

Projet 7: Implémenter un modèle de scoring

Maodo FALL

OpenClassrooms

Soutenance du projet
03 mars 2025



Sommaire

- 1 **Problématique et jeu de données**
 - Problématique
 - Jeu de données
 - Analyse exploratoire des données
- 2 **Modélisation et tracking d'expérience avec MLflow**
- 3 **Pipeline de déploiement continu**
- 4 **Conclusion**

Sommaire

- 1 **Problématique et jeu de données**
 - Problématique
 - Jeu de données
 - Analyse exploratoire des données
- 2 Modélisation et tracking d'expérience avec MLflow
- 3 Pipeline de déploiement continu
- 4 Conclusion

Problématique

Société financière



Problématique

La société financière nommée "Prêt à dépenser" souhaite mettre en œuvre un outil de "scoring crédit" pour déterminer de la solvabilité ou non de ses clients demandant un prêt. Elle souhaite donc développer un algorithme de classification en s'appuyant sur des sources de données variées.

Missions :

- Construire un modèle de scoring qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique.
- Analyser les features qui contribuent le plus au modèle .
- Déployer le modèle sous forme d'API et réaliser une interface de test de cette API.
- Mettre en œuvre une approche MLOps de bout en bout.

Jeu de données

- Les données sont issues d'un dataset sur [kaggle](#).
- Elles sont constituées de dix tables.
- Le jeu de données principal contient :
 - 307511 clients
 - 121 features
 - la variable TARGET qu'on veut prédire.

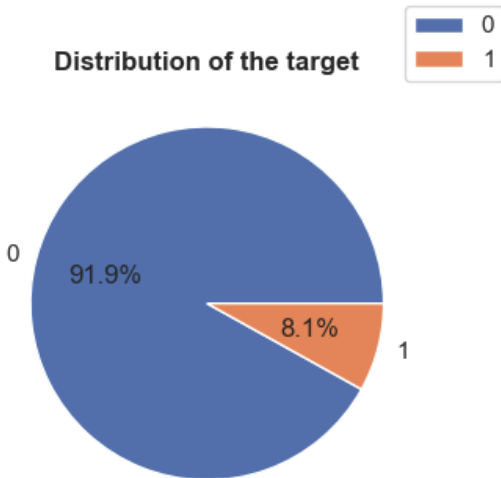
Etapes du traitement des données

Nous avons effectué un travail de :

- exploration des données
- traitement des valeurs manquantes.
- visualisation de la distribution des features.
- étude de la corrélation entre les features et la cible.
- feature engineering à l'aide des kernels kaggle comme [celle-ci](#) incluant des étapes de :
 - one-hot encoding pour les variables catégorielles
 - Création de variables supplémentaires
 - Agrégation des autres tables par numéro de client (somme, moyenne, max, min)
- jointure des tables afin obtenir le jeu de données final pour la modélisation.

Distribution de la TARGET

Distribution of the target



Sommaire

- 1 Problématique et jeu de données
 - Problématique
 - Jeu de données
 - Analyse exploratoire des données
- 2 **Modélisation et tracking d'expérience avec MLflow**
- 3 Pipeline de déploiement continu
- 4 Conclusion

Approche de modélisation

J'ai :

- testé plusieurs modèles de machine learning
- comparé leurs performances
- défini un score métier de performance
- utilisé MLflow pour tracker toutes mes expériences.

Interface UI MLflow

Home_credit_card_models [Provide Feedback](#) [Add Description](#)

