

# Projet 7 : Implémentez un modèle de scoring

22/11/2024

Soukaina GUAOUA ELJADDI

Parcours Data Scientist  
OpenClassrooms

# Plan:

- ❑ Problématique et présentation du jeu de données
- ❑ Démarche de la Modélisation
- ❑ Pipeline de déploiement
- ❑ Analyse de data drift
- ❑ Exemple d'un scoring client via appel à l'API sur le Cloud

# Problématique

**Contexte :** Société financière "**Prêt à dépenser**" qui propose des crédits à la consommation.

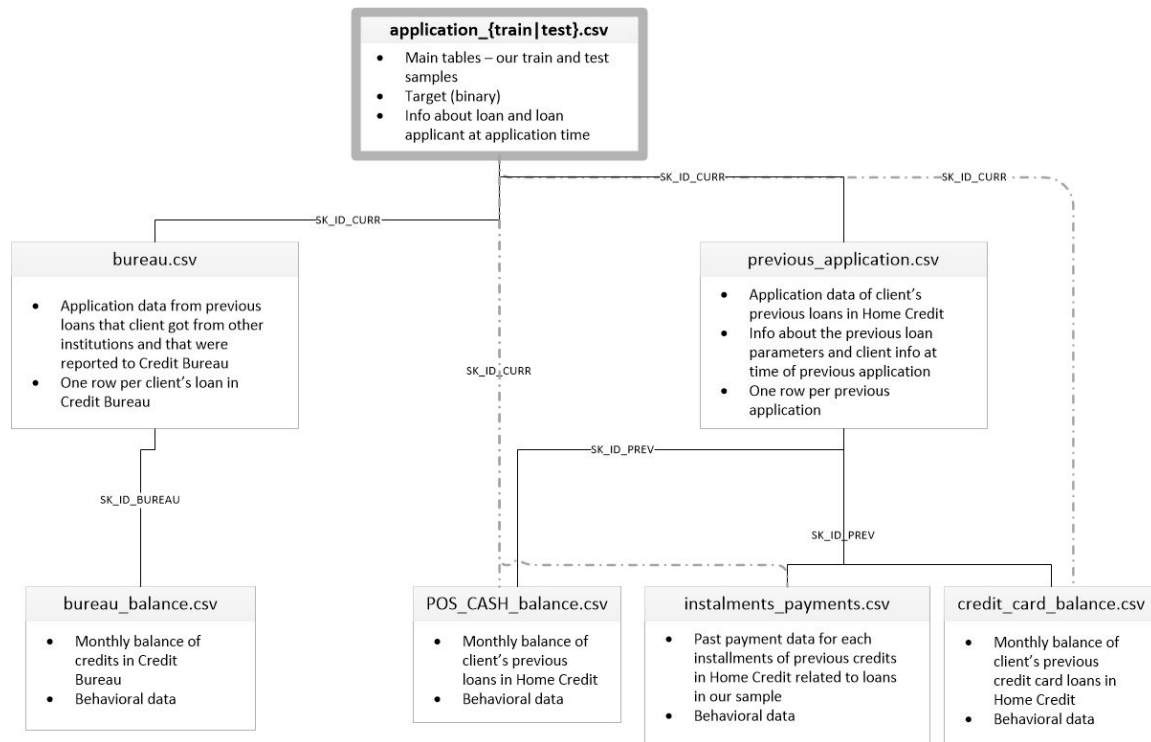
## **Objectif :**

- Mettre en œuvre un outil de “**scoring crédit**” pour calculer la **probabilité** qu’un client rembourse son **crédit**.
- Développer un **algorithme de classification** (**crédit accordé ou refusé**).

