

# PROJET 7

## IMPLÉMENTEZ UN MODÈLE DE SCORING

Xavier Parisot

## **Introduction**

**|01 – Présentation du jeu de données**

**|02 – Présentation du feature engineering**

**|03 – Démarche de modélisation**

**|04 – Pipeline de déploiement**

**|05 - Analyse de Data Drift**

**|06 - API et Dashboard**

**|Conclusion**

# INTRODUCTION

## Rappel de la problématique :

L'entreprise souhaite **mettre en œuvre un outil de “scoring crédit” pour calculer la probabilité** qu'un client rembourse son crédit, puis classifie la demande en crédit accordé ou refusé. Elle souhaite donc développer un **algorithme de classification** en s'appuyant sur des sources de données variées .

**Développer un dashboard interactif** pour que les chargés de relation client puissent expliquer de façon transparente les décisions d'octroi de crédit,

Attendus :

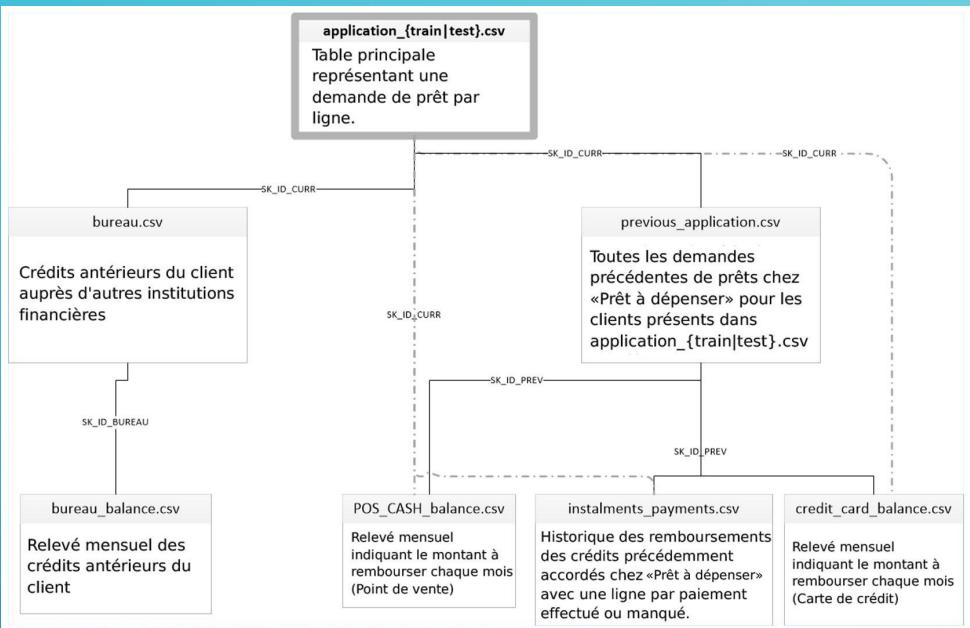
Développement  
d'un algorithme de  
classification

Construction du  
modèle de  
prédiction

Construction d'un  
dashboard  
interactif

Mise en production  
du modèle de  
prédiction et de  
l'api

Mise en place d'un  
pipeline de  
déploiement



Nom des Dataframes	Nb lignes	Nb colonnes	Taux remplissage moyen	Doublons
application_test	48744	121	76.19%	0
application_train	307511	122	75.60%	0
bureau_balance	27299925	3	100.00%	0
bureau	1716428	17	86.50%	0
credit_card_balance	3840312	23	93.35%	0
HomeCredit_columns_description	219	5	87.85%	0
installments_payments	13605401	8	99.99%	0
POS_CASH_balance	10001358	8	99.93%	0
previous_application	1670214	37	82.02%	0
sample_submission	48744	2	100.00%	0

# PRESENTATION DU JEU DE DONNÉES

# ANALYSE DES DONNÉES

## ANALYSE DE LA TARGET

Distribution de la TARGET



La classification est binaire :

- 0 : Clients ne présentant pas de risques de défaut
- 1 : Clients présentant un risque de défaut

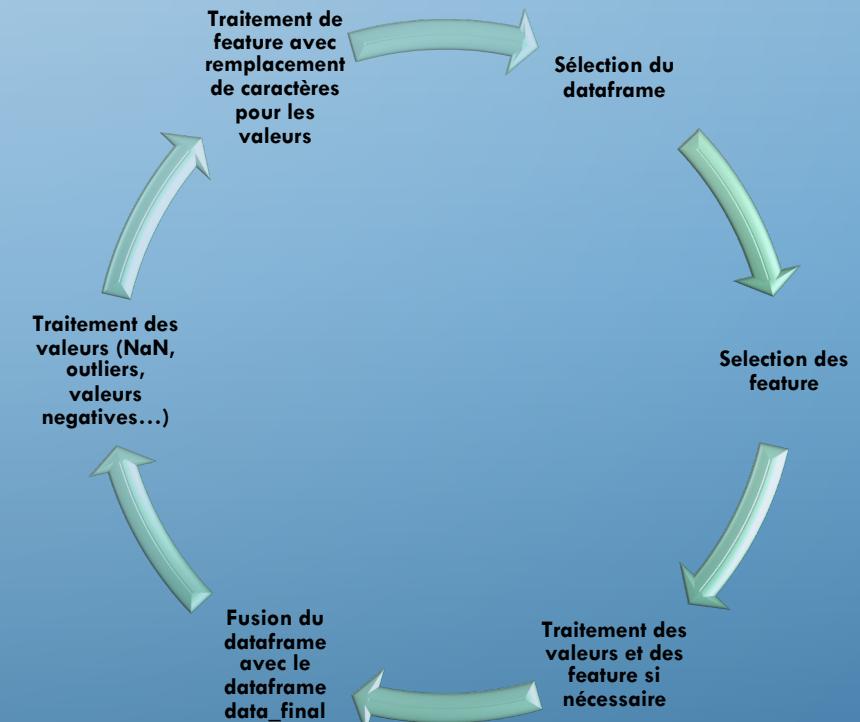
Nous avons un déséquilibre de classe important :