Runs

Evaluation

Experimental

Traces

Time created

State: Active

Datasets

Sort: Created

Columns

Group by

	Run Name	Created	Dataset	Duration	Source	Models	
<input type="checkbox"/>	lgbm-scaled	16 days ago	-	1.4min		+1	
<input type="checkbox"/>	lgbm-scaled	19 days ago	-	2.3min		+1	
<input type="checkbox"/>	lr-gs-scaled	19 days ago	-	8.0s			
<input type="checkbox"/>	rf-under-scaled	19 days ago	-	9.6s			
<input type="checkbox"/>	rf-smote-scaled	19 days ago	-	15.2s			
<input type="checkbox"/>	lr-smote-scaled	19 days ago	-	7.9s			
<input type="checkbox"/>	lr-under-scaled	19 days ago	-	7.1s			
<input type="checkbox"/>	lr-scaled	19 days ago	-	20.4s			
<input type="checkbox"/>	lr-scaled	19 days ago	-	27.3s			
<input type="checkbox"/>	Log_reg-smote-featscale	26 days ago	-	7.4s			
<input type="checkbox"/>	Log_reg-undersample-fea...	26 days ago	-	7.5s			
<input type="checkbox"/>	Log_reg-featscale	26 days ago	-	7.4s			
<input type="checkbox"/>	RF-smote-featscale	28 days ago		6.2min		+1	
<input type="checkbox"/>	LightGBM-featscale	28 days ago	-	19.9s			
<input type="checkbox"/>	LightGBM-featscale	29 days ago	-	11.0s			
<input type="checkbox"/>	RF-undersample-featscale	29 days ago		32.4s		+1	
<input type="checkbox"/>	RF-smote-featscale	29 days ago		5.5min		+1	

20 matching runs

Show more columns (87 total)

Meilleur modèle : LGBMClassifier

Confusion matrix :

[[40950 15587]

[1544 3421]]

Classification report :

	precision	recall	f1-score	support
0	0.96	0.72	0.83	56537
1	0.18	0.69	0.29	4965
accuracy			0.72	61502
macro avg	0.57	0.71	0.56	61502
weighted avg	0.90	0.72	0.78	61502

Accuracy : 0.721

Precision : 0.18

Recall : 0.689

F1 score : 0.285

ROCAUC score : 0.707

Fowlkes_Mallows score : 0.732

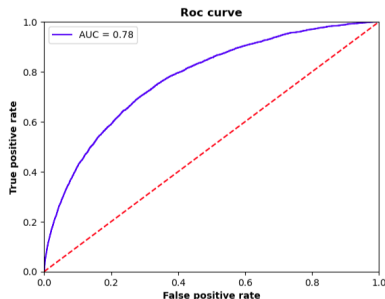


Figure – Statistiques du modèle

Figure – Courbe AUROC

Prédiction au seuil métier optimal

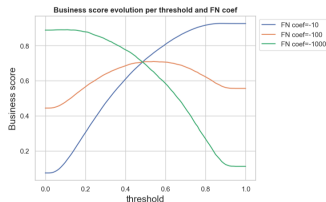


Figure – Optimisation seuil

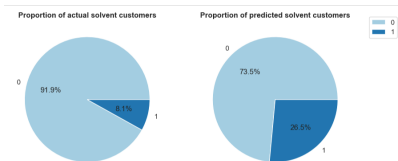


Figure – Prédiction au seuil optimal

Importance des features

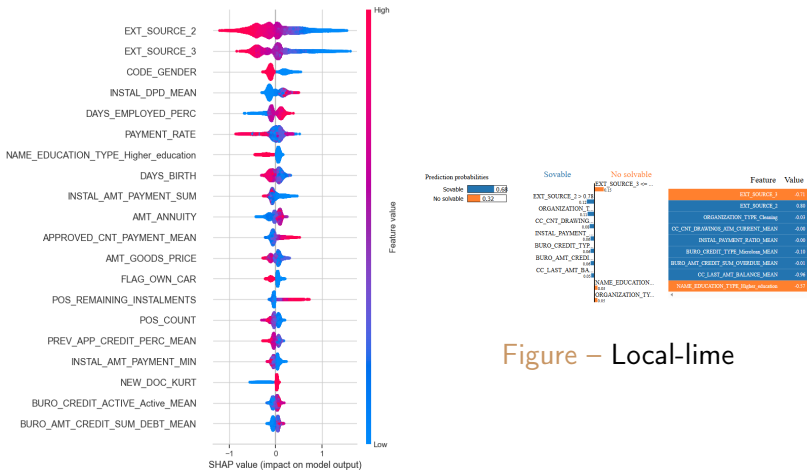


Figure – Local-lime

Figure – Globale-shape

Sommaire

- 1 Problématique et jeu de données
 - Problématique
 - Jeu de données
 - Analyse exploratoire des données
- 2 Modélisation et tracking d'expérience avec MLflow
- 3 Pipeline de déploiement continu
- 4 Conclusion