## **Missions :**

- Construire un **modèle de scoring** qui prédit la **probabilité de faillite** d'un client de façon automatique.
- Utiliser un **Dashboard** pour comprendre le **score attribué**

# Présentation du jeu de données



- application\_{train|test}.csv

- bureau.csv

- balance\_bureau.csv

- Solde\_en\_espèces\_POS.csv

- solde\_de\_la\_carte\_de\_crédit.csv

- application\_precedente.csv

- versements\_paiements.csv

- AccueilCredit\_columns\_descri

**TARGET : 0** (pas de problème de remboursement)

**TARGET : 1** (défaut de remboursement)

n.csv

# Présentation du jeu de données

## Préparation dataset modélisation (1):

- Encodage avec One-Hot Encoding
- Création de nouvelles features
- Agrégation des lignes -> {min, max, mean, size, sum, var}
- Fusion de tous les csv en une seule dataframe "df"
- Suppression des colonnes à valeur unique
- Remplacement des valeurs infinis par des NaN

# Présentation du jeu de données

## Préparation dataset modélisation (2):

- Suppression des colonnes avec plus de 50% de valeurs manquantes
- Séparation du jeu de données: df(356250, 773(ID+TARGET+771 features))
- **Train shape: (307506, 773)**
- **Test shape: (48744, 773)**
- Suppression des features corrélés à plus de 80%
- Extraction des 100 features importantes (Random Forest)

# Démarche de la modélisation

**Régression logistique** : modèle linéaire utilisé pour la classification binaire ou multinomiale. Il prédit la probabilité d'appartenance à une classe en appliquant une fonction sigmoïde ou softmax à une combinaison linéaire des caractéristiques.

**Hyperparamètres principaux :**

- **C**: Inverse de la régularisation (plus petite valeur = régularisation plus forte).
- **solver**: Méthode d'optimisation (e.g., 'liblinear', 'lbfgs', 'saga').
- **penalty**: Type de régularisation ('l1', 'l2', 'elasticnet' ou None).
- **max\_iter**: Nombre maximal d'itérations pour la convergence.

# Démarche de la modélisation

**Random Forest** : Un ensemble d'arbres de décision entraînés sur des sous-échantillons du dataset avec une moyenne (ou vote majoritaire) pour améliorer la robustesse et réduire le surapprentissage.

- **Hyperparamètres principaux :**
  - `n_estimators`: Nombre d'arbres dans la forêt.
  - `max_depth`: Profondeur maximale des arbres (évite le surapprentissage).
  - `min_samples_split`: Nombre minimal d'échantillons requis pour diviser un nœud.
  - `min_samples_leaf`: Nombre minimal d'échantillons requis dans une feuille.
  - `max_features`: Nombre maximal de caractéristiques considérées pour une division (e.g., `'sqrt'`, `'log2'`, ou un entier).



# Démarche de la modélisation

**LightGBM** : Un algorithme de boosting basé sur des arbres de décision, optimisé pour la vitesse et les grandes quantités de données en utilisant des histogrammes pour le regroupement des caractéristiques.

**Hyperparamètres principaux :**

- `num_leaves`: Nombre maximal de feuilles par arbre.
- `learning_rate`: Taux d'apprentissage pour ajuster la contribution de chaque arbre.
- `n_estimators`: Nombre d'arbres ou d'itérations de boosting.
- `max_depth`: Profondeur maximale des arbres.
- `min_data_in_leaf`: Nombre minimum de données dans une feuille.
- `boosting_type`: Type de boosting (`'gbdt'`, `'dart'`, `'goss'`).

# Démarche de la modélisation

**Dummy classifier** : Un modèle simple qui génère des prédictions basées sur des règles de base (e.g., prédire la classe majoritaire, aléatoire ou proportionnelle aux fréquences des classes). Il sert de référence pour évaluer les performances d'autres modèles.

**Hyperparamètres principaux :**

- **strategy**: Stratégie de prédiction ('most\_frequent', 'stratified', 'uniform', ou 'constant').
- **constant**: Valeur constante à prédire (utilisée uniquement si **strategy**='constant').

# Démarche de la modélisation

## Protocole :

- Traitement des valeurs infinis (NaN)
- Traitement des valeurs manquantes (Moyenne)
- Standardisation des données d'entraînement (StandardScaler)
- Séparation des données en train/validation/test
- Traitement du déséquilibre des classes avec SMOTE (crée artificiellement de nouveaux exemples pour les classes minoritaires en utilisant une approche d'interpolation. (92% de 0 et 8% de 1))
- Optimisation des hyperparamètres avec GridSearchCV

# Démarche de la modélisation

## Métriques d'évaluation :

**AUC** (Area Under the Curve) : Indique la capacité du modèle à distinguer les classes.

**F1 Score** : Moyenne harmonique entre la précision (exactitude des prédictions positives) et le rappel (proportion des vrais positifs détectés).

