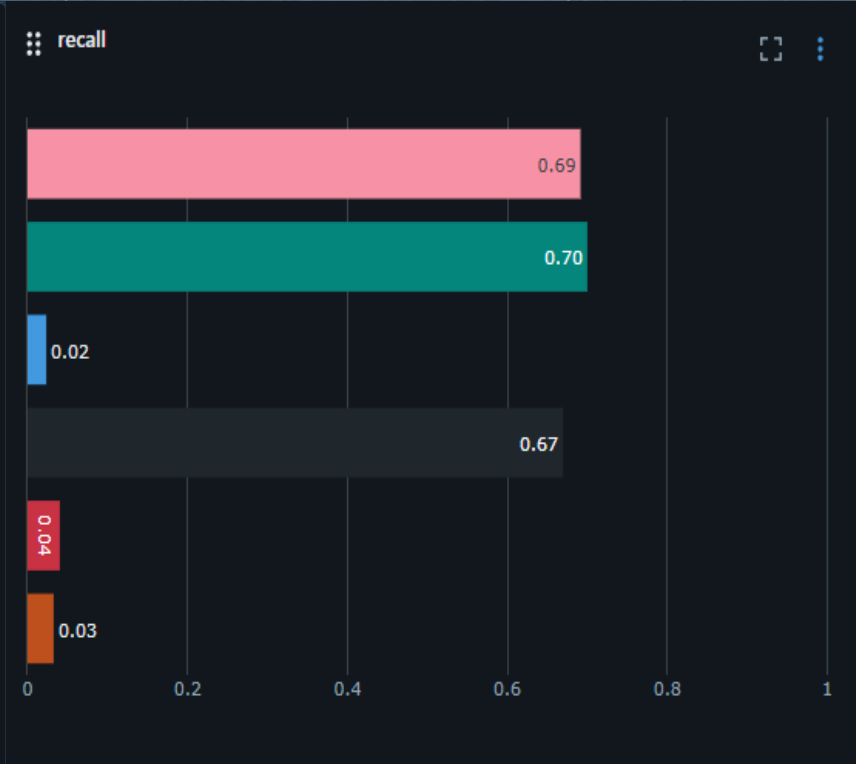
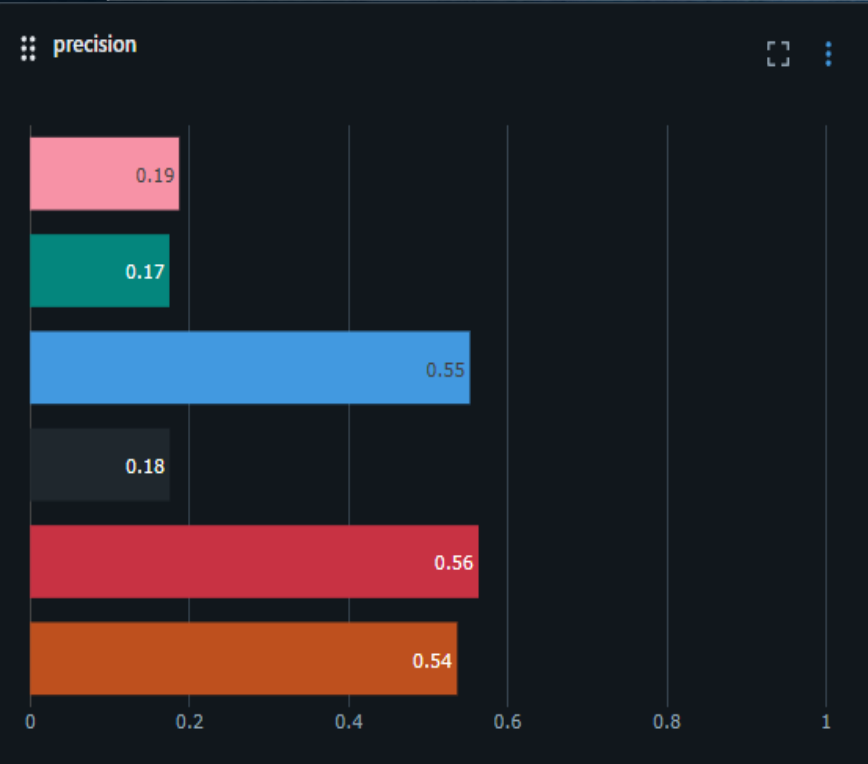
5. Evaluation des modèles