- 92 % de clients sains
- 8 % de clients présentant un défaut de paiement

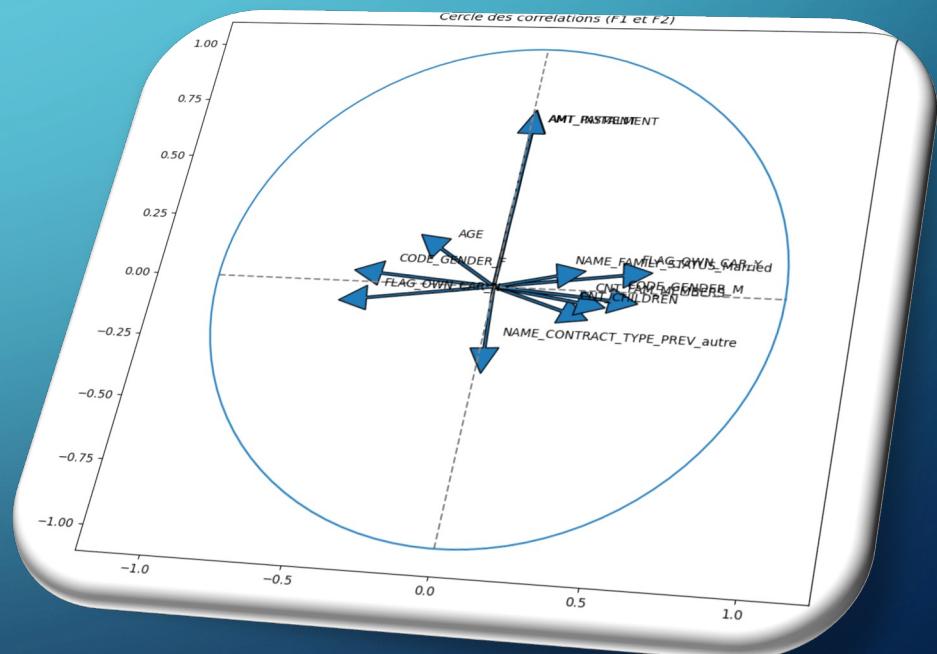
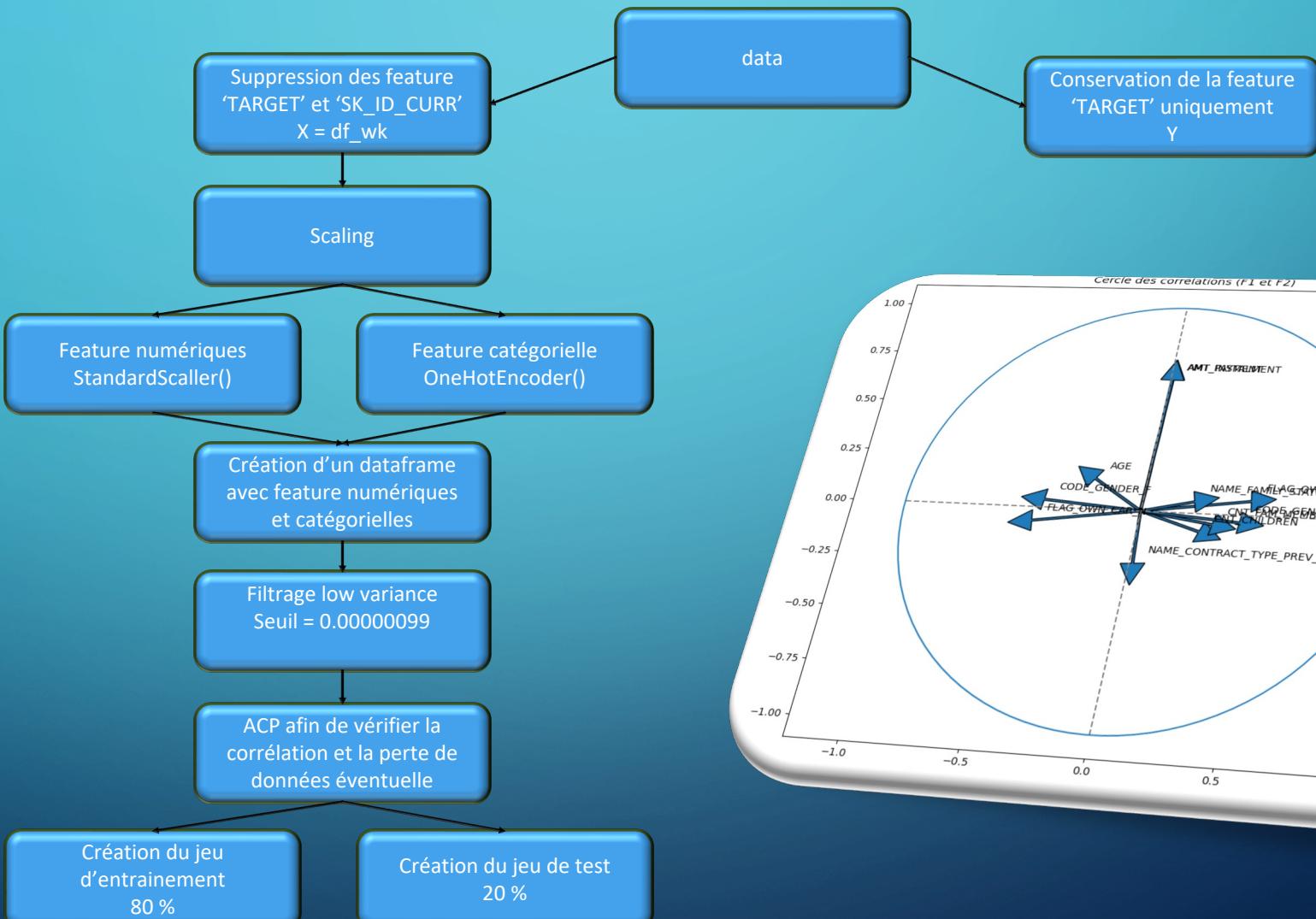
# FEATURE ENGINEERING

