Projet 7

Implémentez un modèle de scoring

Rédigé et présenté par

GASSUC CEDRIC

Mentor : Adrien Chambord

Introduction

Problématique : Comment proposez des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt ?

Objectif : Développer un modèle de scoring pour prédire les risques de défaut de paiement.

Plan

1. Prétraitements des données & Features Enginering
   - Nettoyage des données, Features
2. Stérilisation du modèle
   - Modèle Lightgbm
3. Création d'un API
   - Flask
4. Conteneurisation du modèle
   - Docker
5. Déploiement du modèle sur le cloud
   - Google cloud platform (GCP)
6. Tracking mlflow

1 Prétraitements des données & Features Enginering

1.1 Prétraitements des données

- **Nettoyage**
  - **Valeurs manquantes(** imputation , valeur nulle)
  - **Valeurs aberrantes (** boxplot**)**

1.2 Features enginering

Création des nouvelles variables en pourcentage

2 Equilibre des classes

- **Méthode :** SMOTE(Synthetic Minority Oversampling Technique)
- **Modèle :** Lightgbm

Avant SMOTE

train classification repport :

	precision	recall	f1-score	support
0	0.92	1.00	0.96	226148
1	0.66	0.01	0.02	19860
accuracy			0.92	246008
macro avg	0.79	0.50	0.49	246008
weighted avg	0.90	0.92	0.88	246008

validation classification repport:

	precision	recall	f1-score	support
0	0.92	1.00	0.96	56538
1	0.58	0.01	0.02	4965
accuracy			0.92	61503
macro avg	0.75	0.50	0.49	61503
weighted avg	0.89	0.92	0.88	61503

Avec SMOTE

Train Classification report :

	precision	recall	f1-score	support
Classe 0	0.96	0.70	0.81	197880
Classe 1	0.17	0.71	0.27	17377
accuracy			0.70	215257
macro avg	0.57	0.70	0.54	215257
weighted avg	0.90	0.70	0.77	215257

validation Classification report :

	precision	recall	f1-score	support
Classe 0	0.96	0.70	0.81	84806
Classe 1	0.16	0.68	0.26	7448
accuracy			0.70	92254
macro avg	0.56	0.69	0.54	92254
weighted avg	0.90	0.70	0.76	92254