Github

Lien vers le [dossier github](#)

The screenshot shows the GitHub interface for a repository named 'deployment-scoring-repo' by user 'Maodofall'. The repository is public and has 16 commits. The main branch is 'main'. The repository description is 'Github repo for my scoring model deployment'. The repository is licensed under Apache-2.0. The repository has 0 stars, 1 watching, and 0 forks. The repository has no releases or packages published. The repository has a README file and a requirements.txt file. The repository has a Dockerfile and a .gitignore file. The repository has a dashboard_streamlit file and a tests directory. The repository has an api_test file and a .github/workflows directory. The repository has a README.md file and a requirements.txt file. The repository has a Dockerfile and a .gitignore file. The repository has a dashboard_streamlit file and a tests directory. The repository has an api_test file and a .github/workflows directory.


File	Commit Message	Time
Maodofall Update README.md		6m55s · 2 hours ago
.github/workflows	ajout du repertoire racine du projet au PYTHONPATH	yesterday
api_test	pushing my modifs to aws	2 days ago
dashboard_streamlit	ajout du dossier notebook	20 hours ago
tests	correction du fichier de pylast	yesterday
.gitignore	Initial commit	2 days ago
Dockerfile	pushing my modifs to aws	2 days ago
Fall_Maodo_2_notebook_modelisation_022025...	ajout des notebooks	20 hours ago
Fall_Maodo_2_notebook_pretraitement_02202...	ajout des notebooks	20 hours ago
LICENSE	Initial commit	2 days ago
README.md	Update README.md	2 hours ago
requirements.txt	update requirements file	yesterday