## CRÉATION DU DATAFRAME FINAL

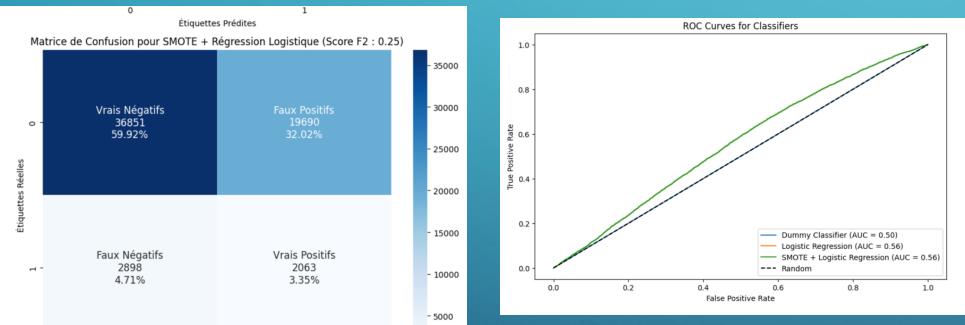
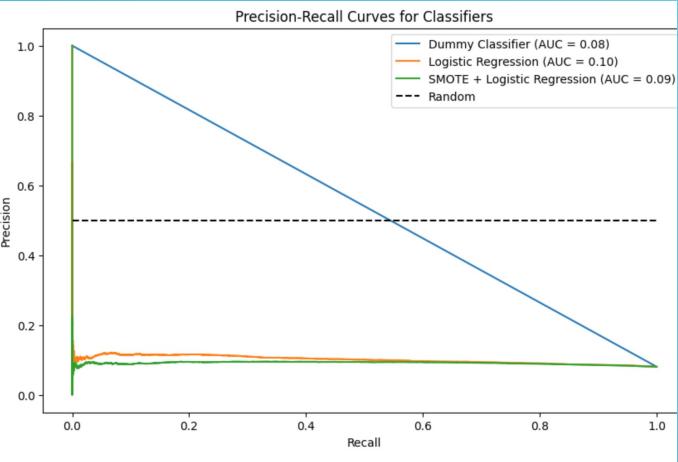
- Nous avons fait le choix de ne pas utiliser un kernel kaggle afin de sélectionner les features les plus cohérentes avec notre problématique. Le traitement fut le suivant :



# FEATURE ENGINEERING



# DÉMARCHE DE MODÈLISATION MODÈLE DE RÉFÉRENCE



## Dummy Regressor, Logistic Regression

- Création des classifiers avec SMOTE

## Courbes ROC

## Courbes Recall

## Matrices de confusion

- Basées sur le score F2

# DÉMARCHE DE MODÈLISATION MODÈLES

*Comparaison des différents modèles sans hyperparamètres*

*Comparaison des différents modèles avec les hyperparamètres*

*Recherche du seuil pour le score personnalisé pour chacun des modèles*

*Choix du modèle*

*Explication du choix du modèle*

# DÉMARCHE DE MODÈLISATION MODÈLES

- **Configuration de l'environnement de suivi :** Utilisation de MLflow pour suivre les métriques, les hyperparamètres et les modèles.
- **Prétraitement et équilibrage des données :** Utilisation de SMOTE pour gérer le déséquilibre des classes.
- **Sélection du modèle :** Utilisation de différents algorithmes de classification.
- **Optimisation des hyperparamètres :** Utilisation de GridSearchCV pour trouver les meilleurs hyperparamètres pour chaque modèle.
- **Évaluation des modèles :** Calcul et suivi de plusieurs métriques pour évaluer la performance des modèles.
- **Enregistrement du modèle :** Sauvegarde des modèles entraînés pour une utilisation ultérieure.

```
F1-Score: 0.6686774413840201
AUC-ROC: 0.6557770318072234
Cohen_Kappa: 0.1223955085036631
Accuracy: 0.6686774413840201

Fitting 3 folds for each of 4 candidates, totalling 12 fits
/Users/xparisot/anaconda3/envs/OC/lib/python3.10/site-packages/_distutils_hack/_init_.py
warnings.warn(
/Users/xparisot/anaconda3/envs/OC/lib/python3.10/site-packages/_distutils_hack/_init_.py
warnings.warn("Setuptools is replacing distutils.")
METRICS FOR RandomForestClassifier _____
Precision: 0.2035475428903751
Recall: 0.1411005845595646
F1-Score: 0.8861825631686775
AUC-ROC: 0.54632893078989
Cohen_Kappa: 0.10773383656862534
Accuracy: 0.8861825631686775

Fitting 3 folds for each of 8 candidates, totalling 24 fits
/Users/xparisot/anaconda3/envs/OC/lib/python3.10/site-packages/_distutils_hack/_init_.py
warnings.warn(
/Users/xparisot/anaconda3/envs/OC/lib/python3.10/site-packages/_distutils_hack/_init_.py
warnings.warn("Setuptools is replacing distutils.")
METRICS FOR XGBClassifier
```

```
def evaluate_models():
    for model_name, params in models.items():
        print(f"Evaluating {model_name}...")
        for param_name, param_values in params.items():
            for param_value in param_values:
                print(f"\t{param_name}: {param_value}")
                # Create pipeline
                pipeline = make_pipeline(SMOTE(), eval_func)
                # Set parameters
                pipeline.set_params(**{f'{model_name}_{param_name}': param_value})
                # Fit and evaluate
                eval_func(pipeline, X_train, y_train, X_val, y_val)
```

Run Name      Created      Dataset      Duration      Source      Models      Metrics