2.2 Enregistrement du modèle au bon format pickle

3 Optimisation des hyperparamètres

Choix de hyperparamètre: **Optuna**

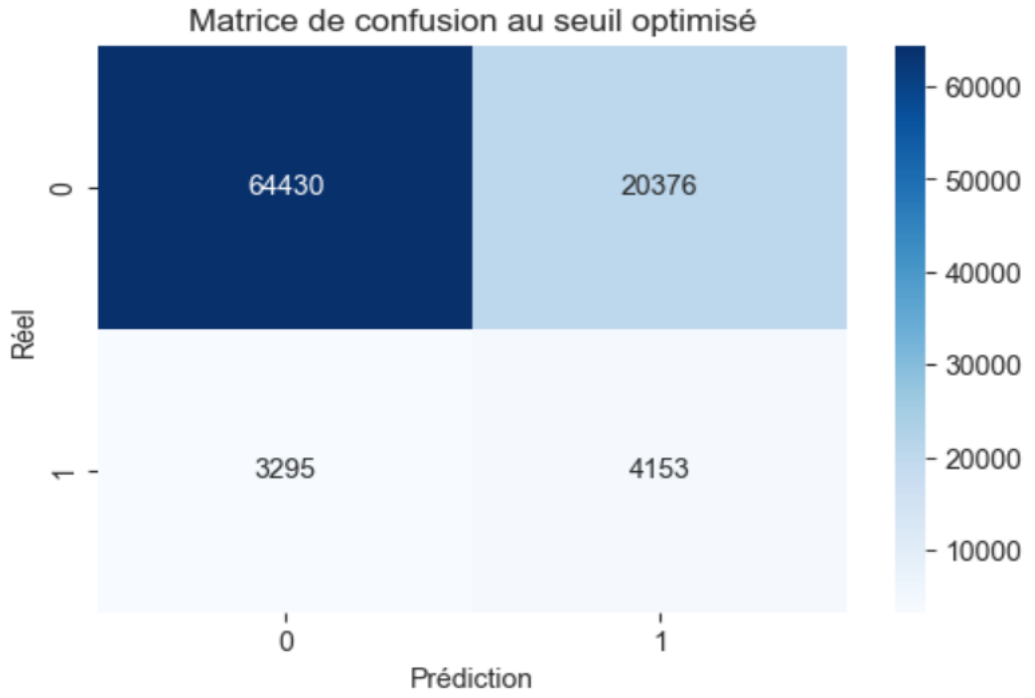
Caractéristiques

- **Rapide**
- **Flexible**
- **Asynchrone**

4 Fonction de coût métier

Cette fonction permet de trouver
