Projet 7:
Implémentez un modèle de scoring

Islem HABIBI
Parcours Data Scientist

Prêt à dépenser

OPENCLASSROOMS

Contexte

- Evaluation du risque de crédit
- Identifier des clients susceptibles de faire faillite.

- Minimisation des pertes financières
- Optimisation de la stratégie d'accord des prêts

Besoin: Construire des modèles prédictifs sophistiqués pour évaluer ce risque de manière automatique et précise

Problématique

Prédiction de la faillite → Analyse approfondie des données → Modèles prédictifs robustes

- **Résultat interprétable par les conseillers de l'agence**
- **mise à disposition du modèle dans un dashboard interactif**

Objectifs

- Construire un modèle de scoring prédictif pour évaluer la probabilité de faillite des clients.
- Déployer l'application dashboard et l'API sur une plateforme Cloud gratuite.
- Développer un dashboard interactif qui permet aux gestionnaires de la relation client d'interpréter les prédictions du modèle et d'améliorer leur connaissance client.

Traitements des données

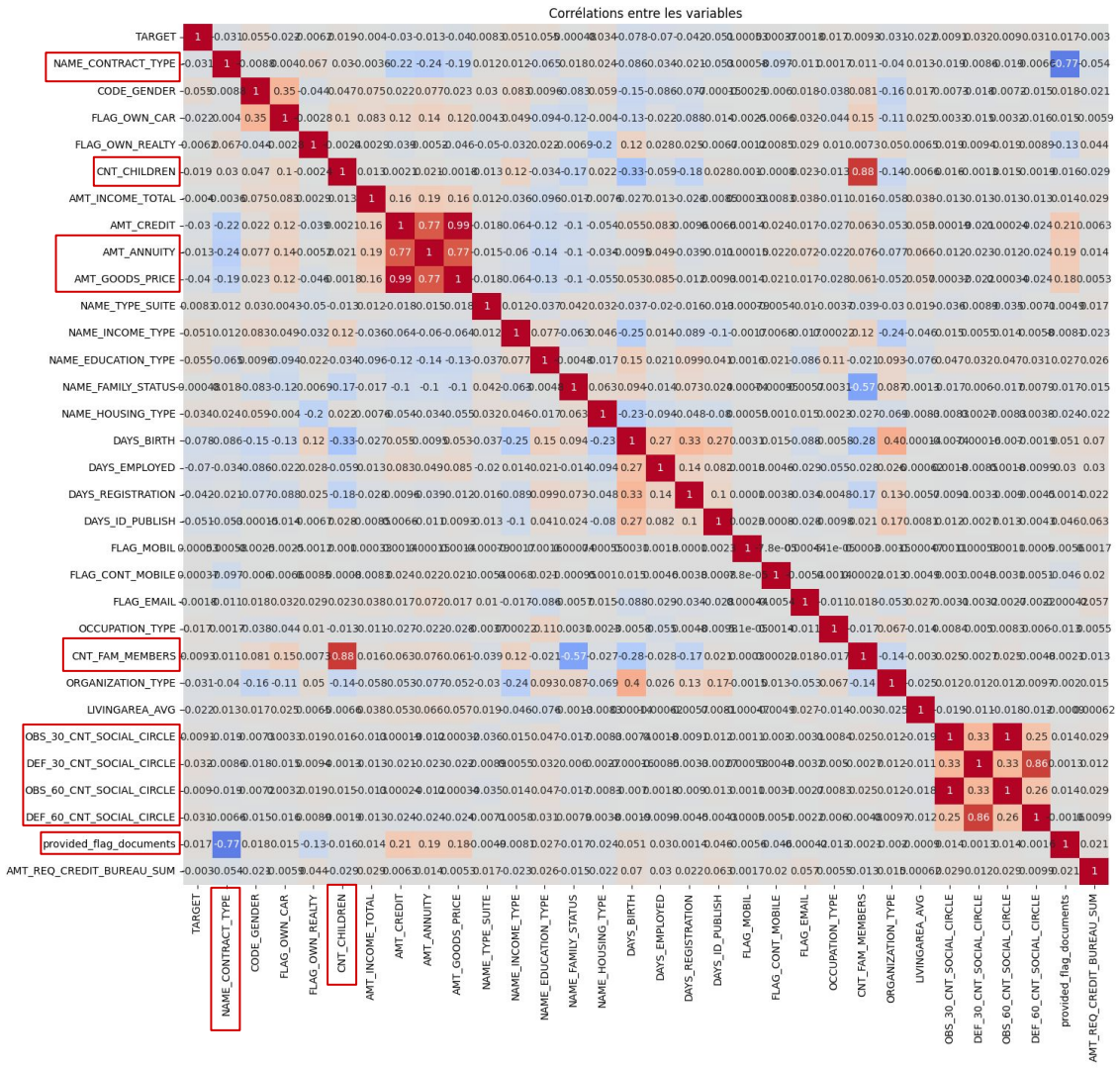
- Fusion des bases "application_train" et "application_test" pour nettoyage.