							Best Score	F2-Score	Training Time
LightGBM	1 day ago	-	15.3min	ipykern...	pyfunc	0.04	0.919	917.8	
XGBClassifier	1 day ago	-	6.9min	ipykern...	pyfunc	0.383	0.669	414	
RandomForestClassifier	1 day ago	-	4.8min	ipykern...	pyfunc	0.153	0.886	285.8	
LinearSVC	1 day ago	-	14.6s	ipykern...	pyfunc	0.383	0.669	12.9	
LightGBM	1 day ago	-	19.1min	ipykern...	pyfunc	0.039	0.919	900.7	
XGBClassifier	1 day ago	-	6.7min	ipykern...	pyfunc	0.384	0.67	399.2	
RandomForestClassifier	1 day ago	-	4.8min	ipykern...	pyfunc	0.159	0.883	283.6	
LinearSVC	1 day ago	-	14.8s	ipykern...	pyfunc	0.383	0.67	12.87	
LightGBM	1 day ago	-	14.9min	ipykern...	pyfunc	0.039	0.918	889.4	
XGBClassifier	1 day ago	-	6.6min	ipykern...	pyfunc	0.383	0.669	395.1	
RandomForestClassifier	1 day ago	-	3.5min	ipykern...	pyfunc	0.147	0.886	205.8	
LinearSVC	1 day ago	-	14.0s	ipykern...	pyfunc	0.383	0.66	12.25	
LightGBM	1 day ago	-	15.1min	ipykern...	pyfunc	0.04	0.918	901.9	
XGBClassifier	1 day ago	-	6.6min	ipykern...	pyfunc	0.384	0.669	396.2	
RandomForestClassifier	1 day ago	-	4.7min	ipykern...	pyfunc	0.151	0.883	279.7	
LinearSVC	1 day ago	-	14.0s	ipykern...	pyfunc	0.383	0.669	12.13	

Registered Models

Name	Latest version	Staging	Production	Created by	Last modified	Tags
best_lgb_model	Version 3	—	—		2023-09-08 09:21...	—
best_rfc_model	Version 4	—	—		2023-09-08 09:22...	—
best_svc_model	Version 3	—	—		2023-08-16 13:06...	—
best_xgb_model	Version 4	—	—		2023-09-08 09:22...	—

Run Name      Created      Dataset      Duration      Source      Version      Models      Metrics      Parameters

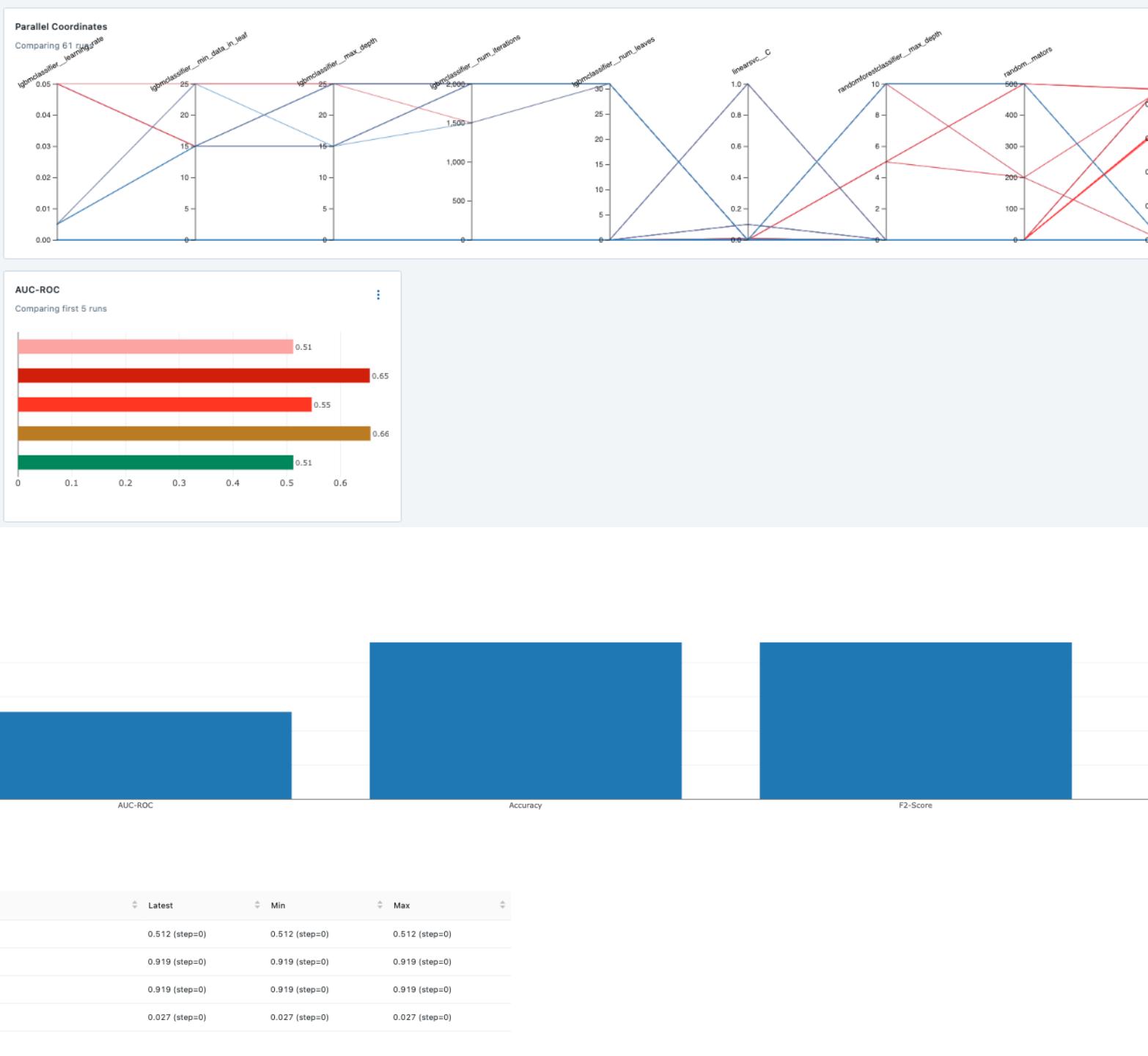
							custom_score	lgbmclassifier	lgbmclassifier	lgbmclassifier	lgbmclassifier
LGBM_with_Custom_Score	2 days ago	-	1.s	ipykern...	c1ca00	-	5.028	0.05	25	15	2000
LGBM_with_Custom_Score	2 days ago	-	1.7s	ipykern...	c1ca00	-	5.031	0.05	25	15	2000
LGBM_with_Custom_Score	5 days ago	-	1.8s	ipykern...	c1ca00	-	5.03	0.05	15	15	2000
LGBM_with_Custom_Score	27 days ago	-	1.8s	ipykern...	-	-	5.029	0.005	25	25	2000
LGBM_with_Custom_Score	28 days ago	-	1.8s	ipykern...	-	-	5.029	0.005	15	15	1500
LGBM_with_Custom_Score	28 days ago	-	1.8s	ipykern...	-	-	5.027	0.005	15	15	2000

