



IMPLÉMENTEZ UN  
MODÈLE DE  
SCORING

---

**DATA SCIENCE**

# SOMMAIRE

01

02

03

04

# SOMMAIRE

Présentation du projet

01

02

03

04

# SOMMAIRE

Présentation du projet

01

Modélisation et  
Déploiement du modèle

02

03

04

# SOMMAIRE

Présentation du projet

01

Modélisation et  
Déploiement du modèle

02

Analyse de data Drift

03

04





# SOMMAIRE

Présentation du projet **01**

Modélisation et  
Déploiement du modèle **02**

Analyse de data Drift **03**

Démo du Dashboard  
dans cloud **04**

# PRESENTATION

---



## Contexte

- Mise en œuvre un outil de “**scoring crédit**”.
- Développement de **Dashboard interactif** .

# PRESENTATION

---

## Objectifs

- Construire un modèle donnant la probabilité de faillite d'un client.
- Mettre en production le modèle à l'aide d'une API.
- Construire un Dashboard interactif



## Contraintes

- Sélectionner et adapter un kernels **Kaggle**
- Utiliser **Dash** pour réaliser le Dashboard interactif
- Réaliser la data Drift avec la librairie **evidently**



# PRESENTATION

---

## Méthodologie

- Réaliser et évaluer le modèle de scoring de prédiction sur la probabilité de faillite d'un client de façon automatique..
- Etablir un pipeline de déploiement du modèle
- Réaliser une analyse de data Drift du modèle
- Mettre en place une Dashboard interactif



## MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

## Présentation du jeu de données

## Description

- ❑ Le jeu de données est composé de sept (7) fichiers au format CSV
- ❑ Les données utilisées proviennent des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.),
- ❑ Elles comprennent des variables telles que le revenu, le montant des crédits, l'historique de remboursement, et des données démographiques.





# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

## Présentation du jeu de données

### Transformation des données

Nous avons sélectionné et adapté un kernels Kaggle aux besoins de la mission.

- ☐ Le Kernel nous a permis d'effectuer les tâches suivantes :
  - ✓ Préparation des données et de
  - ✓ Feature engineering pour le modèle de scoring,
  - ✓ Consolidation du jeu de données en une seule dataframe.
- ☐ La séparation du dataframe en données d'entraînement en données test
- ☐ La modélisation



# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

## Présentation de la modélisation

### Démarche de modélisation

- ❑ L'algorithme, un modèle LightGBM GBDT avec KFold.
- ❑ Hyperparamètres :
  - **Learning Rate** : Pour contrôler la vitesse d'apprentissage.
  - **Nombre d'Estimates** : Pour éviter le surajustement.
  - **Max Depth** : Pour contrôler la complexité du modèle.
- ❑ Trois (3) entraînement du modèle :
  - ✓ Modèle avec déséquilibre des classes,
  - ✓ Modèle avec l'équilibrage des données par SMOTE,
  - ✓ Modèle par Implémentation d'un score métier pour prioriser le FN





# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

## Présentation de la modélisation

### Evaluation du modèle

- ❑ Choix des mesures :
  - ✓ L'AUC (**Area Under the Curve**) : mesure la capacité d'un modèle à distinguer entre les deux classes.
  - ✓ Accuracy : le rapport entre les prédictions correctes et le nombre total de prédictions

### ❑ La synthèse des résultats :

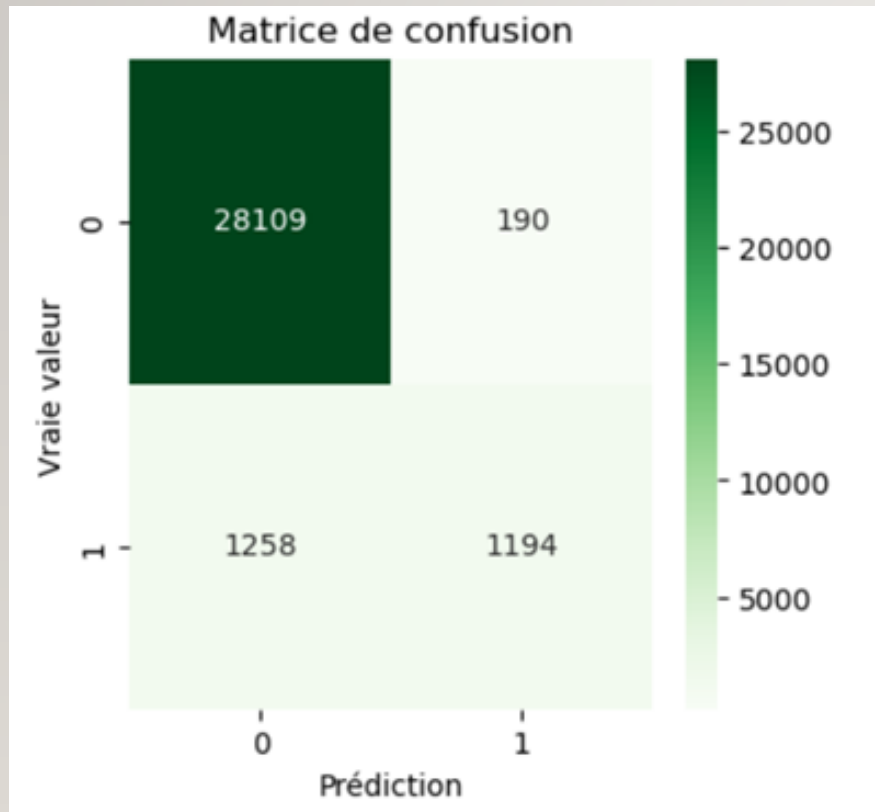
Métrique	Modèle avec déséquilibre	Modèle équilibré par SMOTE	Modèle avec score métier
AUC	0,731	0,889	0,731
Accuracy	0,918	0,953	0,920



# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

## Présentation de la modélisation

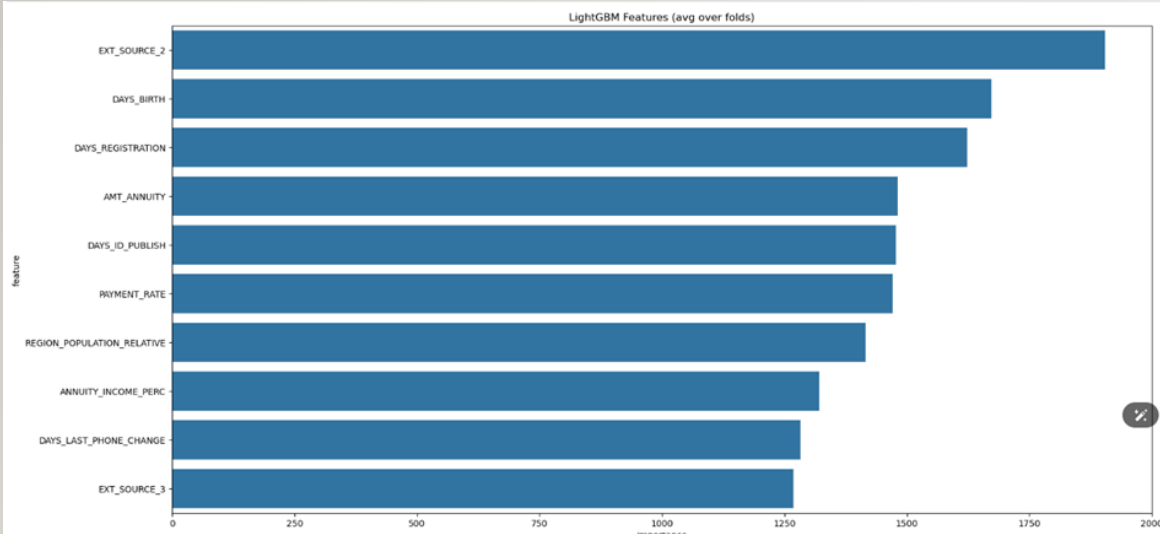
### Matrice de confusion





## Présentation de la modélisation

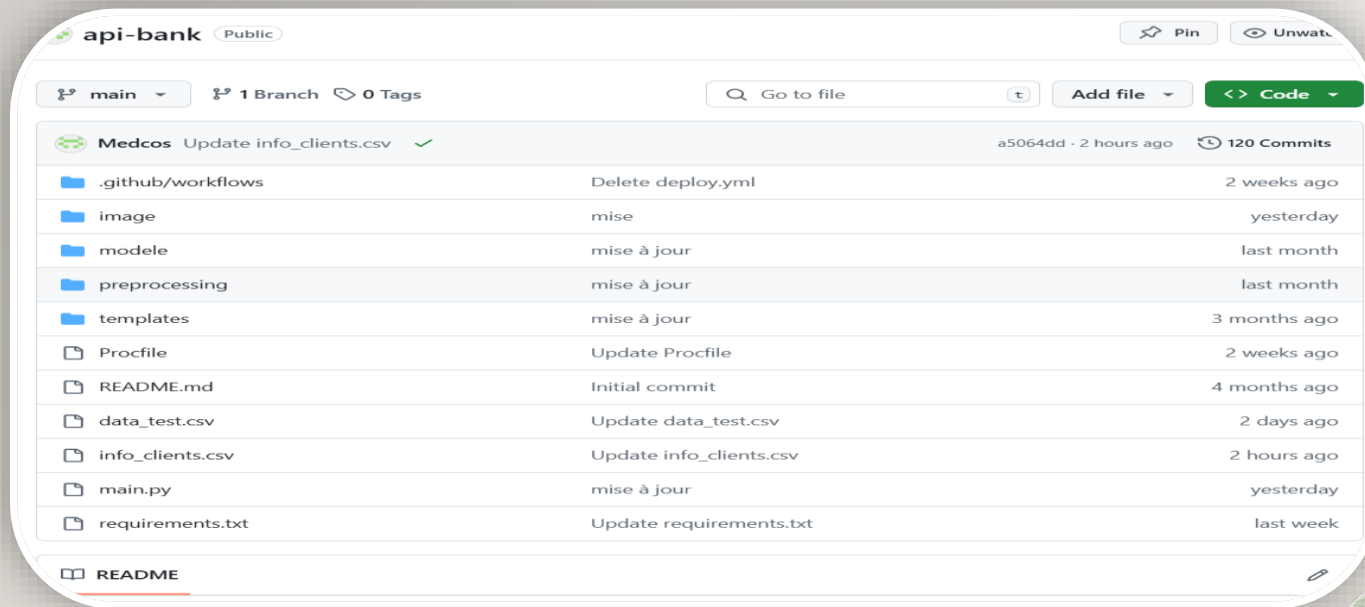
## Feature importance



# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

Pipeline de déploiement : <https://github.com/Medcos>

## Git, Github, tests unitaires

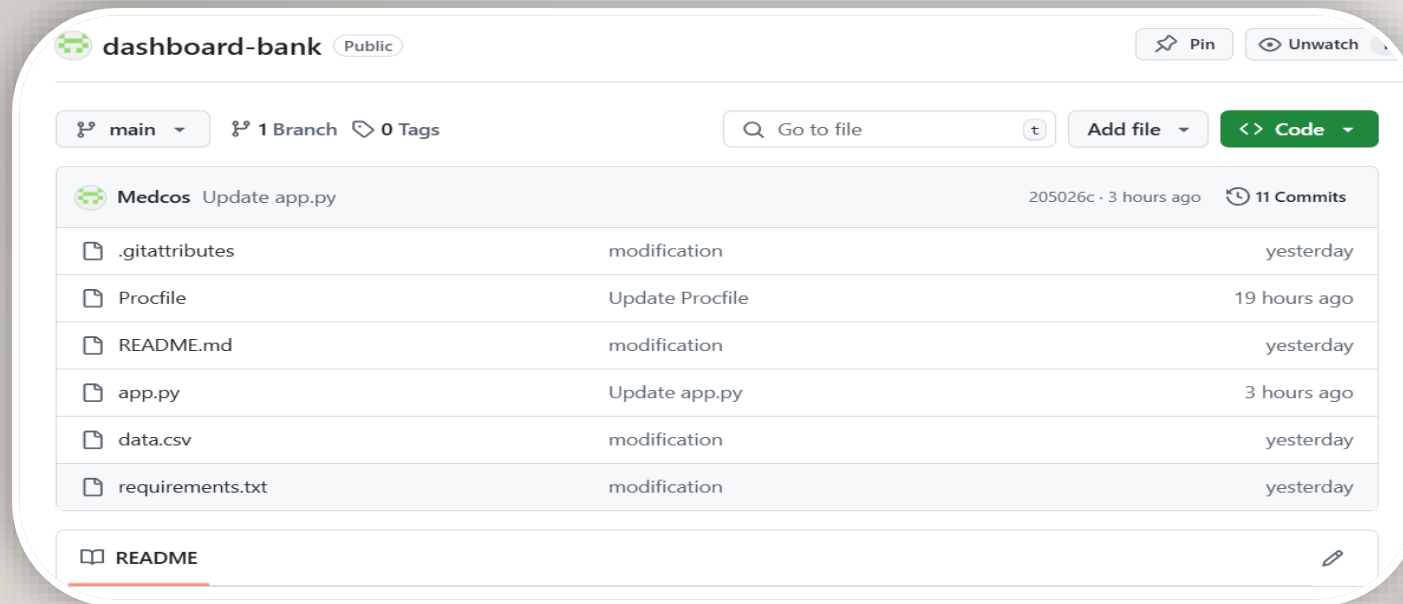




# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

Pipeline de déploiement : <https://github.com/Medcos>

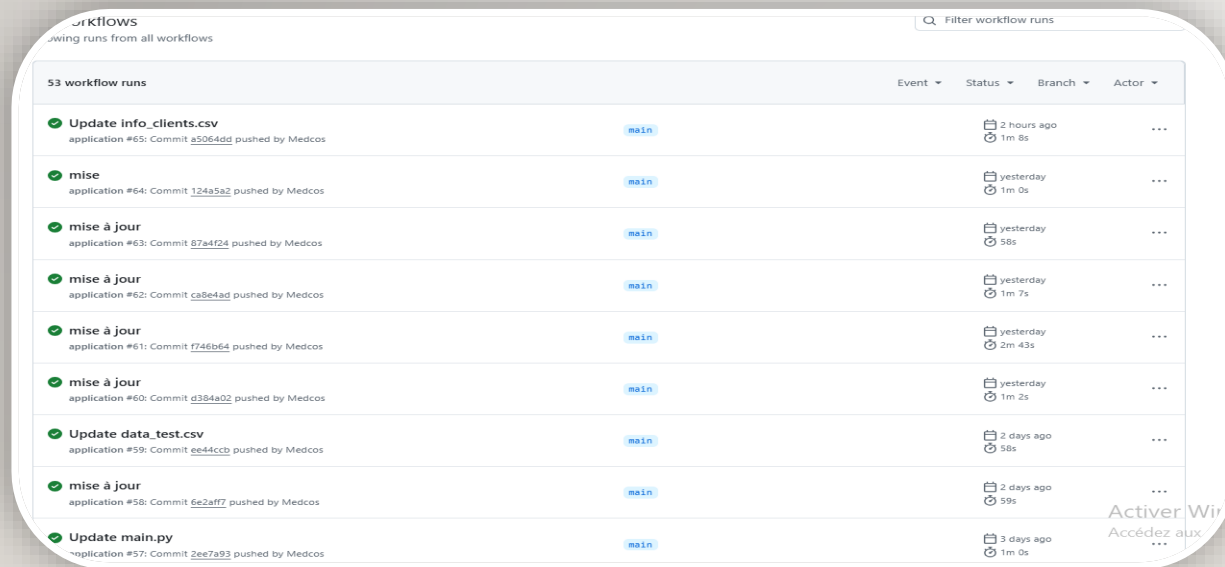
## Git, Github, tests unitaires



# MODÉLISATION ET DÉPLOIEMENT DU MODÈLE

Pipeline de déploiement : <https://github.com/Medcos>

## Git, Github, tests unitaires



The screenshot shows the 'Workflow runs' page for the 'Medcos' repository on GitHub. It displays a list of 53 workflow runs, with the following details visible:

Workflow	Status	Branch	Event	Actor	Time
Update info_clients.csv	Success	main	Pushed by Medcos	Medcos	2 hours ago
mise	Success	main	Pushed by Medcos	Medcos	yesterday
mise à jour	Success	main	Pushed by Medcos	Medcos	yesterday
mise à jour	Success	main	Pushed by Medcos	Medcos	yesterday
mise à jour	Success	main	Pushed by Medcos	Medcos	yesterday
mise à jour	Success	main	Pushed by Medcos	Medcos	yesterday
Update data_test.csv	Success	main	Pushed by Medcos	Medcos	2 days ago
mise à jour	Success	main	Pushed by Medcos	Medcos	2 days ago
Update main.py	Success	main	Pushed by Medcos	Medcos	3 days ago

Watermark: Activer Wi-Fi Accédez aux...



# ANALYSE DE DATA DRIFT

## Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

18  
Columns

3  
Drifted Columns

0.167  
Share of Drifted Columns

- ❑ Aucune data drift détectée sur le jeu de données
- ❑ Pour garantir la fiabilité du modèle, il est essentiel de :
  - ✓ **Monitoring continu** : Surveiller régulièrement les performances du modèle sur de nouvelles données.
  - ✓ **Retraîner régulièrement le modèle** : Mettre à jour le modèle avec de nouvelles données pour l'adapter aux changements.





## DÉMO DU DASHBOARD DÉPLOYÉ

---

Le Dashboard est composé de trois (3) fenêtres :

- ✓ Prédiction éligibilité.
- ✓ Interprétation des résultats
- ✓ Analyse de Drift





# CONCLUSION

---

L'algorithme mis en place va sans doute permettre d'évaluer en toute transparence la probabilité de faillite d'un client.



Le drift des données est un défi majeur en machine learning. Pour garantir la fiabilité du modèle, il est essentiel de mettre en place des mécanismes de détection et de gestion du drift.



En combinant des techniques de monitoring, de retraitement des données et d'apprentissage continu, il est possible de construire un modèle robuste et adaptatif.



MERCI

**OPENCLASSROOM**