- Tracking des runs via MIFlow UI

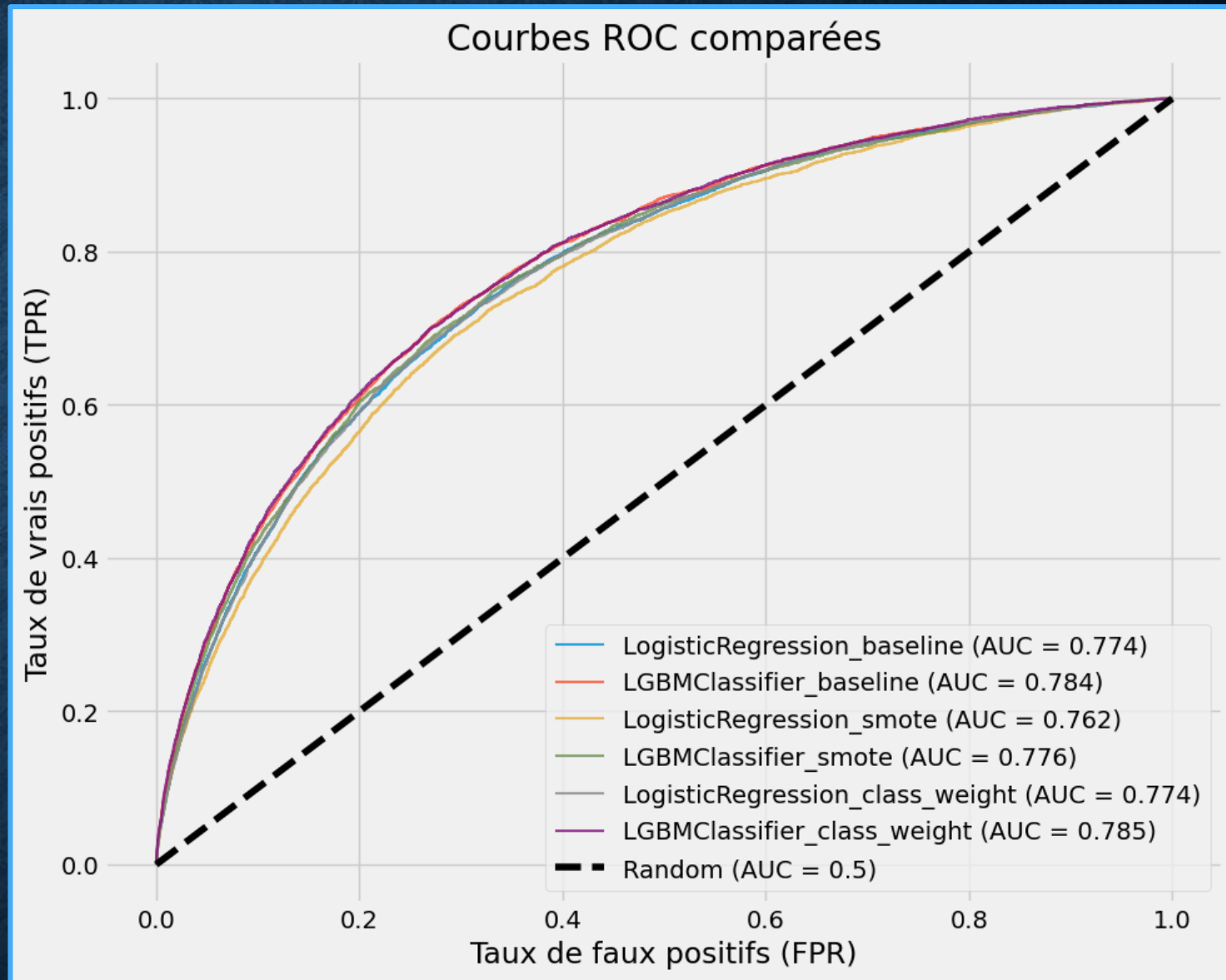


5. Evaluation des modèles

	Precision	Recall	f1
LightGBM_ClassWeight	0,19	0,69	0,29
LogisticRegression_ClassWeight	0,17	0,70	0,28
LightGBM_SMOTE	0,55	0,02	0,04
LogisticRegression_SMOTE	0,18	0,67	0,28
LightGBM_Baseline	0,56	0,04	0,08
LogisticRegression_Baseline	0,54	0,03	0,06



5. Evaluation des modèles

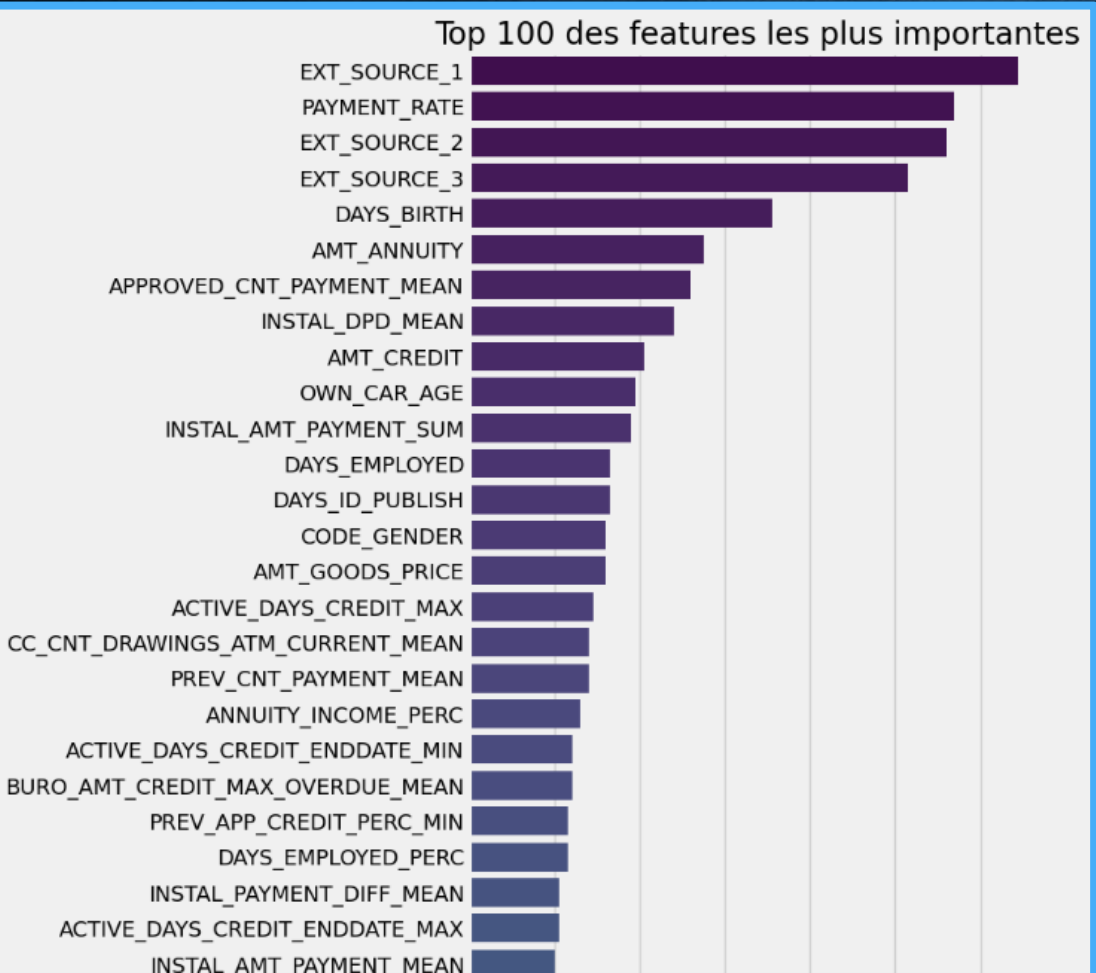



6. Optimisation du modèle retenu

Réduction du nombre de features à partir des importances



Optimisation des Hyperparamètres via GridSearch sur X_reduced



Metric de Scoring  ROC_AUC

	dataset	roc_auc	recall	precision	f1
0	train	0.847347	0.778499	0.216866	0.339232
1	test	0.786665	0.676737	0.187657	0.293835