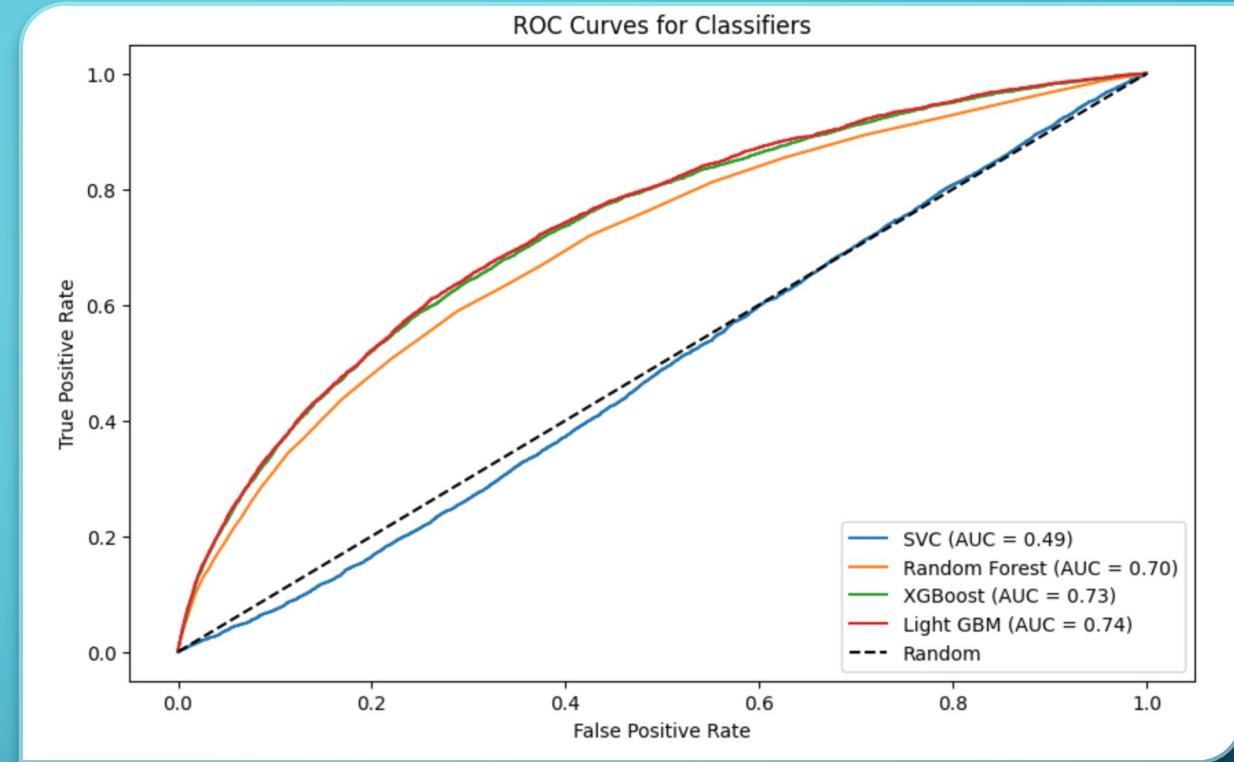
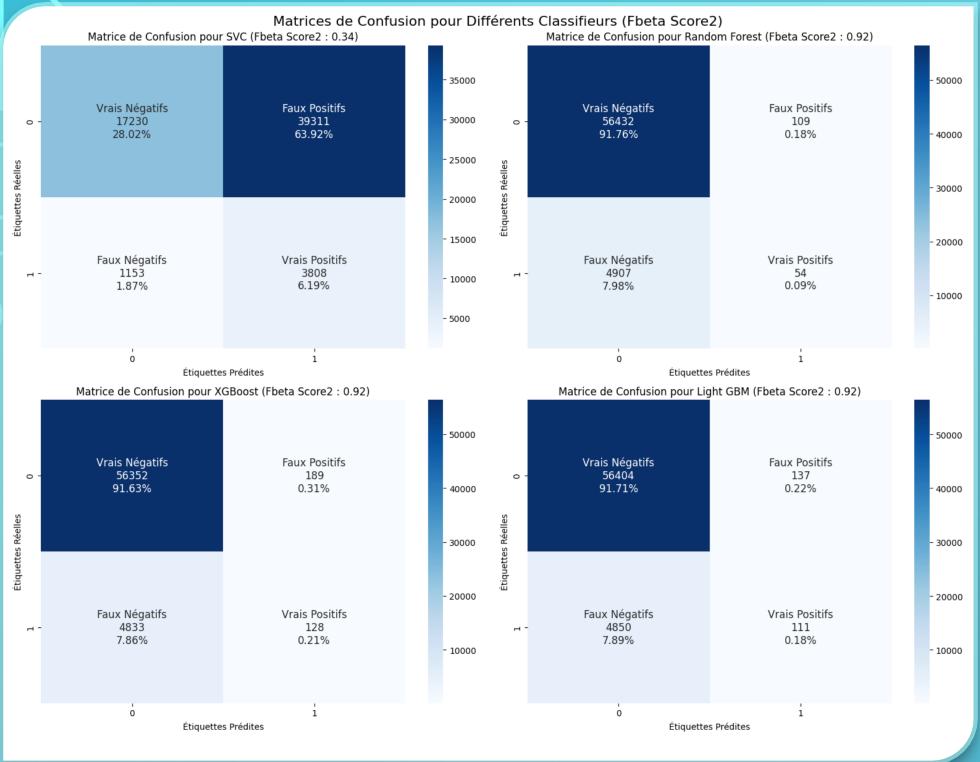
## DÉMARCHE DE MODÈLISATION MODÈLES TRACKING MLFLOW