**Accuracy** : Proportion de prédictions correctes parmi toutes les prédictions.

**Fit time** : Temps nécessaire pour entraîner le modèle sur les données d'entraînement.

# Démarche de la modélisation

## Métriques d'évaluation :

**Predict time** : Temps nécessaire au modèle pour effectuer une prédiction sur les nouvelles données.

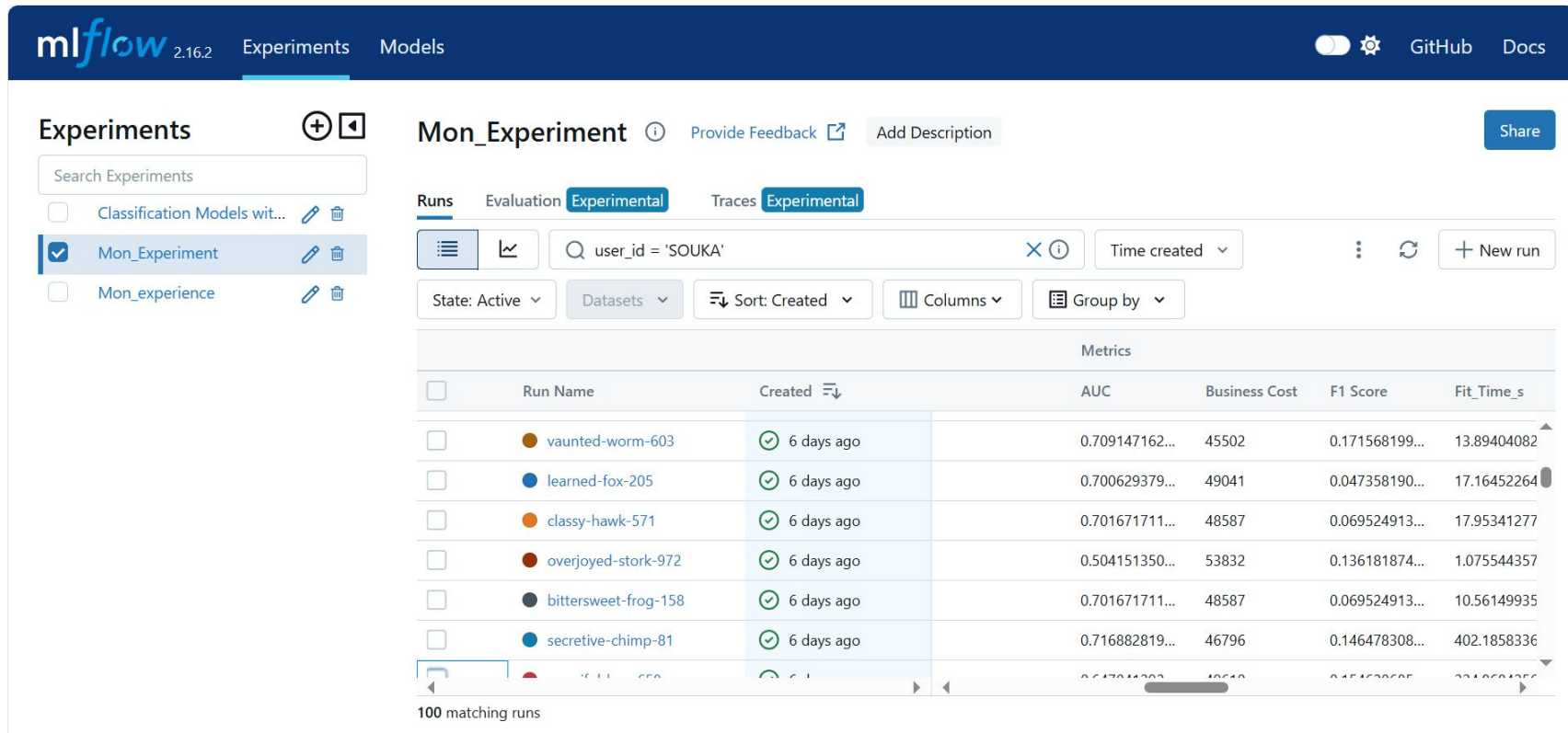
**Business cost** : Mesure le coût global associé aux erreurs de prédiction (faux positifs et faux négatifs) dans un contexte métier.