- Utilisation de "HomeCredit_columns_description" pour la compréhension et sélection des variables.

- Suppression des variables non pertinentes et remplacement par des variables généralistes.

Nettoyage des données

- Transformation des valeurs négatives en valeurs absolues pour certaines variables clés.
- Élimination des outliers et des variables corrélées.
- Séparation des données en ensembles d'entraînement et de test.



Encodage et Normalisation

- Utilisation de la fonction data_processing
  - Encodage des variables catégorielles avec LabelEncoder et OneHotEncoder
  - normalisation des données avec MinMaxScaler

- Préparation des données de test pour utilisation ultérieure.

Entraînement des modèles

- Échantillonnage de 1500 individus de chaque classe cible pour maintenir l'équilibre.
- Cross-validation et optimisation des hyperparamètres avec GridSearchCV.
- Algorithmes testés:
  - Régression Logistique
  - XGBoost
  - K-Neighbors
  - Decision Tree
- Sélection des meilleurs modèles selon diverses métriques de scoring.

Testing des modèles

1

Evaluation des performances des modèles avec `Model_testing`

2

Sélection du meilleur hyperparamètre

1. Le score métier
2. Le recall (sensibilité)
3. La spécificité
4. Le ROC AUC Score
5. L'accuracy (exactitude)
6. Le score de précision