7. Optimisation du Seuil de Prediction

• Modèle retenu ➡ **LightGBM_ClassWeight**

• Variantions du seuil de prediction [0,1]

• Calcul des metrics :

- Recall
- Precision
- F1 Score

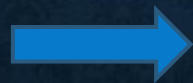
• Introduction d'une metric métier :

- $\text{cout_FP} = 10 \times \text{cout_FN}$

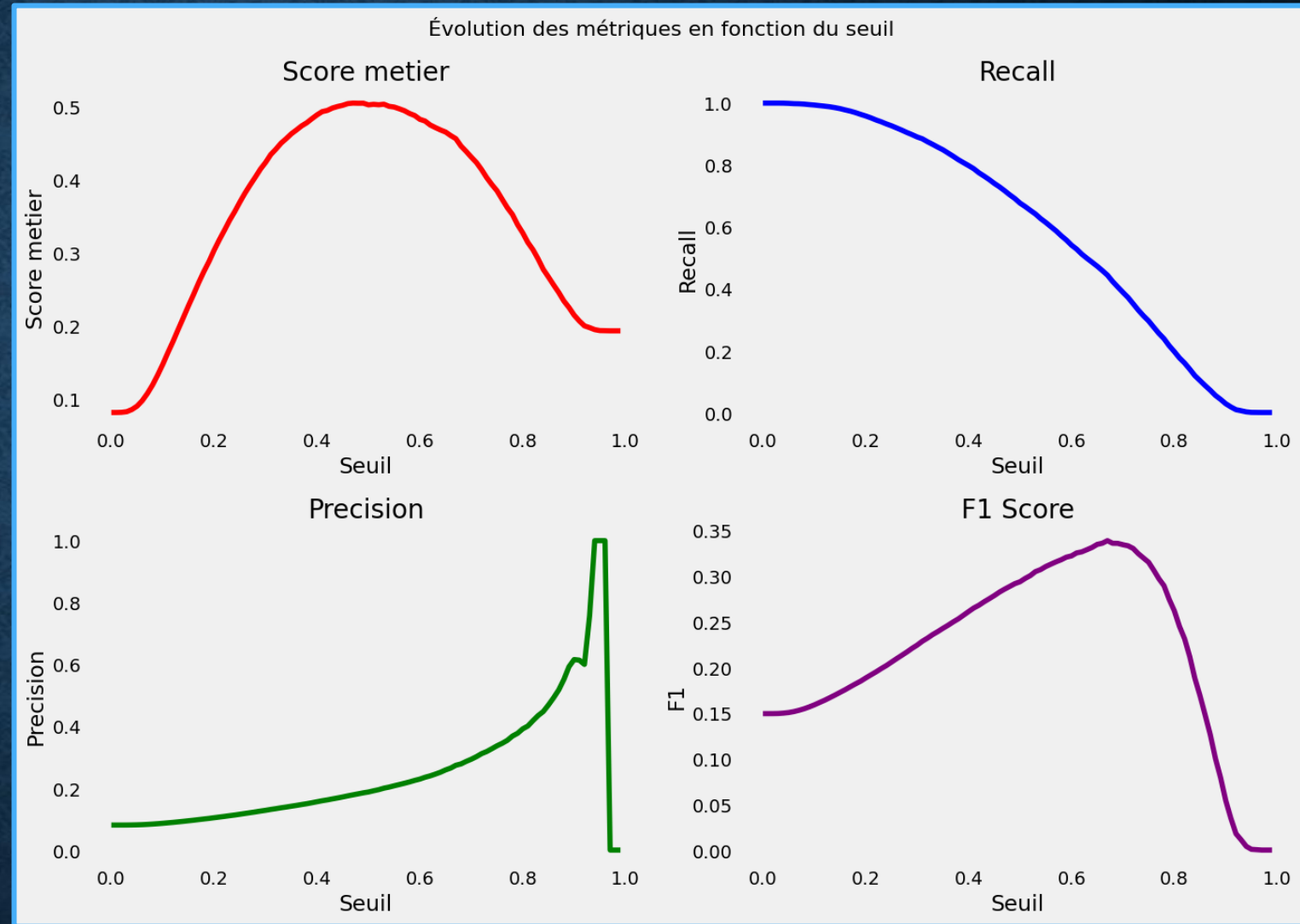


• $\text{Coût total} = 1 \times \text{FP} + 10 \times \text{FN}$

• $\text{Score métier} = 1 - \frac{\text{coût total}}{\text{nombre de clients}}$



• **Seuil optimal = 0,47**



8. Optimisation du modèle avec un Score Metier

- Creation d'une metric d'évaluation avec **make_scorer** basée sur le **score métier**
- Optimisation des Hyperparamètres via GridSearch

Metric de Scoring  **SCORE_METIER**

	dataset	roc_auc	recall	precision	f1	score_metier
0	train	0.878897	0.84864	0.226210	0.357205	0.643454
1	test	0.787130	0.69708	0.185457	0.292970	0.508292

9. Feature Importance Global et Local (SHAP)

Importance globale :