Business score = Somme( $10 \cdot \text{FN} + \text{FP}$ )

- FN = Faux Négatifs | FP = Faux Positifs

# Présentation des résultats

Visualisation du tracking via Mlflow (Vue d'ensemble)



# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

### Comparing 4 Runs from 1 Experiment

Visualizations

Parallel Coordinates Plot

Scatter Plot

Box Plot

Contour Plot

Parameters:

classifier\_\_C X

classifier\_\_class\_weight X

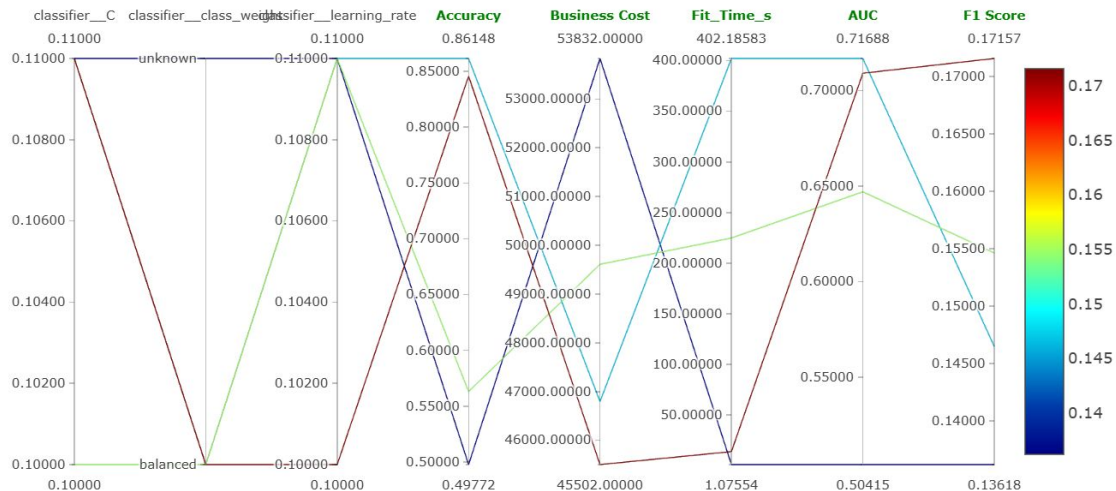
classifier\_\_learning\_rate X

Metrics:

Accuracy X Business Cost X

Fit\_Time\_s X AUC X F1 Score X

Clear All



# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

### ▼ Run details

Run ID:	<a href="#">bcb5cffcb2c54410ae780c794d30...</a>	<a href="#">33612786223a433ab4f89e059ed1...</a>	<a href="#">591db3f0d9bb46bdbbc9207c2709...</a>	<a href="#">8cc77bec940348e6b2a7c9613c72...</a>
Run Name:	vaunted-worm-603	secretive-chimp-81	merciful-hen-658	overjoyed-stork-972
Start Time:	2024-11-12 12:05:18	2024-11-12 11:43:04	2024-11-12 11:39:13	2024-11-12 11:50:12
End Time:	2024-11-12 12:05:40	2024-11-12 11:49:53	2024-11-12 11:43:04	2024-11-12 11:50:20
Duration:	21.9s	6.8min	3.9min	7.5s

### ▼ Parameters



# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

▼ Parameters

☐ Show diff only

classifier__C	0.1		
classifier__class_weight	balanced	balanced	balanced
classifier__learning_rate	0.1		
classifier__max_depth	10	10	
classifier__max_iter			100
classifier__min_samples_split		10	
classifier__n_estimators	50	100	
classifier__num_leaves	20		
classifier__solver			liblinear
classifier__strategy			stratified

# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

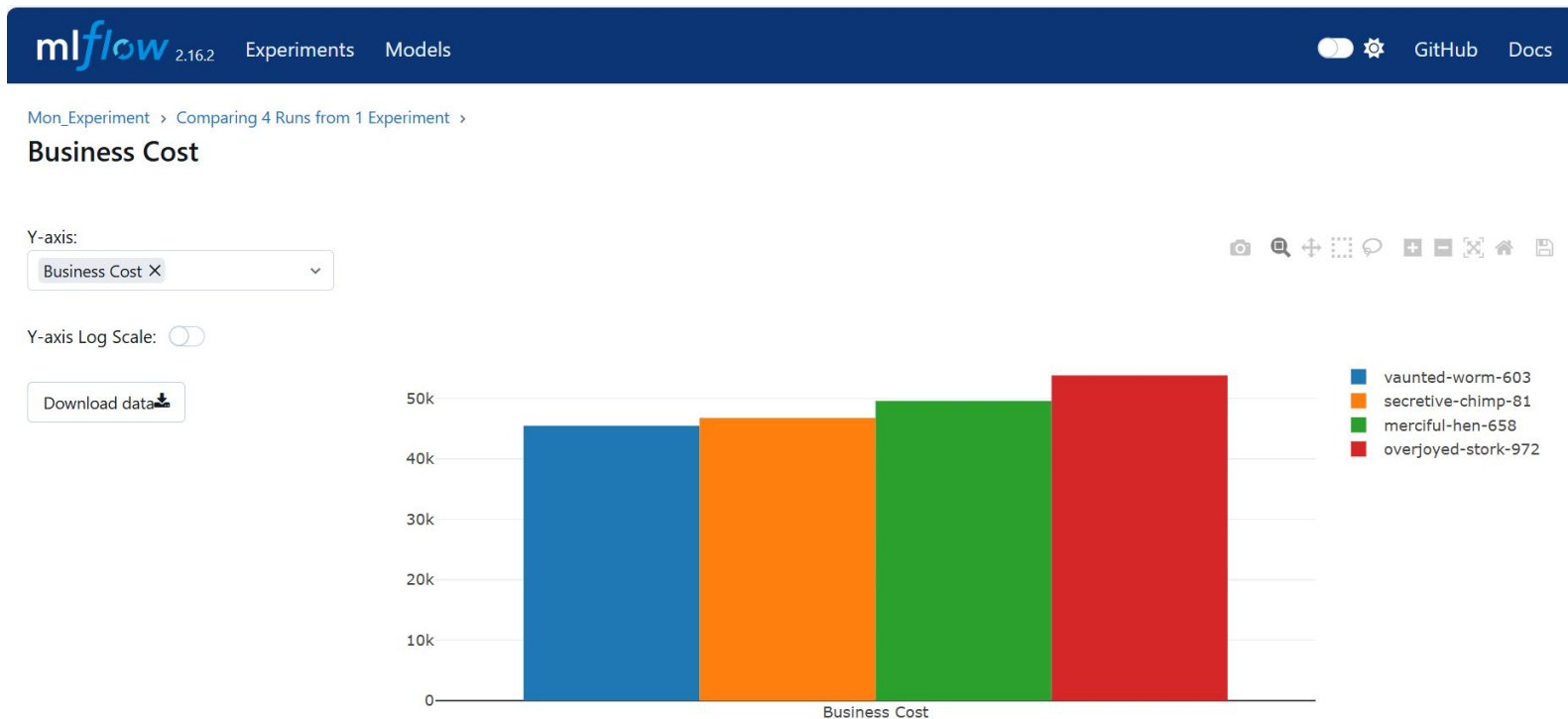
▼ Metrics

☐ Show diff only

AUC	0.709	0.717	0.647	0.504
Accuracy	0.846	0.861	0.563	0.498
Business Cost	45502	46796	49610	53832
F1 Score	0.172	0.146	0.155	0.136
Fit_Time_s	13.89	402.2	224.9	1.076
Predict_Time_s	0.049	0.428	0.009	0.004

# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs



# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

Comparing 4 Runs from 1 Experiment

Visualizations

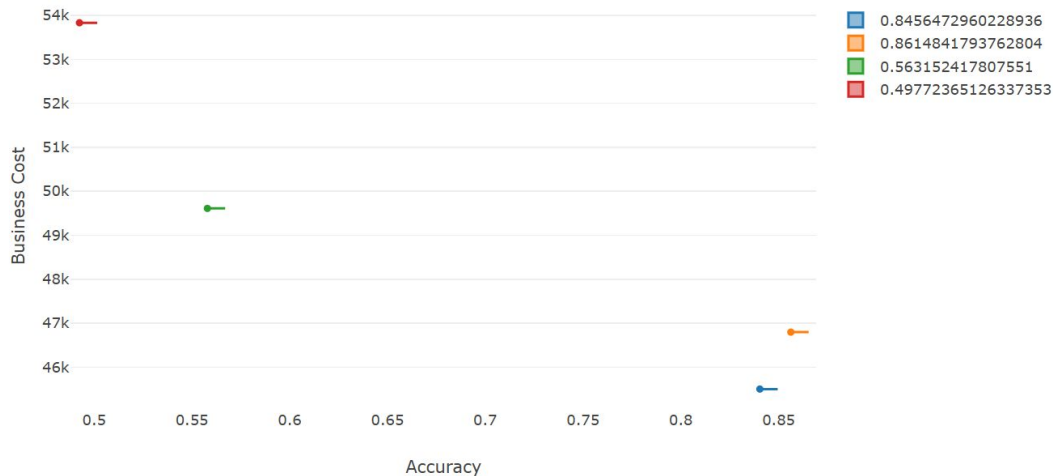
Parallel Coordinates Plot Scatter Plot **Box Plot** Contour Plot

X-axis:

Accuracy

Y-axis:

Business Cost



# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

Mon\_Experiment >

vaunted-worm-603

Register model

Overview

Model metrics

System metrics

Artifacts

LightGBM

MLmodel

conda.yaml

model.pkl

python\_env.yaml

requirements.txt

LightGBM

Register model

Path: file:///C:/Users/SOUKA/Desktop/P7\_mlflow\_logs/142011389048118064/bcb5cfc2c54410ae780c794d303f5b/artifacts/LightGBM

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register it to the model registry](#) to version control

Model schema

Input and output schema for your model. [Learn more](#)

Name

Type

No schema. See [MLflow docs](#) for how to include input and output schema with your model.

Validate the model before deployment

Run the following code to validate model inference works on the example payload, prior to deploying it to a serving endpoint

```
from mlflow.models import validate_serving_input

model_uri = 'runs:/bcb5cfc2c54410ae780c794d303f5b/LightGBM'

# The logged model does not contain an input_example.
# Manually generate a serving payload to verify your model prior
# to deployment.
from mlflow.models import convert_input_example_to_serving_input

# Define INPUT_EXAMPLE via assignment with your own input example
to the model
```

# Présentation des résultats

## Visualisation du tracking via mlflow : Comparaison entre différentes runs

Mon\_Experiment >  
**vaunted-worm-603** ⋮ [Register model](#)

Overview   Model metrics   System metrics   **Artifacts**

▼ **LightGBM**

- MLmodel
- conda.yaml
- model.pkl
- python\_env.yaml
- requirements.txt

**LightGBM** [Register model](#)

Path: file:///C:/Users/SOUKA/Desktop/P7\_mlflow\_logs/142011389048118064/bcb5cffcb2c54410ae780c794d303f5b/artifacts/LightGBM [🔗](#)

```
serving_payload =  
convert_input_example_to_serving_input(INPUT_EXAMPLE)  
  
# Validate the serving payload works on the model  
validate_serving_input(model_uri, serving_payload)
```

**Make Predictions**

Predict on a Pandas DataFrame:

```
import mlflow  
logged_model = 'runs:/bcb5cffcb2c54410ae780c794d303f5b/LightGBM' 🔗  
  
# Load model as a PyFuncModel.  
loaded_model = mlflow.pyfunc.load_model(logged_model)  
  
# Predict on a Pandas DataFrame.  
import pandas as pd  
loaded_model.predict(pd.DataFrame(data))
```

Predict on a Spark DataFrame:

# Présentation des résultats

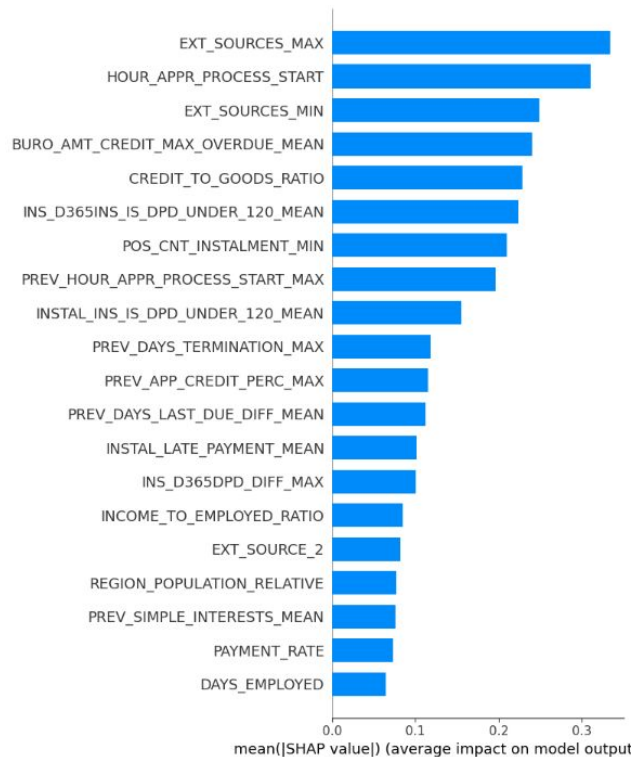
## Analyse des résultats

	Model	Best AUC	Best F1 Score	Accuracy	Business Cost	Fit Time (s)	Predict Time (s)
0	Logistic Regression	0.647041	0.154621	0.563152	49610	224.868426	0.009357
1	Random Forest	0.716883	0.146478	0.861484	46796	402.185834	0.427655
2	LightGBM	0.709147	0.171568	0.845647	45502	13.894041	0.049382
3	Dummy Classifier	0.504151	0.136182	0.497724	53832	1.075544	0.004001

Le modèle recommandé est **Lightgbm** : meilleur (business cost, Best F1 Score, Fit time)

Seuil optimal : 0.24, Coût métier minimum : 38010

# Présentation des résultats





# Pipeline de déploiement

Tests unitaire : Test de la fonction de prédiction de l'API (unittest)

Dépôt local

Git  
add  
commit push

API déploiement  
Render

Github  
Actions

Dashboard  
Streamlit cloud

# Pipeline de déploiement

Github

([https://github.com/SoukainaG/P7\\_API\\_d-ploiment](https://github.com/SoukainaG/P7_API_d-ploiment))

The screenshot shows the GitHub repository page for 'P7\_API\_d-ploiment' by user 'SoukainaG'. The repository is public and has 1 branch and 0 tags. The file list includes:

File/Folder	Description	Last Commit
.devcontainer	Added Dev Container Folder	3 days ago
.github/workflows	fichier yaml modifié	last week
__pycache__	Modifications dans app.py et test_app.py	last week
README.md	Ajout du fichier README.md	3 weeks ago
analyse_drift.py	modèle lightgbm modifié	last week
app.py	Modifications dans app.py et test_app.py	last week
data_R.xlsx	Ajout du data modif	yesterday
data_R1.xlsx	Ajout data modif	yesterday
df_final_cleaned_S.xls	Ajout du jeu de données df_final_cleaned_S.xls	2 days ago
df_final_cleaned_S.xlsx	jeu de données extension modif	2 days ago

The right sidebar shows the repository's metadata: No description, website, or topics provided. It also lists Readme, Activity, 0 stars, 1 watching, and 0 forks. The Releases and Packages sections indicate no releases or packages published, with links to create a new release or publish a first package. The Languages section is also visible.