ENREGISTREMENT ET SUIVI DES MODELES

# DÉMARCHE DE MODÈLISATION MODÈLES TRACKING MLFLOW

## METRICS ET HYPERPARAMETRES



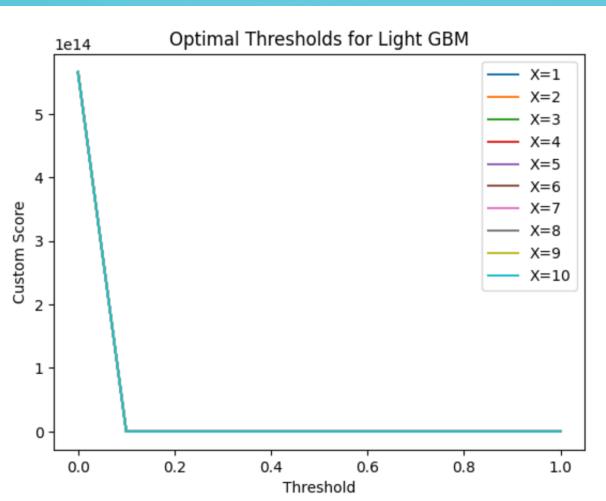


# DÉMARCHE DE MODÈLISATION MODÈLES AFFICHAGE DES RÉSULTATS

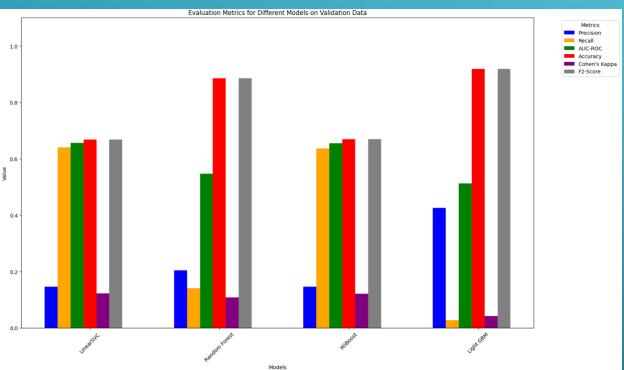
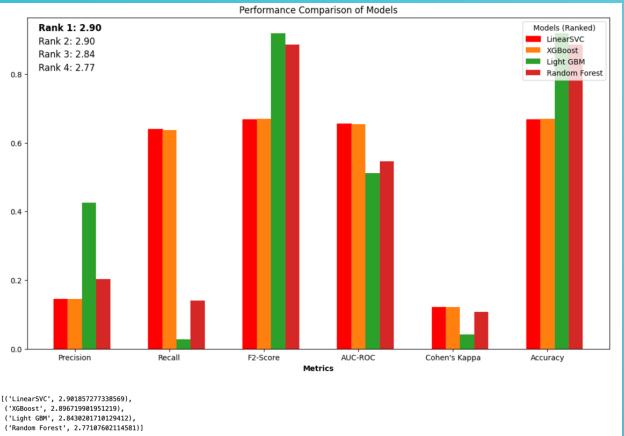
# DÉMARCHE DE MODÈLISATION MODÈLES

## CALCUL DU SEUIL DE SCORE PERSONNALISÉ

```
Best Values:  
    Model Best X Best Threshold Best Custom Score  
0  Random Forest     1        0.8          1.0  
1  XGBoost           1        0.9          1.0  
2  Light GBM         1        0.9          1.0  
  
All Optimal Values:  
    X Threshold Custom Score      Model  
0   1       0.0  5.65410e+14 Random Forest  
1   1       0.1  1.48653e+01 Random Forest  
2   1       0.2  3.22481e+00 Random Forest  
3   1       0.3  1.43962e+00 Random Forest  
4   1       0.4  1.11562e+00 Random Forest  
... ... ... ...  
325 10      0.6  1.00114e+01 Light GBM  
326 10      0.7  1.00036e+01 Light GBM  
327 10      0.8  1.00010e+01 Light GBM  
328 10      0.9  1.00000e+01 Light GBM  
329 10      1.0  1.00000e+01 Light GBM  
  
[330 rows x 4 columns]  
    Model Best X Best Threshold Best Custom Score  
0  Random Forest     1        0.8          1.0  
1  XGBoost           1        0.9          1.0  
2  Light GBM         1        0.9          1.0  
  
Best Values:  
    Model Best X Best Threshold Best Custom Score  
0  Random Forest     1        0.8          1.0  
1  XGBoost           1        0.9          1.0  
2  Light GBM         1        0.9          1.0
```



# DÉMARCHE DE MODÈLISATION CHOIX DU MODÈLE



## METRICS FOR LinearSVC

Precision: 0.14593477262287552  
Recall: 0.6403950816367667  
AUC-ROC: 0.6557770318072234  
Accuracy: 0.6686774413840201  
Cohen's Kappa: 0.1223955085036631  
F2-Score 0.6686774413840201

## METRICS FOR Random Forest

Precision: 0.2035475428903751  
Recall: 0.1411005845595646  
AUC-ROC: 0.54632893078989  
Accuracy: 0.8861825631686775  
Cohen's Kappa: 0.10773383656862534  
F2-Score 0.8861825631686775

## METRICS FOR XGBoost

Precision: 0.1455483395513795  
Recall: 0.6369683531546059  
AUC-ROC: 0.6544350794619352  
Accuracy: 0.6690839322298462  
Cohen's Kappa: 0.12160026532360613  
F2-Score 0.6690839322298462

## METRICS FOR Light GBM

Precision: 0.4253968253968254  
Recall: 0.027010683329973795  
AUC-ROC: 0.511904733256929  
Accuracy: 0.9185717537641053  
Cohen's Kappa: 0.04156442150100215  
F2-Score 0.9185717537641054

**LinearSVC:** rappel élevé mais précision très faible. Cela signifie qu'il est bon pour identifier les vrais positifs mais génère beaucoup de faux positifs. Score F2 donne plus de poids au rappel, le plus bas parmi les modèles.

