Projet de Scoring de Crédit – API & Déploiement

🎯 Objectif du projet

L'objectif de ce projet est de développer et déployer une **API de scoring de crédit** permettant de prédire la probabilité de défaut de paiement d'un client.

Cette API constitue une **brique centrale** d'une solution décisionnelle destinée à la société financière, nommée **"Prêt à dépenser"** souhaitant **automatiser et fiabiliser** ses décisions d'octroi de prêts.

L'approche suit un cycle complet de mise en production d'un modèle de Machine Learning :

1. **Modélisation** → entraînement et sélection d'un modèle de scoring.
2. **Industrialisation** → encapsulation dans une API (FastAPI).
3. **Tests unitaires** → validation automatisée des fonctionnalités.
4. **Containerisation** → mise en place d'un environnement reproductible via Docker.
5. **Déploiement cloud** → hébergement de l'API sur Render avec CI/CD depuis GitHub.

📂 Organisation du projet

1. Code source

- **Zouak_Baya_1_API_082025.py** → code principal de l'API (FastAPI).
- **utils.py** → fonction de prétraitement (feature engineering)
- **modele_de_scoring.pkl** → pipeline de Machine Learning pré-entraîné (Joblib).
- **application_train.csv** → jeu de données utilisé pour générer le schéma Pydantic (colonnes des clients).

2. Tests

- **tests/test_api.py** → tests unitaires avec pytest + httpx.
    - Vérifie que l'API démarre correctement.
    - Vérifie que l'endpoint /predict renvoie bien une prédiction valide.

3. Dépendances

- **requirements.txt** → dépendances nécessaires en production.
- **requirements-dev.txt** → dépendances additionnelles pour le développement (tests, httpx, etc.).

4. Containerisation

- **Dockerfile** → décrit l'image Docker de l'API.
  - Basé sur **Python 3.10**
  - Installe les dépendances
  - Expose le service via **Uvicorn** sur le port 8000.

5. CI/CD

- **.github/workflows/tests.yml** → pipeline GitHub Actions qui exécute automatiquement les tests à chaque push ou pull request.
- **.github/workflows/deploy.yml** → pipeline GitHub Actions qui :
  - construit l'image Docker,
  - la publie sur **GitHub Container Registry (GHCR)**,
  - permet le déploiement sur Render.

🚀 Déploiement dans le cloud

L'API est hébergée sur **Render** via l'image stockée dans **GHCR**.

Étapes de déploiement

1. **Push du code** → déclenche les workflows GitHub Actions.
2. **Tests** → validation automatique avec pytest.
3. **Build & Push image** → l'image Docker est publiée sur GHCR.
4. **Render** → le service récupère l'image la plus récente et la déploie.

URL d'accès

Une fois déployée, l'API est accessible publiquement via :
👉 https://scoring-api-latest.onrender.com
👉 Documentation interactive : **/docs**

🔎 Utilisation de l'API

Endpoint **/predict**

- **Méthode** : POST
- **Entrée** : données client au format JSON (120 colonnes issues de application_train.csv)
- **Sortie** :

{

```
  "SK_ID_CURR": 100001,
  "probability": 0.745,
  "prediction": 1,
  "decision_message": "Refusé (risque de défaut élevé)"
}
```


✅ Conclusion

Ce projet illustre la mise en production d'un modèle de **scoring de crédit** avec une approche complète :

- modélisation,
- industrialisation,
- tests,
- CI/CD,
- containerisation et déploiement cloud.

Il constitue un **cas d'usage concret de MLOps appliqué au secteur bancaire**.

📂 Code source disponible sur GitHub :
☞ https://github.com/BayaZouak/scoring-api