le seuil optimal qui minimise
la perte métier

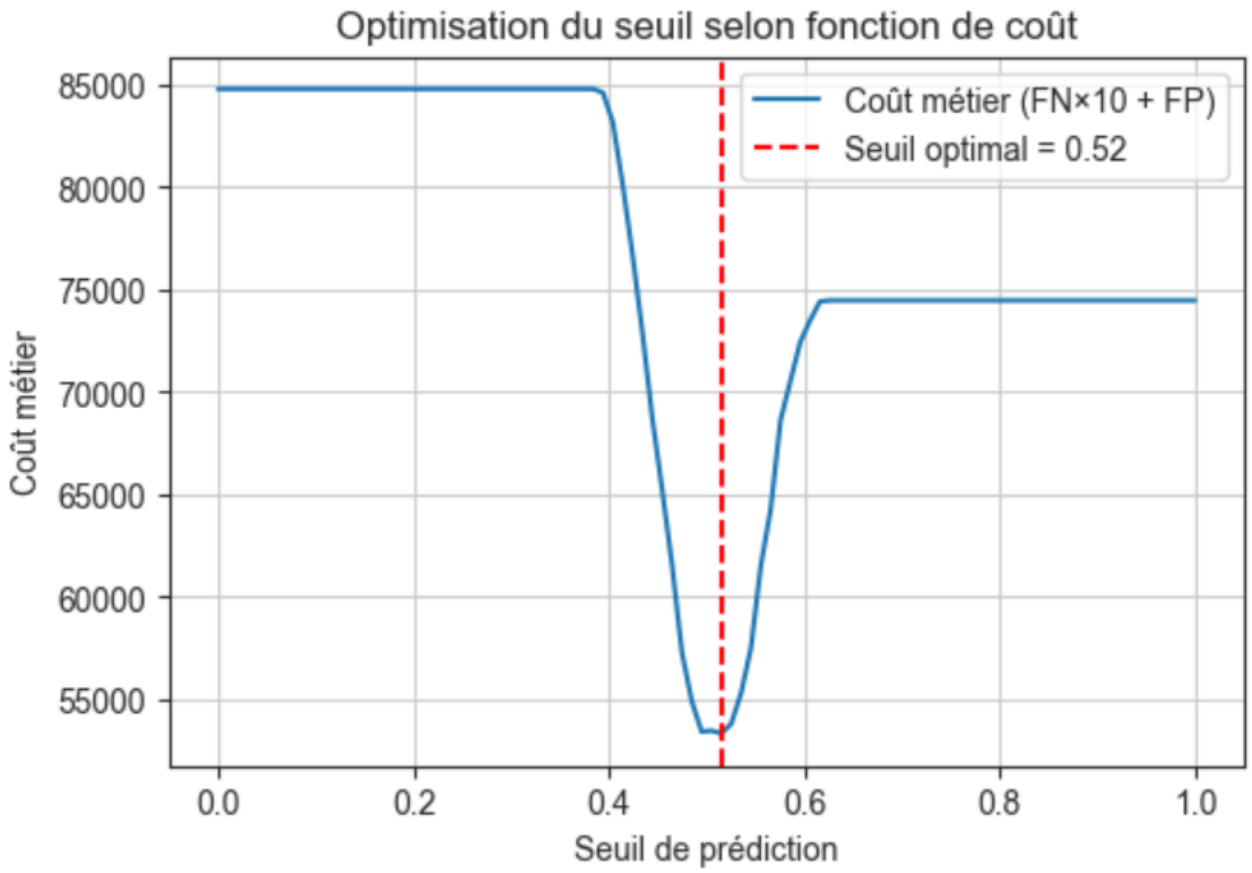
$$Fcm = FN*10 + FP*1$$



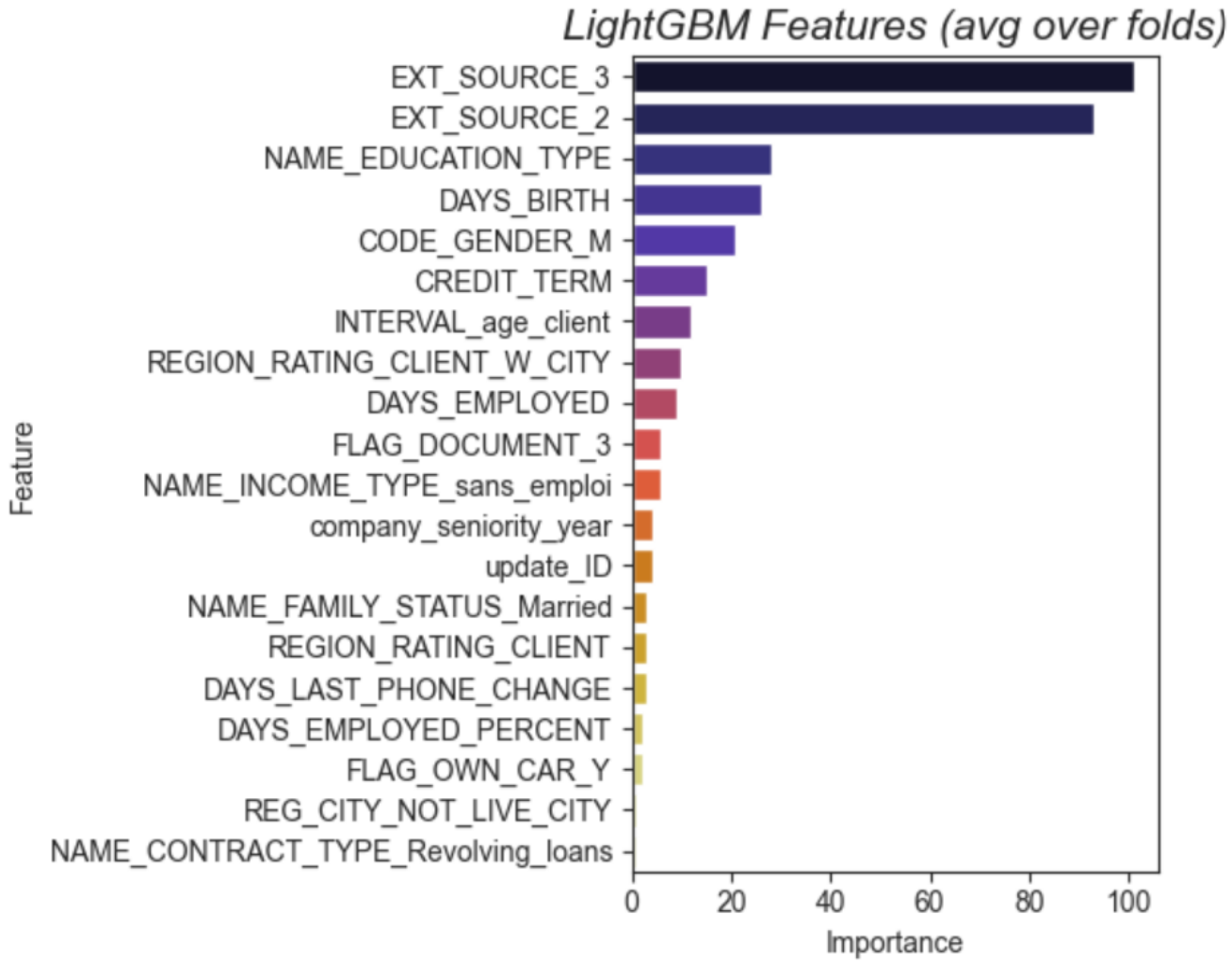
- **FN:** Vrai défaut classé bon client(perte)
- **FP:** Bon client classé comme défaut(manque à gagner)