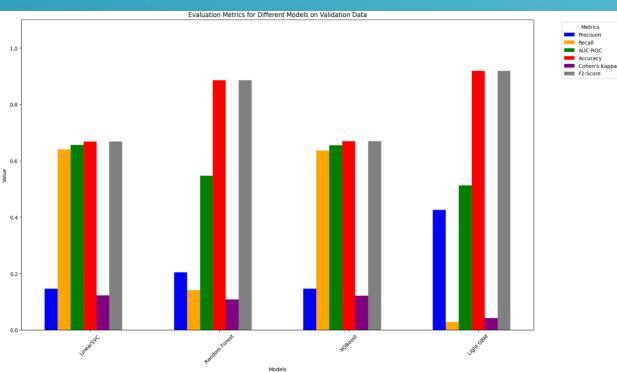
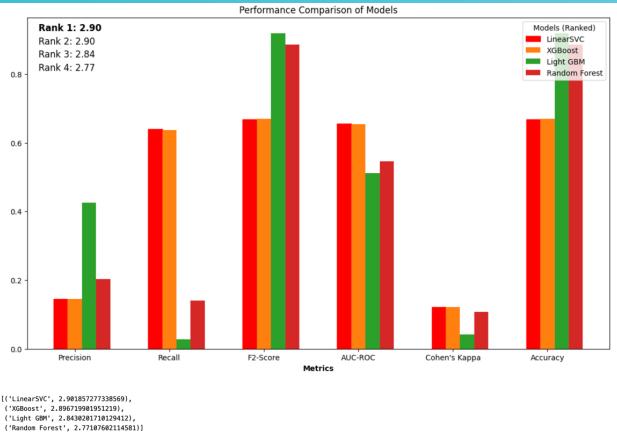
**Random Forest:** précision et un rappel relativement bas, mais excellente exactitude. F2 plus élevé parmi les modèles, il équilibre bien le rappel et la précision.

**XGBoost:** rappel élevé précision faible, similaire à LinearSVC. F2 bas.

**Light GBM:** excellente précision, très faible rappel. F2 le plus élevé, bon équilibre entre rappel et précision.

L'exactitude est la plus importante, Random Forest et Light GBM sont les meilleurs. Pour une raison de poids de modèle, je choisis Light GBM

# DÉMARCHE DE MODÈLISATION CHOIX DU MODÈLE



## METRICS FOR LinearSVC

Precision: 0.14593477262287552  
 Recall: 0.6403950816367667  
 AUC-ROC: 0.6557770318072234  
 Accuracy: 0.6686774413840201  
 Cohen's Kappa: 0.1223955085036631  
 F2-Score 0.6686774413840201

## METRICS FOR Random Forest

Precision: 0.2035475428903751  
 Recall: 0.1411005845595646  
 AUC-ROC: 0.54632893078989  
 Accuracy: 0.8861825631686775  
 Cohen's Kappa: 0.10773383656862534  
 F2-Score 0.8861825631686775

## METRICS FOR XGBoost

Precision: 0.1455483395513795  
 Recall: 0.6369683531546059  
 AUC-ROC: 0.6544350794619352  
 Accuracy: 0.6690839322298462  
 Cohen's Kappa: 0.12160026532360613  
 F2-Score 0.6690839322298462

## METRICS FOR Light GBM

Precision: 0.4253968253968254  
 Recall: 0.027010683329973795  
 AUC-ROC: 0.511904733256929  
 Accuracy: 0.9185717537641053  
 Cohen's Kappa: 0.04156442150100215  
 F2-Score 0.9185717537641054

**LinearSVC:** rappel élevé mais précision très faible. Cela signifie qu'il est bon pour identifier les vrais positifs mais génère beaucoup de faux positifs. Score F2 donne plus de poids au rappel, le plus bas parmi les modèles.

**Random Forest:** précision et un rappel relativement bas, mais excellente exactitude. F2 plus élevé parmi les modèles, il équilibre bien le rappel et la précision.

**XGBoost:** rappel élevé précision faible, similaire à LinearSVC. F2 bas.

**Light GBM:** excellente précision, très faible rappel. F2 le plus élevé, bon équilibre entre rappel et précision.

L'exactitude est la plus importante, Random Forest et Light GBM sont les meilleurs. Pour une raison de poids de modèle, je choisis Light GBM

# MODELE : METRIQUES UTILISEES

$$\text{Accuracy} = \frac{\text{Vrai positif} + \text{Vrai négatif}}{\text{Total}}$$

$$\text{Precision}(i) = \frac{\text{nb d'individus correctement attribués à la classe } i}{\text{nb d'individus attribués à la classe } i}$$

$$\text{Recall}(i) = \frac{\text{nb d'individus correctement attribués à la classe } i}{\text{nb d'individus appartenant à la classe } i}$$