# Pipeline de déploiement

← → ↻ share.streamlit.io/deploy

soukainag ▾ My apps My profile Explore Discuss ↗

← Back

## Deploy an app

Repository ⓘ [Paste GitHub URL](#)

SoukainaG/P7\_API\_d-ploiment

Branch

master

Main file path

scoring\_interface.py

App URL (optional)

p7apid-ploiment-7penemqyxiuuyw6jr644us .streamlit.app

# Analyse de data drift

## Dataset Drift

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

**773**

Columns

**117**

Drifted Columns

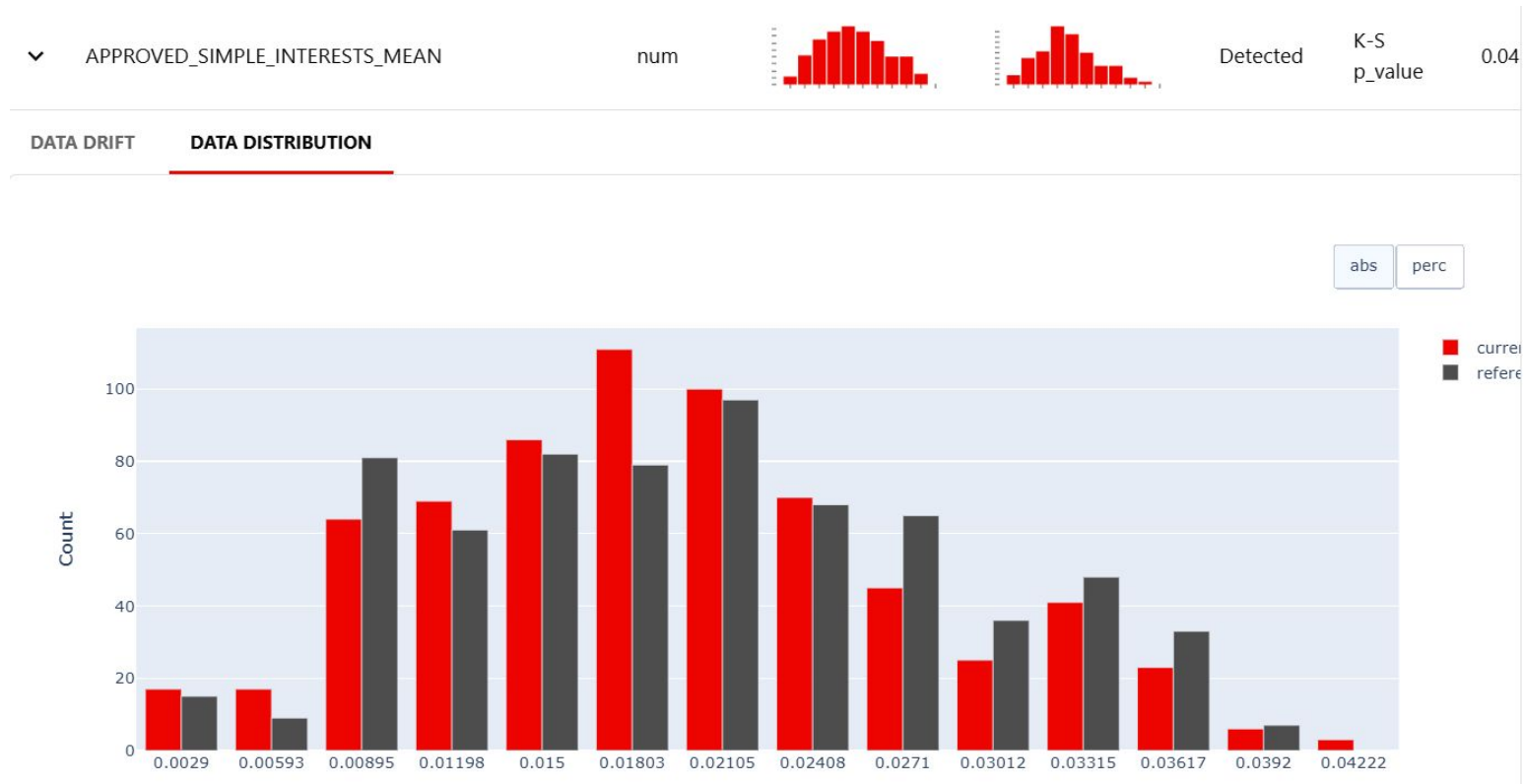
**0.151**

Share of Drifted Columns

## Data Drift Summary

Drift is detected for 15.136% of columns (117 out of 773).

# Analyse de data drift



# Exemple d'un scoring client via appel à l'API sur le Cloud

<https://p7apid-ploiement-e26adgmbbclbb5rzd4ujfd.streamlit.app/>