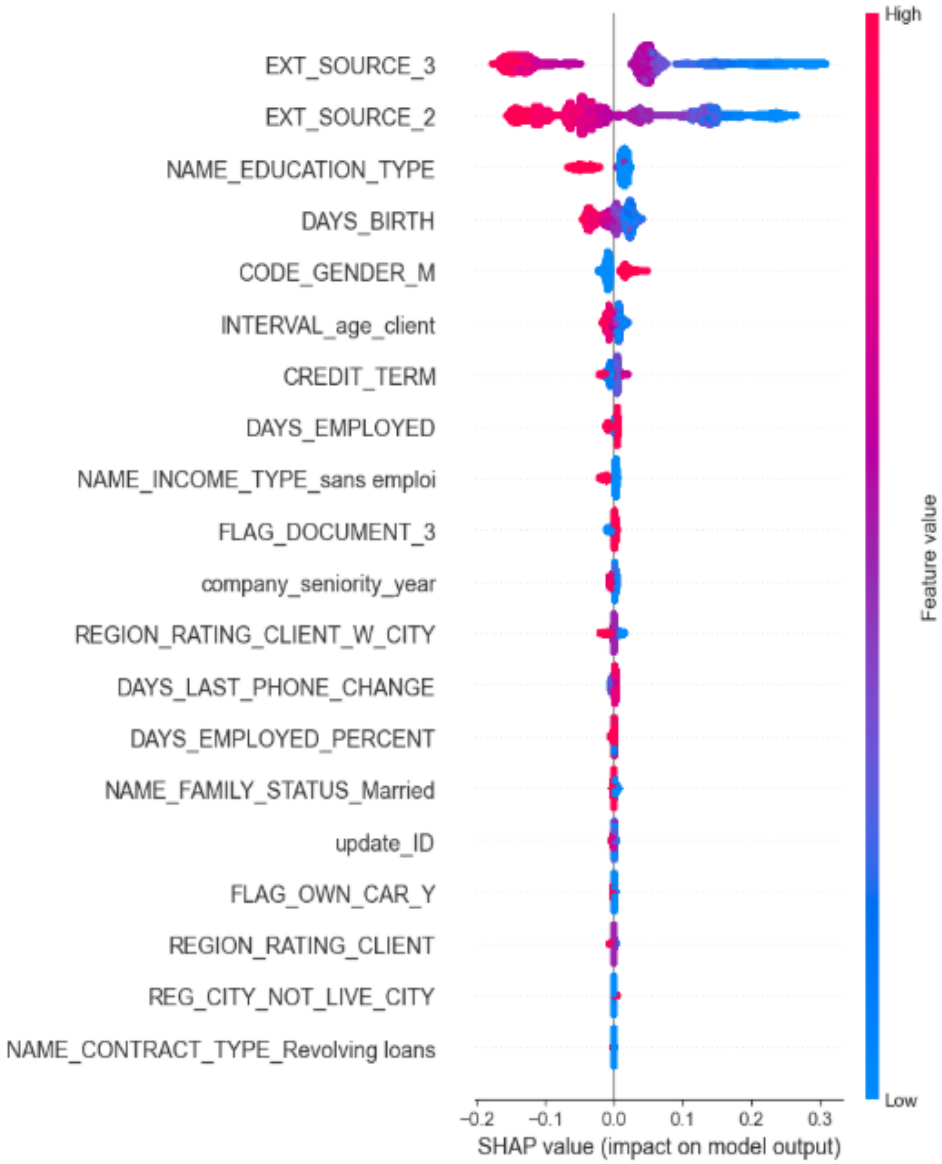
4 Seuil optimal et coût métier

- Seuil Optimal : **0,515**
- Coût métier minimal **: 53326**

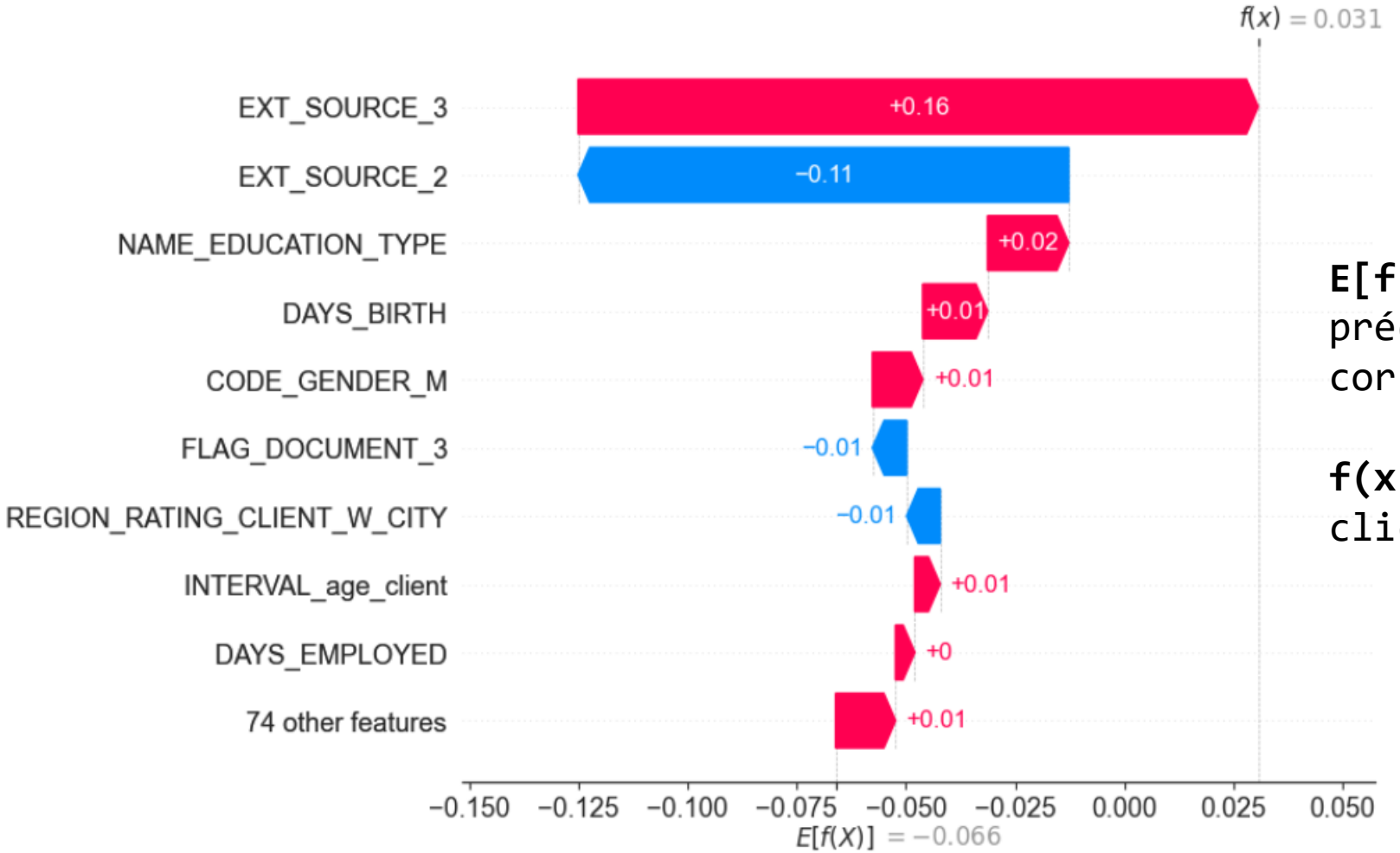


4 Features Importance





5 Analyse de la situation du client 15



E[f(X)] = -0.066: Moyenne des prédictions du modèle (cela correspond à un client "standard")