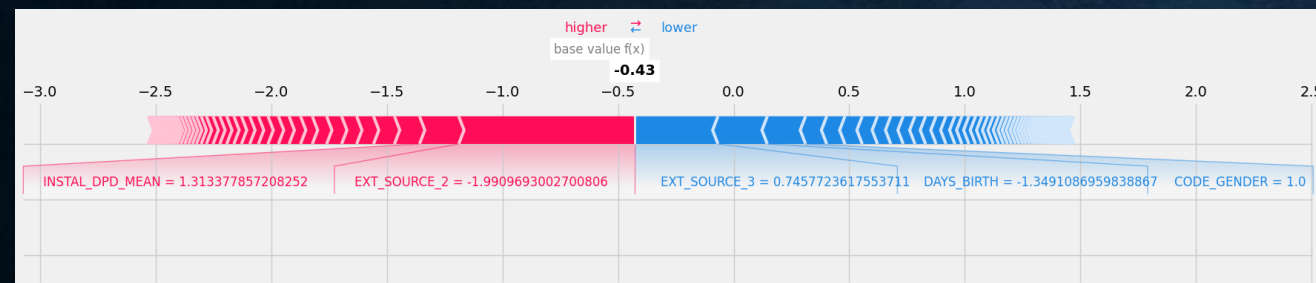
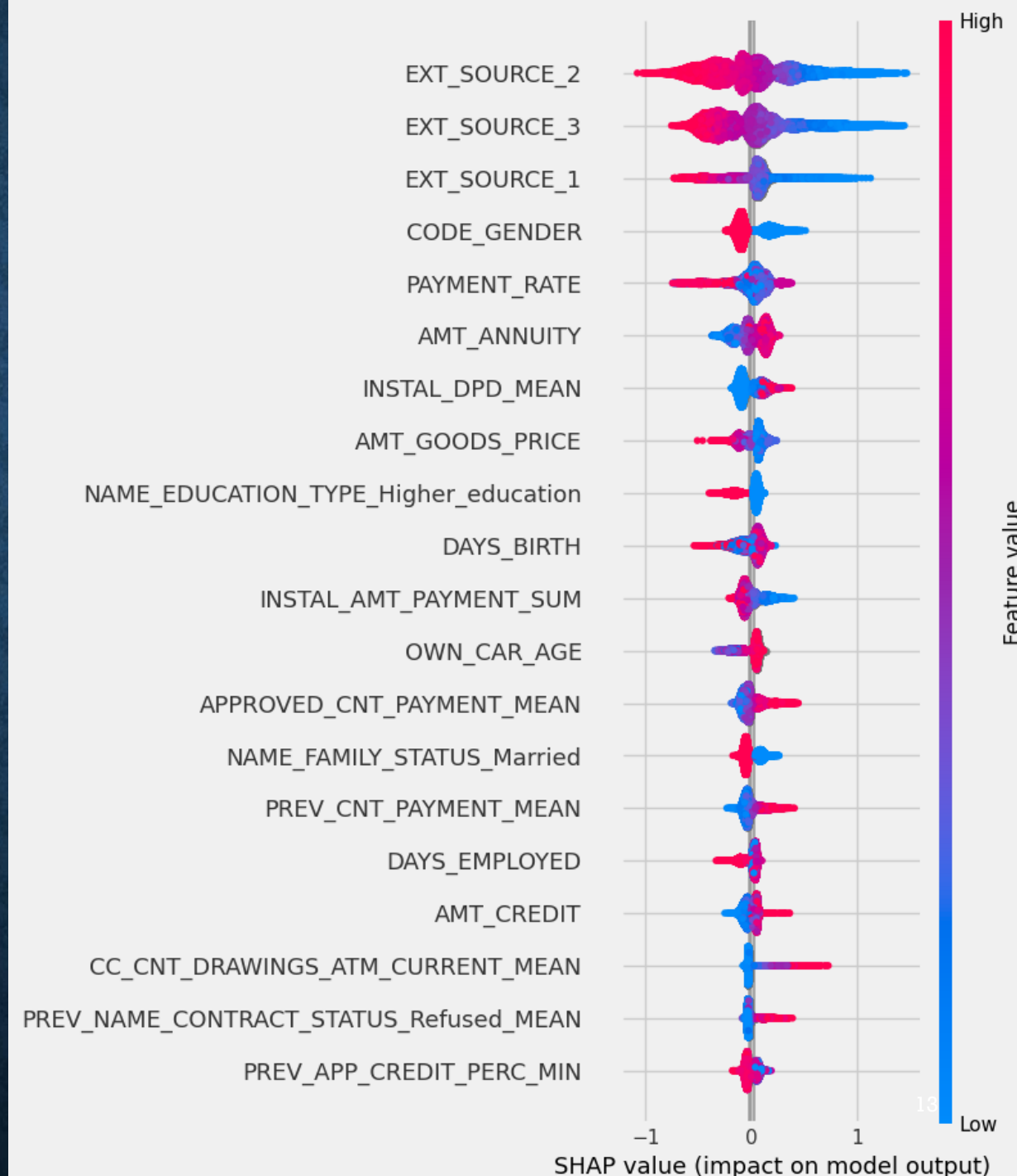
- Mesure l'impact moyen de chaque variable sur les prédictions du modèle
- Identifie les variables clés influençant la décision à l'échelle du Dataset
- Aide à comprendre les facteurs principaux derrière le score de crédit

Importance locale (explications individuelles) :

- Analyse comment chaque variable influence la prédiction pour un client spécifique
- Permet d'expliquer pourquoi un client est classé « bon » ou « mauvais payeur »
- Utile pour la transparence et la confiance auprès des décideurs et clients

Intérêts :

- Amélioration de l'interprétabilité du modèle
- Support à la prise de décision métier
- Facilite la détection des biais ou variables non pertinentes