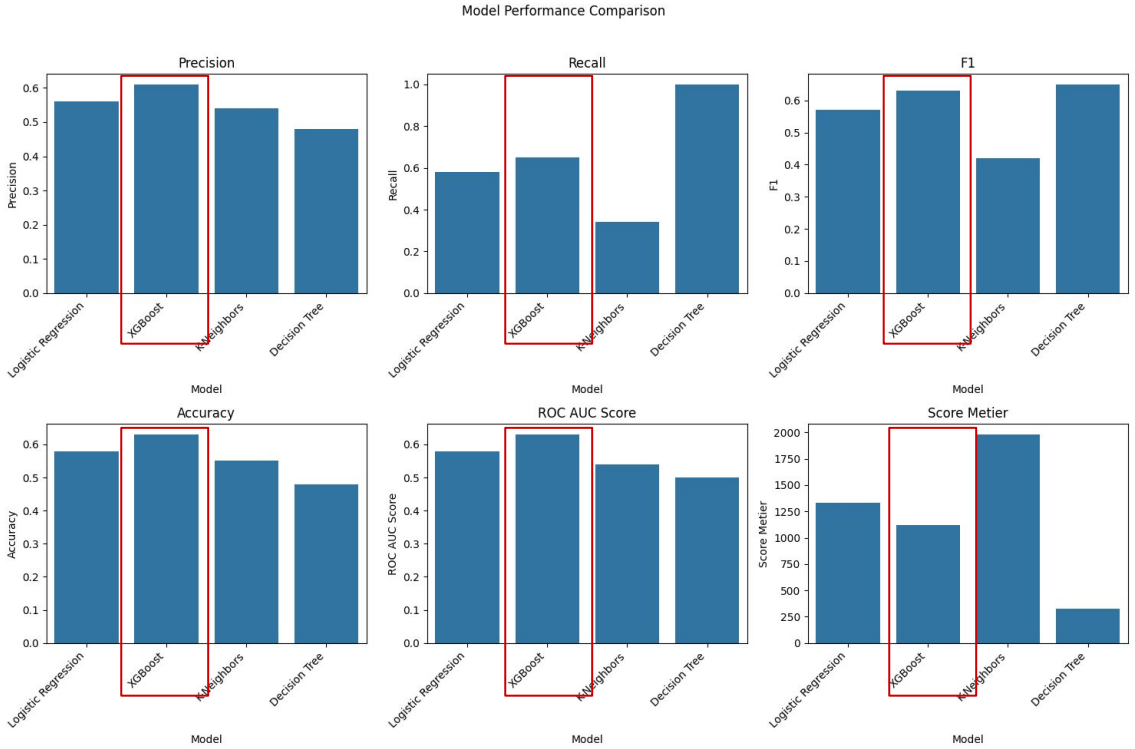
	Model	Scoring Method	Best Parameters	Best Estimator	Best Score
0	Logistic Regression	Accuracy	{'C': 10, 'max_iter': 200, 'penalty': 'l2'}	LogisticRegression(C=10, max_iter=200)	0.589583
1	Logistic Regression	Precision	{'C': 10, 'max_iter': 200, 'penalty': 'l2'}	LogisticRegression(C=10, max_iter=200)	0.592035
2	Logistic Regression	Recall	{'C': 175, 'max_iter': 200, 'penalty': 'l2'}	LogisticRegression(C=175, max_iter=200)	0.613359
3	Logistic Regression	F1	{'C': 175, 'max_iter': 200, 'penalty': 'l2'}	LogisticRegression(C=175, max_iter=200)	0.600706
4	Logistic Regression	ROC AUC	{'C': 10, 'max_iter': 200, 'penalty': 'l2'}	LogisticRegression(C=10, max_iter=200)	0.589402
5	XGBoost	Accuracy	{'eta': 0.2, 'learning_rate': 0.1, 'max_depth'...	XGBClassifier(base_score=None, booster=None, c...	0.613333
6	XGBoost	Precision	{'eta': 0.2, 'learning_rate': 0.01, 'max_depth...	XGBClassifier(base_score=None, booster=None, c...	0.792308
7	XGBoost	Recall	{'eta': 0.2, 'learning_rate': 0.2, 'max_depth'...	XGBClassifier(base_score=None, booster=None, c...	0.645529
8	XGBoost	F1	{'eta': 0.2, 'learning_rate': 0.2, 'max_depth'...	XGBClassifier(base_score=None, booster=None, c...	0.627743
9	XGBoost	ROC AUC	{'eta': 0.2, 'learning_rate': 0.1, 'max_depth'...	XGBClassifier(base_score=None, booster=None, c...	0.613075
10	K-Neighbors	Accuracy	{'algorithm': 'auto', 'n_neighbors': 8, 'weigh...	KNeighborsClassifier(n_neighbors=8, weights='d...	0.558750
11	K-Neighbors	Precision	{'algorithm': 'auto', 'n_neighbors': 2, 'weigh...	KNeighborsClassifier(n_neighbors=2)	0.592859
12	K-Neighbors	Recall	{'algorithm': 'auto', 'n_neighbors': 9, 'weigh...	KNeighborsClassifier(n_neighbors=9)	0.608482
13	K-Neighbors	F1	{'algorithm': 'auto', 'n_neighbors': 8, 'weigh...	KNeighborsClassifier(n_neighbors=8, weights='d...	0.581713
14	K-Neighbors	ROC AUC	{'algorithm': 'auto', 'n_neighbors': 8, 'weigh...	KNeighborsClassifier(n_neighbors=8, weights='d...	0.558253
15	Decision Tree	Accuracy	{'criterion': 'gini', 'max_depth': 10, 'max_fe...	DecisionTreeClassifier(max_depth=10, max_featu...	0.585833
16	Decision Tree	Precision	{'criterion': 'gini', 'max_depth': 3, 'max_fea...	DecisionTreeClassifier(max_depth=3, max_featur...	0.608544
17	Decision Tree	Recall	{'criterion': 'gini', 'max_depth': 3, 'max_fea...	DecisionTreeClassifier(max_depth=3, max_featur...	0.867184
18	Decision Tree	F1	{'criterion': 'gini', 'max_depth': 3, 'max_fea...	DecisionTreeClassifier(max_depth=3, max_featur...	0.649018
19	Decision Tree	ROC AUC	{'criterion': 'gini', 'max_depth': 5, 'max_fea...	DecisionTreeClassifier(max_depth=5, max_featur...	0.582096