README Apache-2.0 license

Implémenter et déployer un modèle de scoring

Les commits

Commits

 main

🔍 All users

📅 All time

Commits on Mar 1, 2025

Update README.md

Maodofall authored 2 hours ago ✓ 1 / 1

Verified 6cd958b

📄 <>

Commits on Feb 28, 2025

ajout des notebooks

Maodofall committed 20 hours ago ✓ 1 / 1

ea070f6

📄 <>

ajout du dossier notebook

Maodofall committed 20 hours ago

3e7eFab

📄 <>

Update README.md

Maodofall authored 20 hours ago ✓ 1 / 1

Verified c87fe2c

📄 <>

Update README.md

Maodofall authored yesterday ✓ 1 / 1

Verified 2519db2

📄 <>

maj du frontend

Maodofall committed yesterday ✓ 1 / 1

ebd9671

📄 <>

ajout du dashboard streamlit

Maodofall committed yesterday ✓ 1 / 1

0e9e51c

📄 <>

correction du fichier de pytest

Maodofall committed yesterday ✓ 1 / 1

416f5ed

📄 <>

correction du test unitaire

Maodofall committed yesterday ✗ 0 / 1

1811acd

📄 <>

update requirements file

Maodofall committed yesterday ✗ 0 / 1

2ea2984

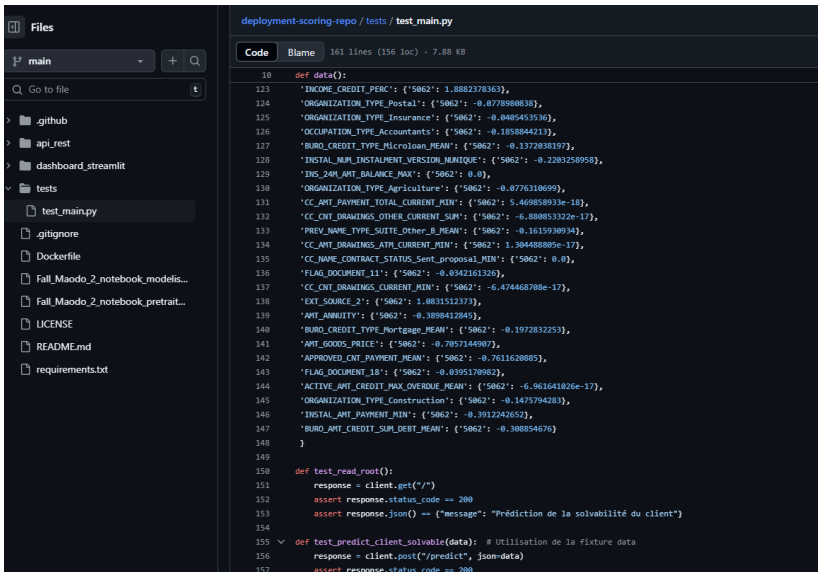
📄 <>

ajout du répertoire racine du projet au PYTHONPATH

h1r-9911

📄 <>

Les tests unitaires



The screenshot shows a code editor interface. On the left is a file explorer with a 'Files' tab. It shows a project structure with folders like '.github', 'api_rest', 'dashboard_streamlit', and 'tests'. The 'tests' folder is expanded, showing a file named 'test_main.py'. On the right is the code editor for 'test_main.py'. It shows a Python file with a 'data()' function and a 'test_read_root()' function. The 'data()' function returns a dictionary of test data. The 'test_read_root()' function uses a client to get the root of the deployment-scoring-repo. The code is as follows:

```
deployment-scoring-repo / tests / test_main.py

Code Blame 161 lines (156 loc) · 7.88 KB

10 def data():
123     'INCOME_CREDIT_PERC': {'5062': 1.8882378363},
124     'ORGANIZATION_TYPE_Postal': {'5062': -0.0778980838},
125     'ORGANIZATION_TYPE_Insurance': {'5062': -0.0405453536},
126     'OCCUPATION_TYPE_Accountants': {'5062': -0.1858844213},
127     'BURO_CREDIT_TYPE_Microloan_MEAN': {'5062': -0.1372838197},
128     'INSTAL_NUM_INSTALMENT_VERSION_UNIQUE': {'5062': -0.2203258958},
129     'INS_24M_AMT_BALANCE_MAX': {'5062': 0.0},
130     'ORGANIZATION_TYPE_Agriculture': {'5062': -0.0776310699},
131     'CC_AMT_PAYMENT_TOTAL_CURRENT_MIN': {'5062': 5.469858933e-18},
132     'CC_CNT_DRAWINGS_OTHER_CURRENT_SUM': {'5062': -6.880853322e-17},
133     'PREV_NAME_TYPE_SUITE_Other_B_MEAN': {'5062': -0.1615930934},
134     'CC_AMT_DRAWINGS_ATM_CURRENT_MIN': {'5062': 1.304488805e-17},
135     'CC_NAME_CONTRACT_STATUS_Sent_proposal_MIN': {'5062': 0.0},
136     'FLAG_DOCUMENT_11': {'5062': -0.0342161326},
137     'CC_CNT_DRAWINGS_CURRENT_MIN': {'5062': -6.474468708e-17},
138     'EXT_SOURCE_2': {'5062': 1.0831512373},
139     'AMT_ANNUITY': {'5062': -0.3898412845},
140     'BURO_CREDIT_TYPE_Mortgage_MEAN': {'5062': -0.1972832253},
141     'AMT_GOODS_PRICE': {'5062': -0.7057144907},
142     'APPROVED_CNT_PAYMENT_MEAN': {'5062': -0.7611620885},
143     'FLAG_DOCUMENT_18': {'5062': -0.0395170982},
144     'ACTIVE_AMT_CREDIT_MAX_OVERDUE_MEAN': {'5062': -6.961641026e-17},
145     'ORGANIZATION_TYPE_Construction': {'5062': -0.1475794283},
146     'INSTAL_AMT_PAYMENT_MIN': {'5062': -0.3912242652},
147     'BURO_AMT_CREDIT_SUM_DEBT_MEAN': {'5062': -0.308854676}
148 }
149
150 def test_read_root():
151     response = client.get("/")
152     assert response.status_code == 200
153     assert response.json() == {"message": "Prédiction de la solvabilité du client"}
154
155 def test_predict_client_solvable(data): # Utilisation de la fixture data
156     response = client.post("/predict", json=data)
157     assert response.status_code == 200
```