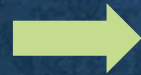


10. API et mise en production CI/CD

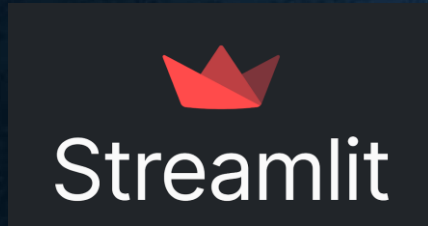
- Outils utilisés :



Versioning du code



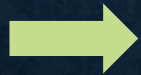
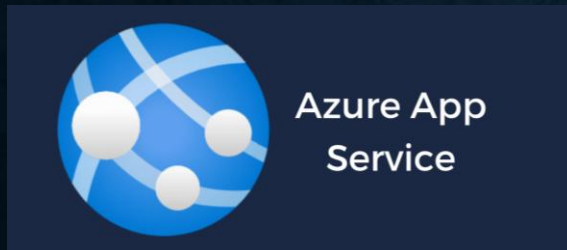
Création de l'API



Dashboard pour les résultats



Automatisation du déploiement et des testes unitaires



Hébergement de l'API en ligne

11. Workflow CI/CD

- Déclenchement automatique du Workflow CI/CD ➡ **Push sur Main**

lint-format

- Vérifie la qualité du code
- Corrige le formatage, indentation, etc.

unit-tests

- Exécute les tests prévus
- Stoppe si une erreur survient (maxfail)

build

- Prépare les fichiers nécessaires pour le déploiement
- Génère un artefact partagé entre les jobs

deploy

- Récupère l'artefact généré lors du build
- Déploie l'application sur Azure App Service

main_projet7-credit-default-risk.yml

on: push

✔ lint-format

6s

✔ unit-tests

1m 9s

✔ build

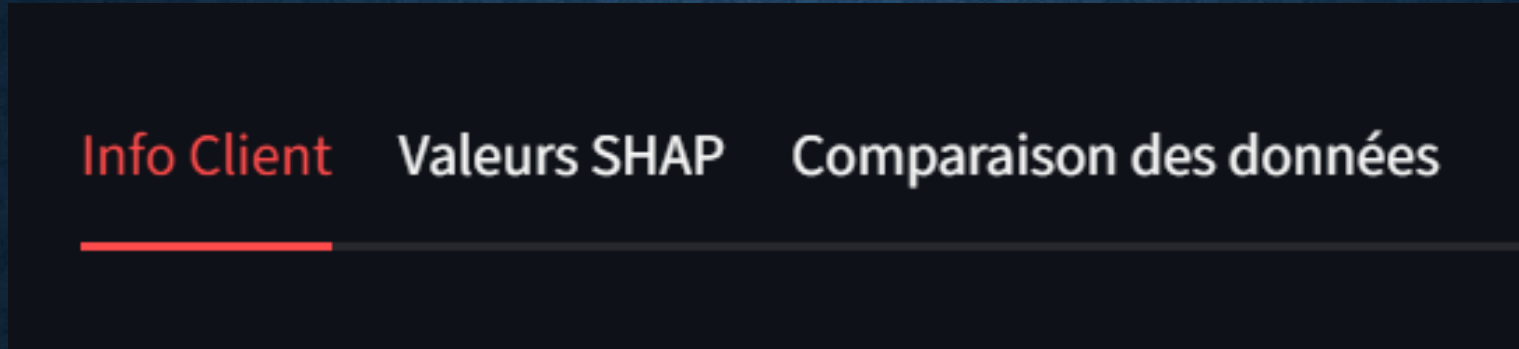
6s

✔ deploy

7m 56s

12. Fonctionnalités clés du Dashboard

Mise en place de 3 onglets de fonctionnalités :



Onglet 1 – Info Client

- Sélection d'un client via son identifiant
- Affichage des informations générales
- Prédiction du score avec visualisation
- la modification des informations client

Onglet 2 – Valeurs SHAP

La contribution des features via :

- Features importance Local
- Features importance Global

Onglet 3 – Comparaison

Comparaison du profil client via :

- Analyse univariée
- Analyse Bivariée
- Filtre de comparaison

13. Présentation détaillée du Dashbord

Onglet 1 – Info Client

Info Client Valeurs SHAP Comparaison des données

Aperçu des données :

	SK_ID_CURR	EXT_SOURCE_1	PAYMENT_RATE	EXT_SOURCE_2	EXT_SOURCE_3	DAY
11	360159	None	0.0266	0.4128	0.5065	
12	296995	None	0.0943	None	0.8537	
13	417896	0.3102	0.0324	0.5171	None	
14	326343	None	0.0543	0.6765	0.6024	
15	387431	None	0.05	0.6985	0.5011	

Selectionner un client via son SK_ID_CURR :

180994

Information du client sélectionné :

Reset

SK_ID_CURR	EXT_SOURCE_1	PAYMENT_RATE	EXT_SOURCE_2	EXT_SOURCE_3
180994	0.6977	0.0527	0.5976	

ID

180994

Sexe

Femme

Âge

29 ans

Prédire

Manage app

Prédire - Prédiction du score client via :
une route **/predict**

Résultats de la prédiction:

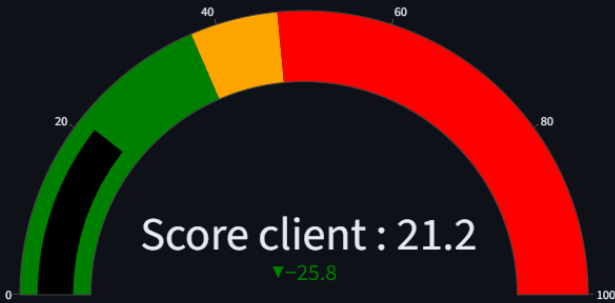
- **Jauge** de Visualisation (Plotly)
- Message d'information coloré (**risque faible**, **à surveiller**, **risque élevé**).