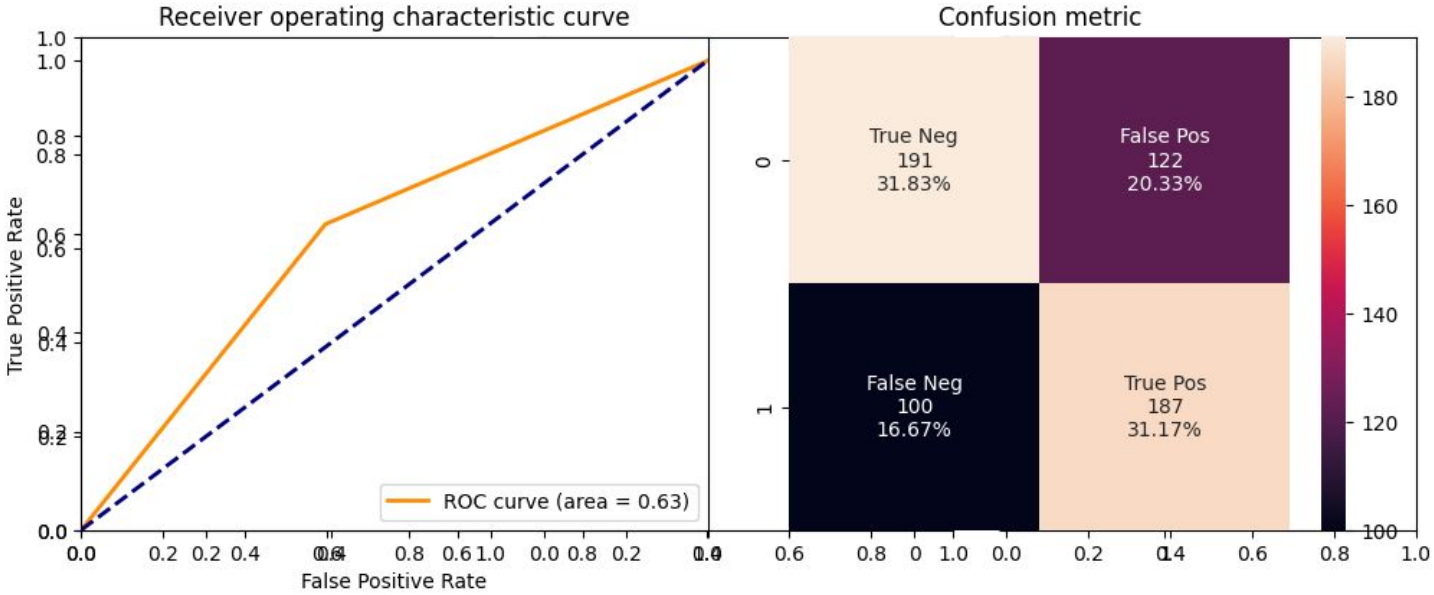
Optimisation et sélection du modèle final

- Test de différents seuils pour optimiser le seuil de classification.
- La fonction `best_threshold` pour identifier le meilleur seuil

Évaluation des performances des modèles avec diverses métriques

Sélection du modèle final





Precision	Recall	F1	Accuracy	ROC AUC Score	Score Metier
0.61	0.65	0.63	0.63	0.63	1122

Analyse du Data Drift

- Utilisation de la bibliothèque Evidently pour détecter le drift dans les données.

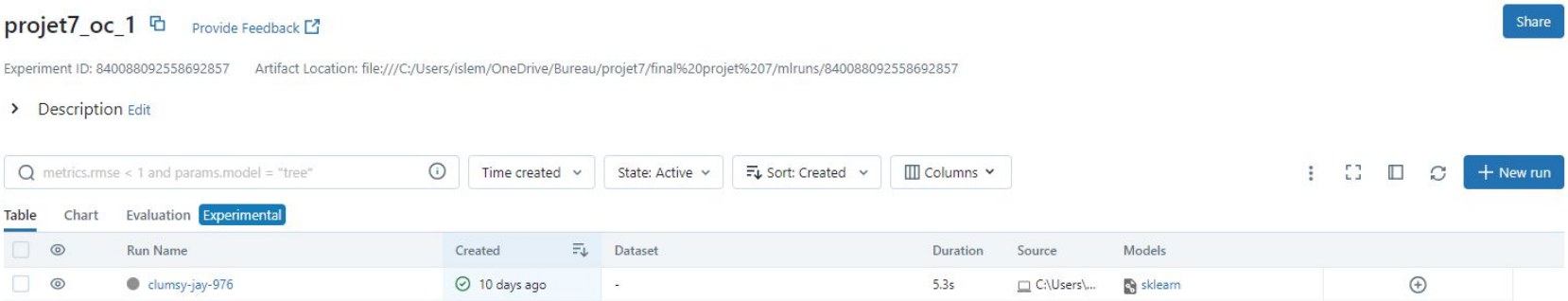
- Une dérive de données a été détectée dans 28,571 % des colonnes (8 sur 28).