Déploiement de l'API

Lien vers l'API

The screenshot displays the Amazon Elastic Container Service (ECS) console interface. The top navigation bar shows the service name 'deployment-scoring-service' and its status 'Actif'. The left sidebar contains links to various ECS features like Clusters, Tâches, Journaux, etc. The main content area provides an overview of the service, including its ARN, current deployment status, and performance metrics. Two line charts are visible: 'Utilisation du processeur' (CPU usage) and 'Utilisation de la mémoire' (Memory usage), both showing a sharp spike followed by a drop.

Amazon Elastic Container Service

deployment-scoring-service

Aperçu du service

Statut: Actif

Tâches (1 souhaitées): 0 En attente | 1 En cours d'exécution

Définition de la tâche: [révision deployment-scoring-task-1](#)

Statut du déploiement: Réussite

État et métriques | Tâches | Journaux | Déploiements | Événements | Configuration et mise en réseau | Autoscaling du service | Balises

Statut

Nom du service: [deployment-scoring-service](#)

ARN du service: [arn:aws:ecs:us-east-1:98553977050:service/deployment-scoring-cluster2/deployment-scoring-service](#)

État actuel des déploiements: 1 tâche terminée

Créé à: 27 février 2025 à 20:32 (UTC+1:00)

Période de grâce de la vérification de l'état: 0 secondes

État

☐ Recommandations d'alarmes

1h 3h 12h 1j 1sem. **Personnalisées** Fuseau horaire UTC + - Ajouter au tableau de bord

Utilisation du processeur

Percent

19.1

9.63

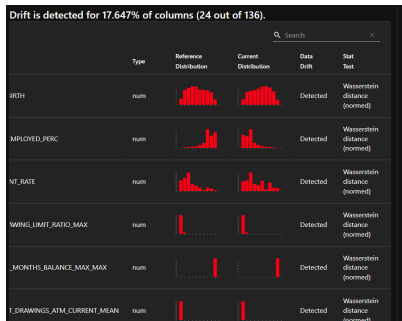
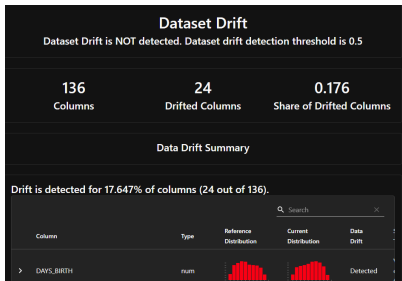
Utilisation de la mémoire

Percent

5.15

2.56

Analyse du data drift



Sommaire

- 1 Problématique et jeu de données
 - Problématique
 - Jeu de données
 - Analyse exploratoire des données
- 2 Modélisation et tracking d'expérience avec MLflow
- 3 Pipeline de déploiement continu
- 4 Conclusion

Conclusion

J'ai développé un modèle de scoring crédit afin d'évaluer la solvabilité des clients de la société "Prêt à dépenser".

- **Exploration et traitement des données** : nettoyage, gestion des valeurs manquantes et feature engineering.
- **Modélisation et expérimentation** : comparaison de plusieurs modèles avec MLflow pour optimiser les performances.
- **Choix du meilleur modèle** : **LGBMClassifier**, offrant un bon compromis entre meilleur score métier et rapidité.
- **Déploiement du modèle** : création d'une API avec **AWS ECS** et mise en place d'un pipeline CI/CD via **GitHub Actions**.
- **Suivi et maintenance** : analyse du **data drift** pour garantir la robustesse du modèle sur le long terme.

Conclusion

Perspectives d'amélioration :

- Affiner les hyperparamètres pour maximiser la performance.
- Intégrer de nouvelles sources de données pour enrichir les prédictions.
- Automatiser davantage le monitoring du modèle en production.

Grâce à cette approche **MLOps complète**, j'ai démontré qu'un **modèle de scoring fiable, scalable et maintenable** peut être mis en place efficacement.

MERCI BEAUCOUP !