- Modification des infos client via :
tableau des features

SK_ID_CURR	EXT_SOURCE_1	PAYMENT_RATE	EXT_SOURCE_2	EXT_SOURCE_3
276712	0.319739	0.0289	0.2813	0.151

Reset - Annulation des modifications via :
Bouton « **Reset** »



Score client : 21 / 100 — Risque faible (seuil = 47)



Score client : 43 / 100 — À surveiller (seuil = 47)

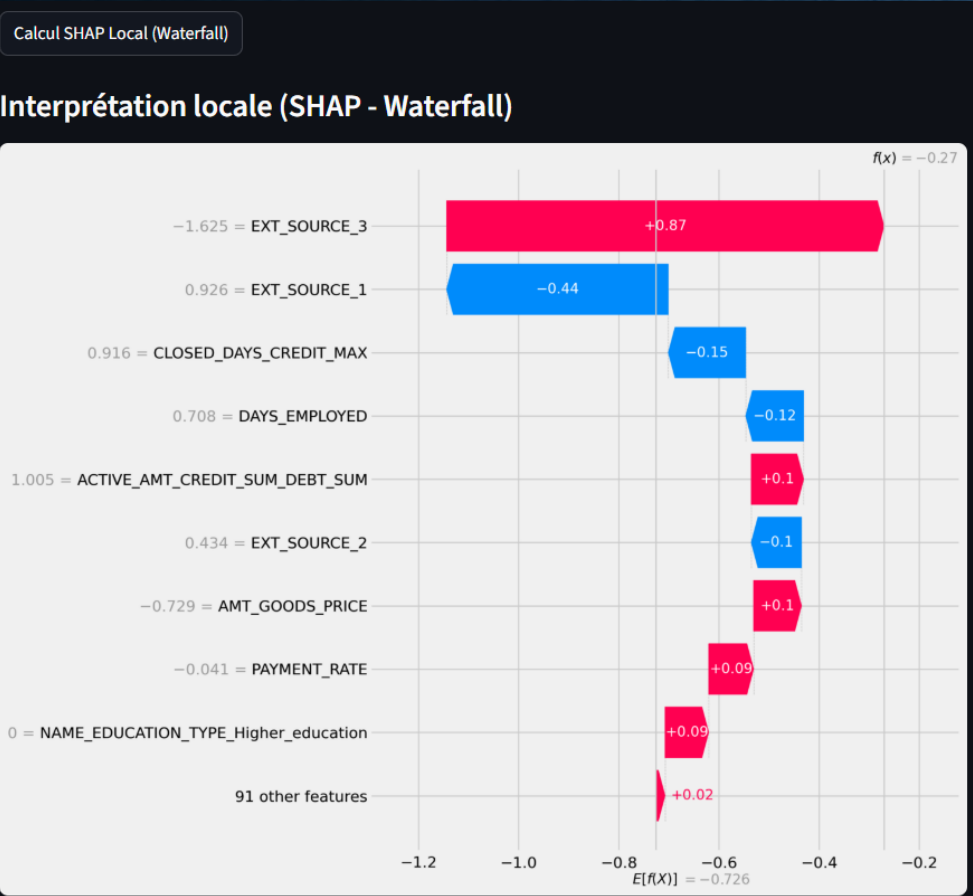


Score client : 62 / 100 — Risque élevé (seuil = 47)