Déploiement

- Enregistrement du modèle final avec MLflow.

- Déploiement local sous forme d'API Flask.

- Présentation des fonctionnalités de l'API, y compris les valeurs de Shapley et les graphiques SHAP.



Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/7faa0e8554a24261a5cce0b499c4026c/model'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

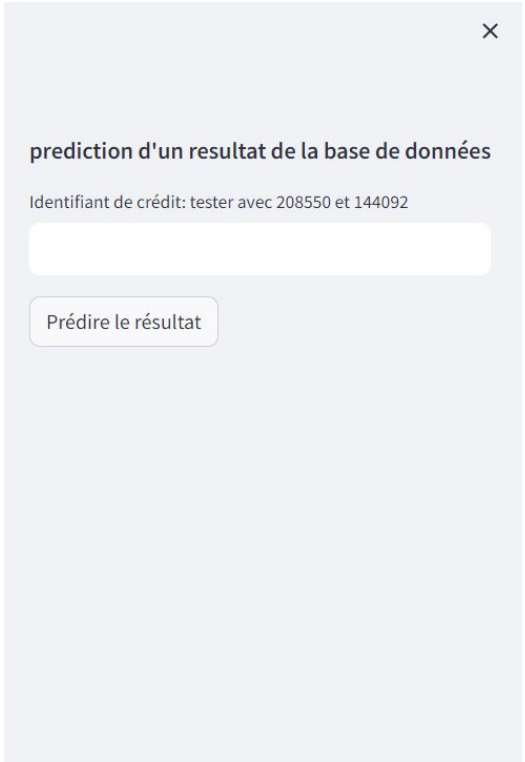
Predict on a Pandas DataFrame:

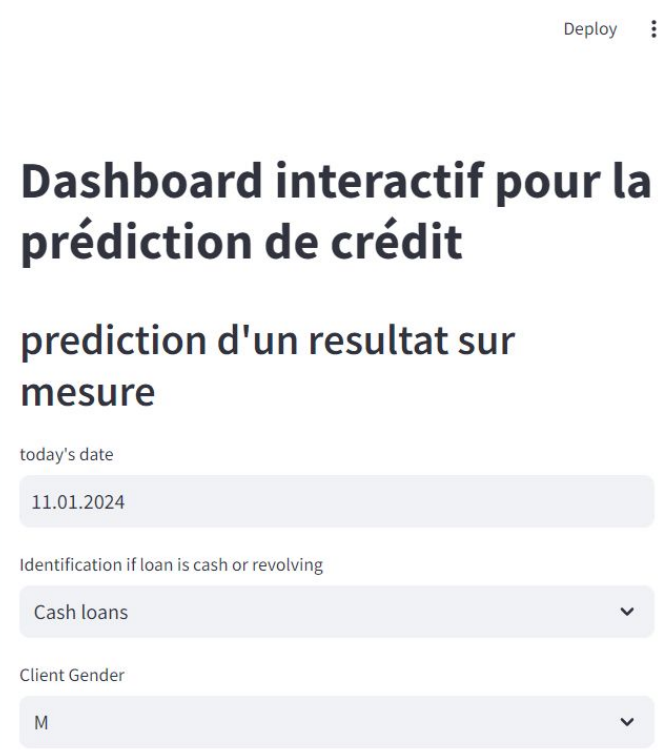
```
import mlflow
logged_model = 'runs:/7faa0e8554a24261a5cce0b499c4026c/model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
loaded_model.predict(pd.DataFrame(data))
```

Dashboard Interactif

- Développement d'un tableau de bord avec Streamlit pour l'utilisation de l'API de prédiction de crédit.