f(x) = 0.031: Score prédit pour ce client

3 **Création d'un API**

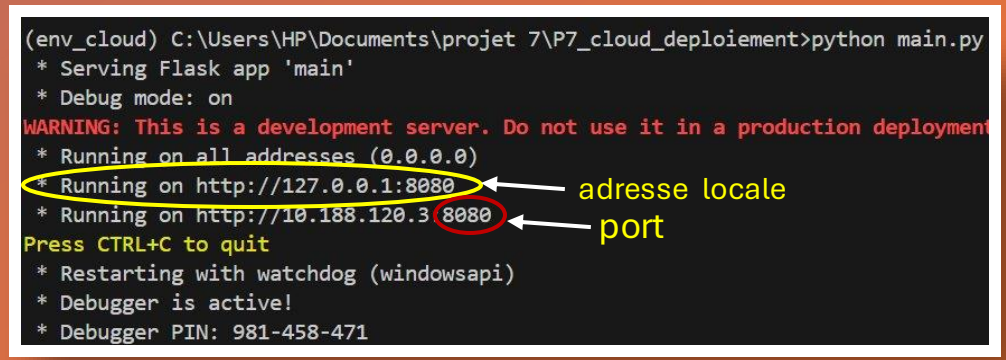
Application Flask

Interface légère pour servir le modèle

- **Flask** : Micro-framework Python léger, idéal pour créer des APIs.

- **Endpoint principal (`/predict`)** :
- Reçoit des données en JSON (via POST)
- Prétraite les données si nécessaire
- Appelle le modèle entraîné (chargé en mémoire ou via fichier)
- Retourne la prédiction sous forme JSON

Exécution de l'API Flask en mode développement local (localhost:8080)



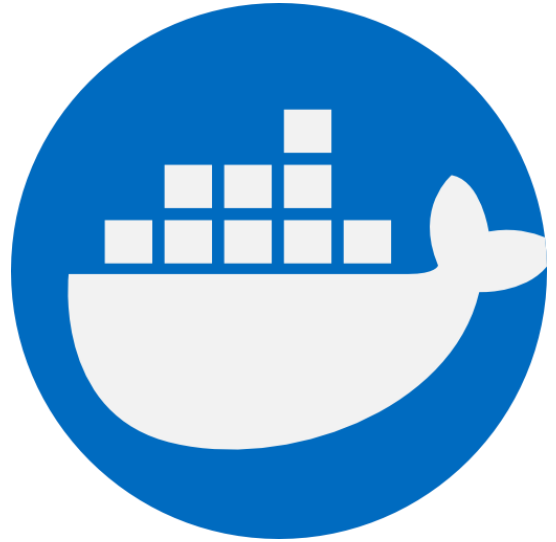
4 Conteneurisation du modèle : Docker

Objectif

Isoler l'application Flask et sesdépendances dans un conteneur Dockerpour un déploiement cohérent et sécurisésur le Cloud.

Fichiers essentiels

- **Dockerfile**
- **Requirements**
- **Main.py (app docker)**



Docker

Rendre l'application portable et déployable

Test du modèle en local

```
200 {"message":"Bienvenue sur l'API de Prediction sur l'accord d'un pr\u00eat Bancaire"}

Réponse du endpoint d'accueil: {"message":"Bienvenue sur l'API de Prediction sur l'accord d'un pr\u00eat Bancaire"}

Prédiction : 0
Probabilité : 0.4385
decision : Accordé
PS C:\Users\HP\Documents\projet 7\P7_cloud_deploiement>
```

5 Déploiement du modèle : Google Cloud Run

Google cloud platform (GCP)

Objectif

Déployer l'image Docker de l'application Flask sur le Cloud

Étapes du déploiement

1. **Build & push image** sur Google Container Registry

2. **Déploiement sur Cloud Run**

6 UI Streamlit

Interface web

Streamlit

Déployée séparément, envoie des requêtes à l'API

6 Test du modèle à distance

Application de prédiction de prêt

Téléversez un fichier CSV contenant les données d'un ou plusieurs clients.

Choisissez un fichier CSV

Drag and drop file here
Limit 200MB per file • CSV

Browse files

donnees_clients_50_final.csv 26.1KB

Données chargées

	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	NAME_EDUCATION_TYPE	DAYS_BIRTH	DAYS_EMP
0	135000	568800	20560.5	3	53	
1	99000	222768	17370	1	49	
2	202500	663264	69777	3	55	
3	315000	1575000	49018.5	1	38	
7	166500	180000	14220	3	26	
8	315000	364896	28957.5	3	35	
9	162000	45000	5337	3	28	

Lancer la prédiction

Résultats des prédictions

	Client	Prédiction	Probabilité	Décision
0	1	0	0.3668	Accordé
1	2	1	0.637	Refusé
2	3	0	0.1985	Accordé
3	4	0	0.2829	Accordé

Conclusion

Merci pour Votre Compréhension