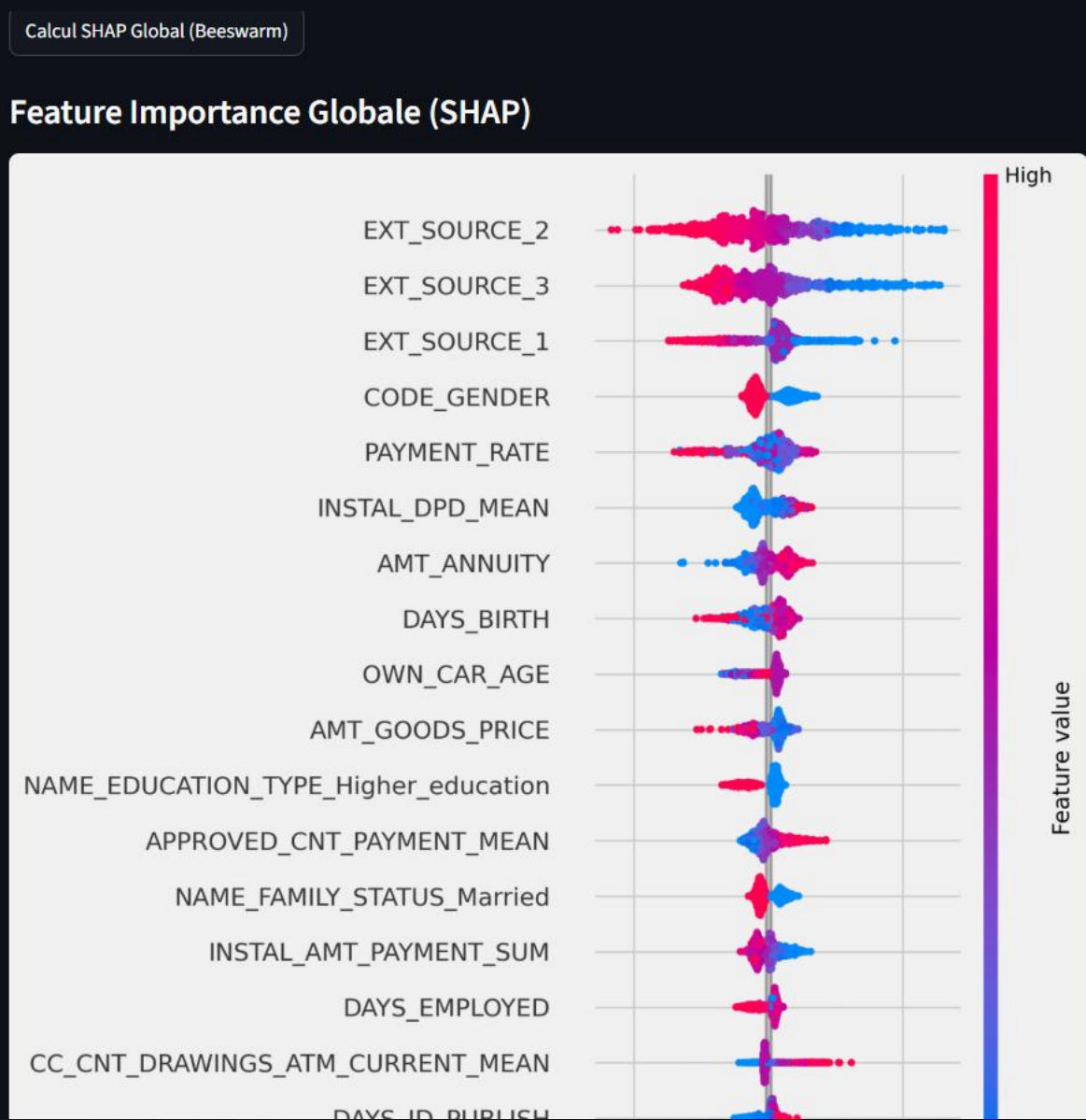
13. Présentation détaillée du Dashboard

Onglet 2 – Valeurs SHAP

Calcul des features importances Local (Waterfall) via :
une route **/shap_local**



Calcul des features importances Global (Beeswarm) via :
une route **/shap_global**



13. Présentation détaillée du Dashboard

Onglet 3 – Comparaison

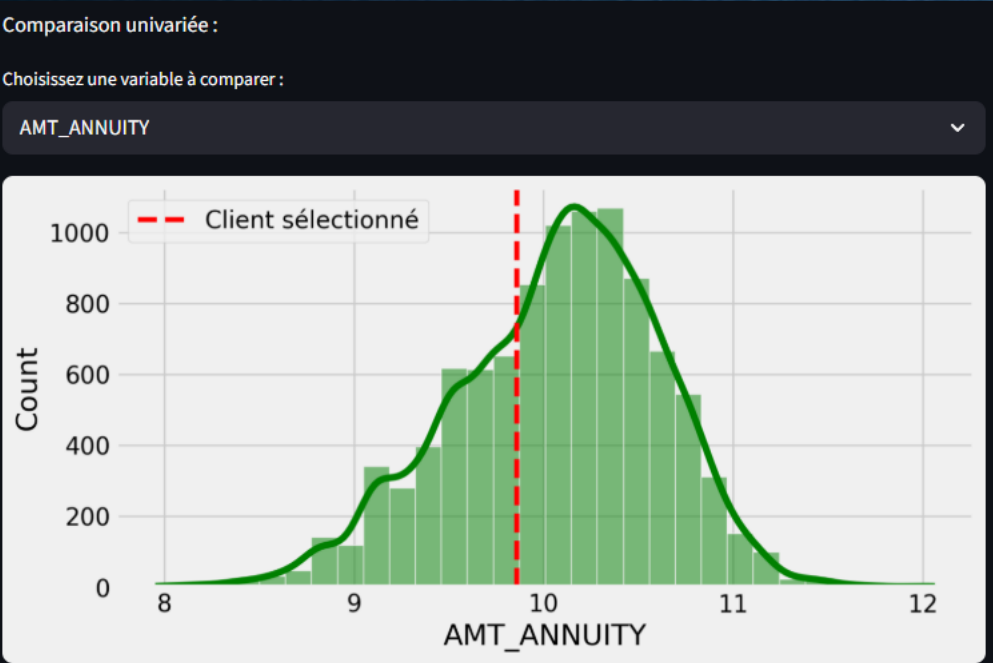
Comparaison du profil du client à :

- Toute la population.
- Un sous-groupe de même sexe.
- Un sous-groupe de même tranche d'âge.
- Une combinaison des deux).

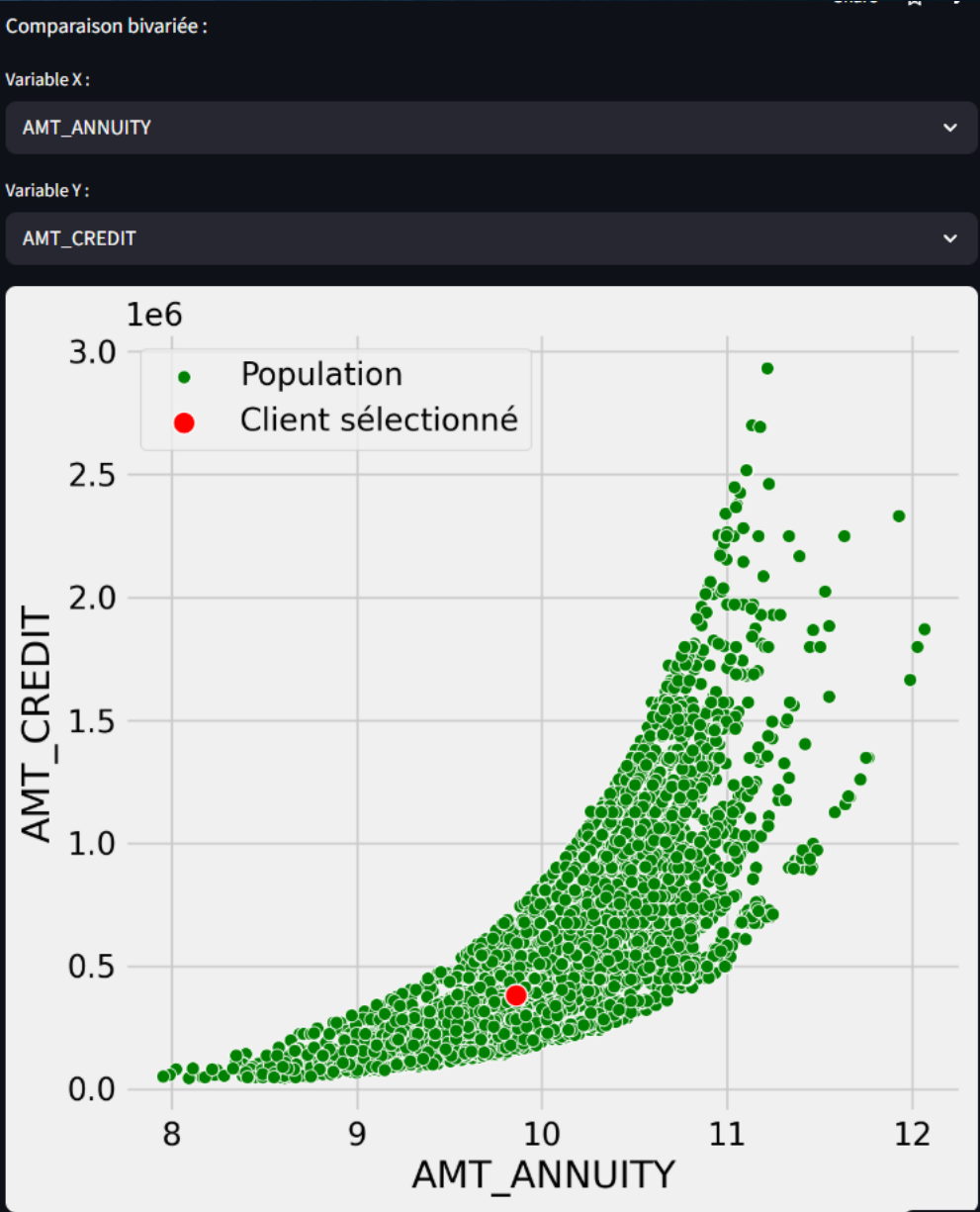
Filtrer les clients similaires :