- Présentation des fonctionnalités interactives pour les utilisateurs.





prediction d'un resultat de la base de données

Identifiant de crédit: tester avec 208550 et 144092

208550

Prédire le résultat

Il y a une probabilité de 42.93 % que le créditeur puisse rencontrer des difficultés de paiement. La demande de crédit peut être approuvée.

prediction d'un resultat de la base de données

Identifiant de crédit: tester avec 208550 et 144092

144092

Prédire le résultat

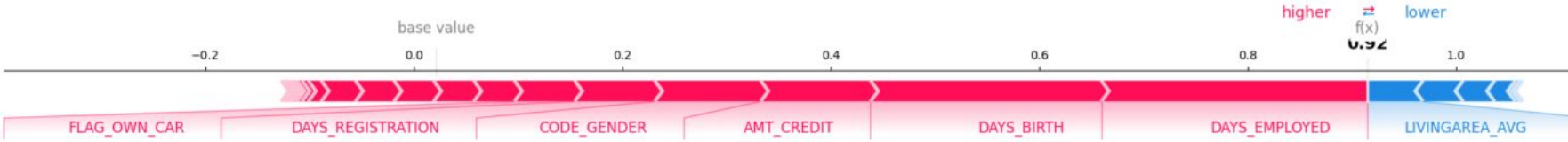
Il y a une probabilité 53.40 % que le créditeur rencontre des difficultés de paiement. La demande de crédit ne peut pas etre approuvée

Total enquiries number to Credit Bureau about the client

0

Prédire un nouveau résultat

Il y a une probabilité 71.40 % que le créditeur rencontre des difficultés de paiement. La demande de crédit ne peut pas etre approuvée



Sauvegarde de tous les livrables dans un référentiel GitHub (outil de gestion de version)

https://github.com/islem-habibi/projet_7

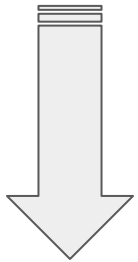
```
C:\Users\islem\OneDrive\Bureau\projet_7>echo "# projet_7" >> README.md

C:\Users\islem\OneDrive\Bureau\projet_7>ls
README.md               data_drift_test.ipynb      mlflow_flask_deployment.py        test_unitaire.py
app_streamlit.py        data_processing_module.py  projet7_final_version_2024.ipynb
data_drift_report.html  data_stability.html        score_metier_func.py

C:\Users\islem\OneDrive\Bureau\projet_7>git init
Initialized empty Git repository in C:/Users/islem/OneDrive/Bureau/projet_7/.git/

C:\Users\islem\OneDrive\Bureau\projet_7>git add .
warning: in the working copy of 'data_drift_test.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'data_processing_module.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'projet7_final_version_2024.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'score_metier_func.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'test_unitaire.py', LF will be replaced by CRLF the next time Git touches it

C:\Users\islem\OneDrive\Bureau\projet_7>git commit -m "first commit"
[master (root-commit) a19384c] first commit
 10 files changed, 13308 insertions(+)
 create mode 100644 README.md
 create mode 100644 app_streamlit.py
 create mode 100644 data_drift_report.html
 create mode 100644 data_drift_test.ipynb
 create mode 100644 data_processing_module.py
 create mode 100644 data_stability.html
 create mode 100644 mlflow_flask_deployment.py
 create mode 100644 projet7_final_version_2024.ipynb
 create mode 100644 score_metier_func.py
 create mode 100644 test_unitaire.py

C:\Users\islem\OneDrive\Bureau\projet_7>git branch -M main

C:\Users\islem\OneDrive\Bureau\projet_7>git remote add origin https://github.com/islem-habibi/projet_7.git

C:\Users\islem\OneDrive\Bureau\projet_7>git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 2.56 MiB | 1.96 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/islem-habibi/projet_7.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\islem\OneDrive\Bureau\projet_7>
```

Conclusion

Cross-validation : Modèle robuste et performant

déploiement: API Flask

Dashboard interactif: Streamlit



Impact sur la gestion des risques financiers et la prise de décision en matière de crédit pour "Prêt à dépenser".

Merci pour votre attention