



Mise en production d'un modèle de scoring crédit

Transformer un modèle MLOps en solution exploitable : API, Docker, CI/CD et monitoring en production

Par Abdourahamane LY



Contexte Métier : Prêt à Dépenser

L'entreprise Prêt à Dépenser utilise un modèle de scoring pour estimer la probabilité de défaut client. Le département Crédit Express a pour objectif de l'exploiter pour :

Analyser les demandes de crédit en quasi temps réel.



Aider à la décision (accepter/refuser un crédit).



Maîtriser le risque associé.

Le modèle existait déjà, entraîné et versionné. Ma mission était de le rendre pleinement opérationnel en conditions réelles.

Enjeux et objectifs du projet



Décision rapide

Répondre vite aux demandes tout en maîtrisant le risque

Fiabilité

Modèle stable, monitoré et compréhensible dans le temps

Traçabilité

Historique complet pour audit et amélioration continue

Industrialisation

Du notebook au service utilisable par toutes les équipes

Monitoring & Livrables

Ce que nous surveillons

Données d'entrée

Features clients : profils, montants ...



Performance opérationnelle

Latence, temps d'inférence, codes HTTP



Sorties modèle

Probabilités de défaut et prédictions

Dérive des données

Comparaison prod vs entraînement

✅ Analyse avec Evidently AI : **Data drift** : changements de distribution des features, et **Target drift** : évolution des scores et labels. Les logs JSON alimentent l'analyse de dérive et la détection d'anomalies.

Les Principaux Livrables

Une API FastAPI

Des tests

Un système de logging

Un README

Un Dockerfile

Un pipeline CI/CD

Une analyse de dérive



Optimisations de performance

État initial

Temps de réponse ~300 ms, modèle rechargé, image
Docker volumineuse

1

2

3

Résultats

Latence réduite à ~100-150 ms, API stable sous charge

Optimisations

Chargement unique, Docker slim, warm-up au démarrage

Goulots d'étranglement corrigés

Chargement modèle

Une seule fois au démarrage,
réutilisé pour toutes les
requêtes

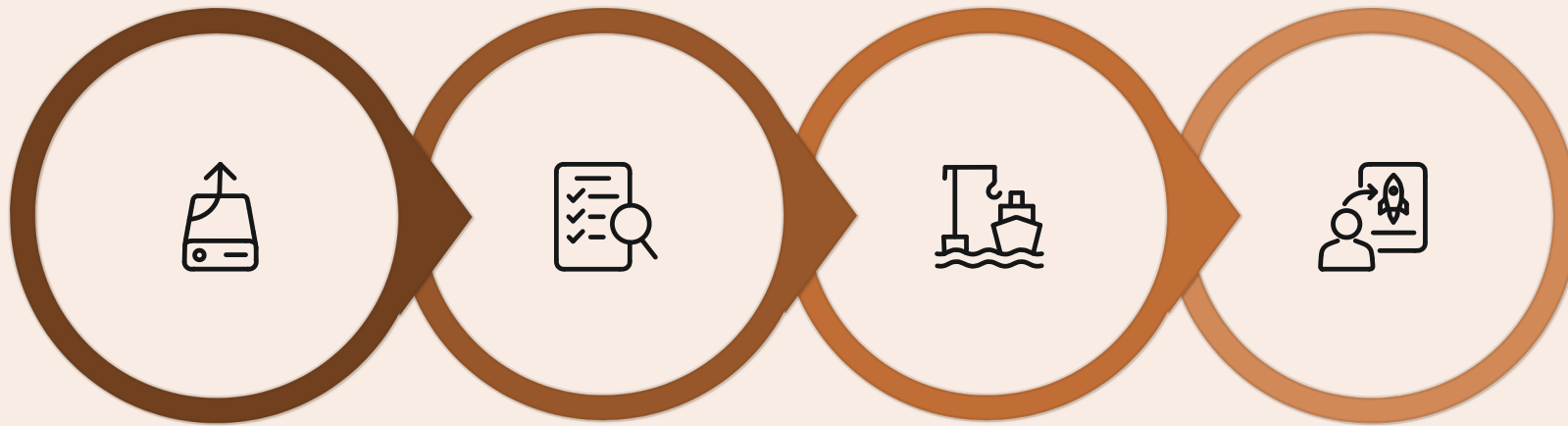
Image Docker allégée

Python 3.12-slim, dépendances
minimales uniquement

Warm-up startup

Prédiction fictive pour éviter le
cold start

Architecture du dépôt et pipeline CI/CD



Push
Code

Tests
automatiques

Build
Docker

Déploiement
Render

**Structure
organisée**

Déploiement automatisé à chaque push : tests, build et mise en production sans intervention manuelle

Bilan et perspectives d'évolution



Robustesse

Validation Pydantic, gestion des erreurs, logs traçables, codes HTTP explicites



Monitoring continu

Analyse dérive régulière, surveillance de la latence, possibilité d'alertes automatiques



Scalabilité

API stateless, scaling horizontal facile, modèle en mémoire répliquable

Prochaines étapes

Améliorations possibles

- Dashboard temps réel
- Alertes automatiques
- Réentraînement automatique

Un modèle de scoring crédit réellement exploitable en production : API robuste, conteneurisée, déployée automatiquement avec monitoring sérieux de la dérive et des performances.

Conclusion

L'impact principal est de permettre au département **Crédit Express** de prendre des décisions rapides et maîtrisées, tout en gardant une visibilité continue sur le comportement du modèle.

Ce projet illustre exactement le type de problématiques que je souhaite adresser en environnement professionnel, à l'interface entre **data science**, **production** et **métier**.

Merci pour votre attention.

Je suis maintenant disponible pour répondre à vos questions.