- ☒ Vue globale
- ☐ Même sexe
- ☐ Même tranche d'âge
- ☐ Même sexe et tranche d'âge

Analyse univariée : histogramme avec la valeur du client en surbrillance.



Analyse bivariée :
Nuage de points avec le client mis en évidence.



14. Détection du Data Drift (Evidently)

Le **data drift** désigne une modification dans la distribution des données entre l'entraînement du modèle et sa mise en production.

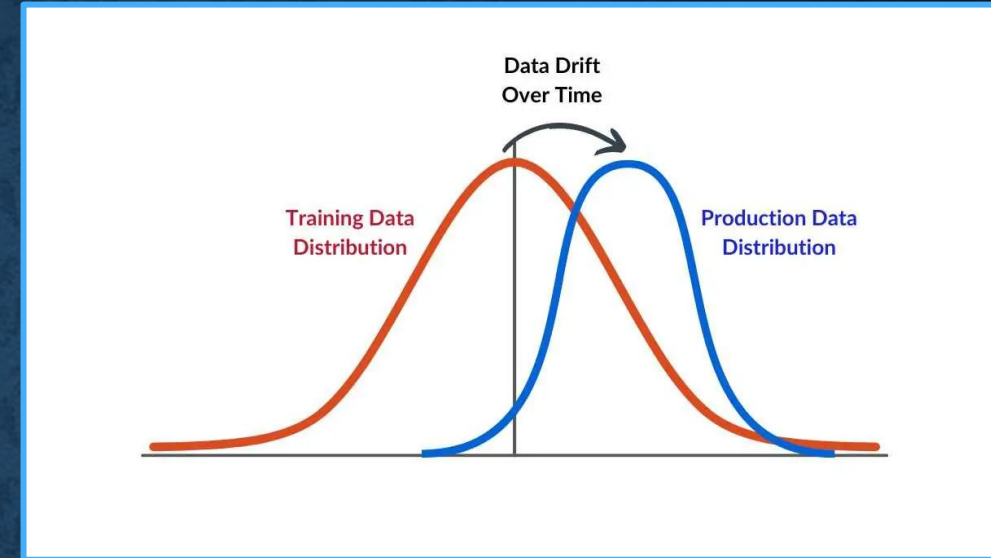
Cela peut affecter la qualité des prédictions si le modèle n'est plus confronté aux mêmes types de données qu'au moment de son apprentissage.

Méthodologie :

- Utilisation de la bibliothèque Evidently
- Comparaison entre le jeu de données d'entraînement (app_train) et les données récentes (app_test).

Evidently utilise des tests statistiques et des distances pour détecter le drift :

- **Variables numériques** : distance de Wasserstein et test de Kolmogorov-Smirnov
- **Variables catégorielles** : distance de Jensen-Shannon et test du Chi²



14. Détection du Data Drift (Evidently)

- ✓ Pas de drift global détecté (seuil 0,5 non dépassé).
- ⚠ Drift détecté sur 9 variables (≈ 7,4 % des colonnes).

Dataset Drift

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

121
Columns

9
Drifted Columns

0.0744
Share of Drifted Columns

Data Drift Summary

Drift is detected for 7.438% of columns (9 out of 121).

Search

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> AMT_REQ_CREDIT_BUREAU_QRT	num			Detected	Wasserstein distance (normed)	0.359052
> AMT_REQ_CREDIT_BUREAU_MON	num			Detected	Wasserstein distance (normed)	0.281765
> AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210785
> AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334
> AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161102
> AMT_REQ_CREDIT_BUREAU_YEAR	num			Detected	Wasserstein distance (normed)	0.151135

15. Conclusion du projet

Résultats clés :

- Modèle LightGBM optimisé avec AUC de 0.787 en production
- Gestion du déséquilibre via class weighting (recall à 69%)
- Sélection des 100 variables les plus prédictives
- Seuil optimal à 0.47 basé sur le coût métier ($FP=10 \times FN$)

Mise en production :

- Architecture MLOps complète : API FastAPI + Dashboard Streamlit
- Pipeline CI/CD automatisé (tests unitaires, déploiement Azure)
- Surveillance active : 9 variables en drift détectées (7.4%)