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

$$AUC = \int_0^1 h(t_1) dt_1.$$

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)},$$

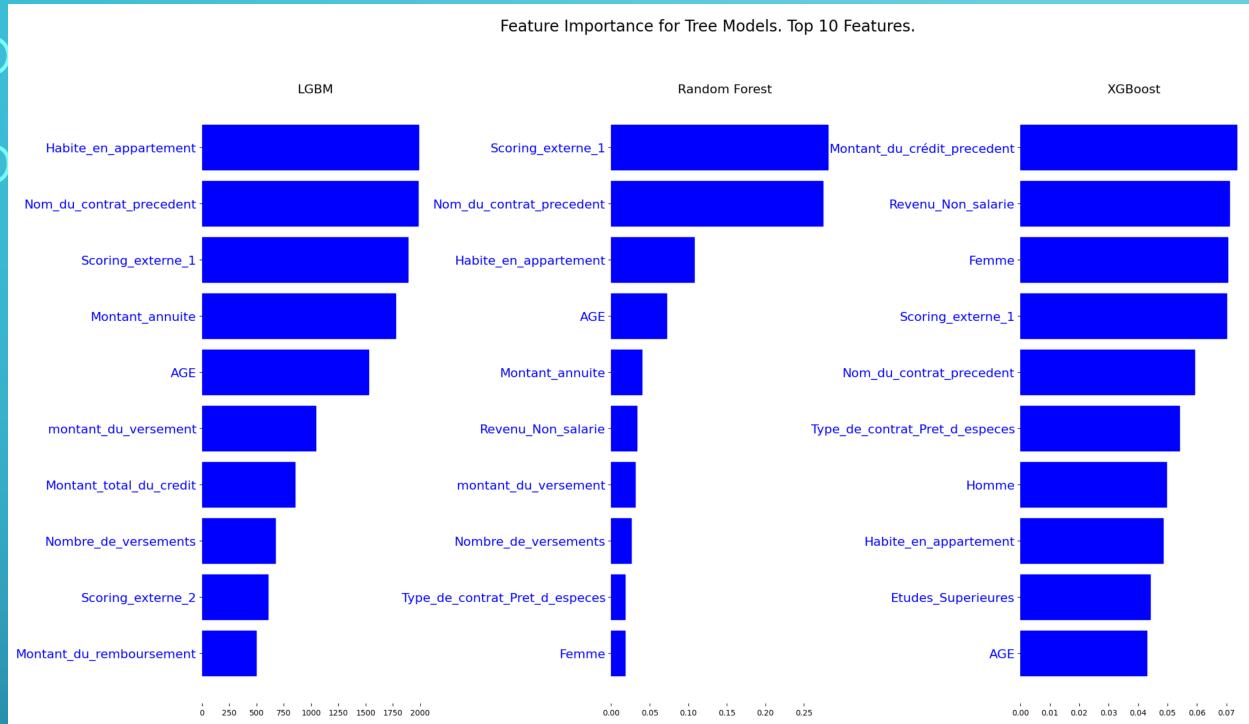


# MODÈLE : PROCESSUS DE RÉÉQUILIBRAGE

Pour la création du pipeline de prédiction :

Cette stratégie combine deux approches pour gérer le déséquilibre des classes :

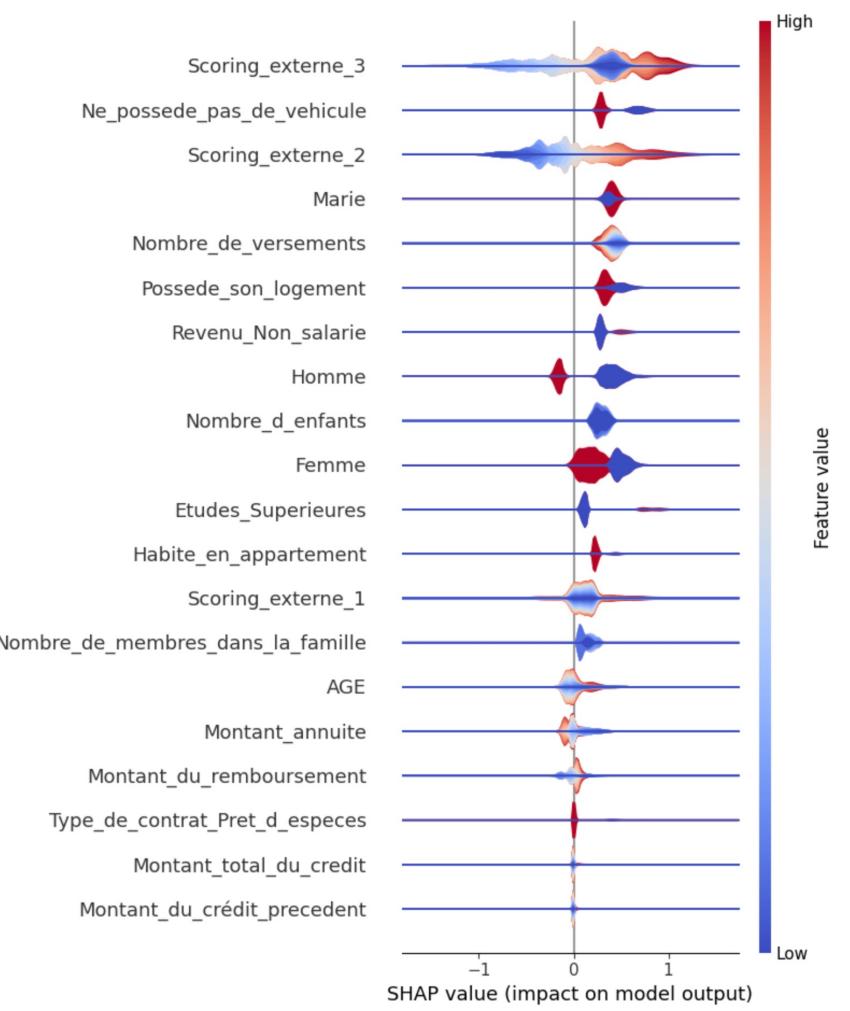
1. **Over-sampling avec SMOTE** : pour augmenter le nombre d'instances de la classe minoritaire.
  2. **Poids de classe dans le modèle** : pour donner plus d'importance à la classe minoritaire lors de l'entraînement du modèle.
- En combinant ces deux techniques, le code vise à créer un modèle qui est à la fois sensible à la classe minoritaire et performant en termes de métriques d'évaluation.



Les 5 caractéristiques les plus communes par ordre décroissant sont :

- Scoring\_externe\_1 : 3 fois
- Nom\_du\_contrat\_precedent : 3 fois
- AGE : 2 fois
- Montant\_annuite : 2 fois
- Habite\_en\_appartement : 2 fois

# FEATURE IMPORTANCE DES MODELES

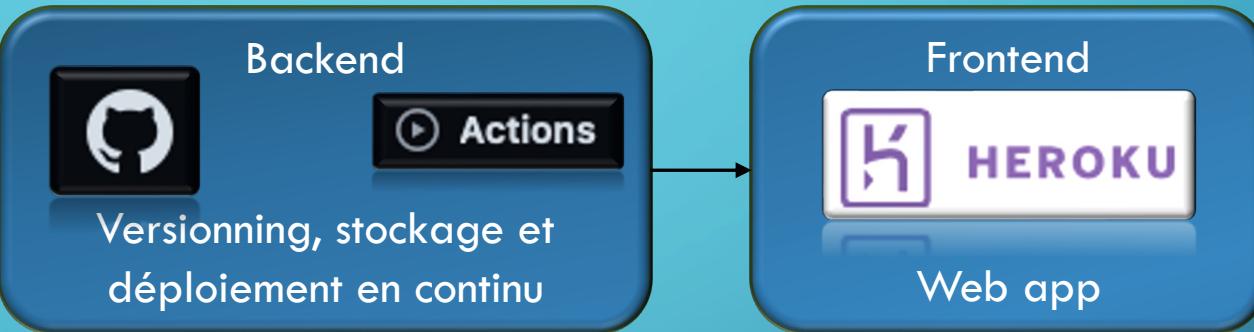


# PIPELINE DE DEPLOIEMENT

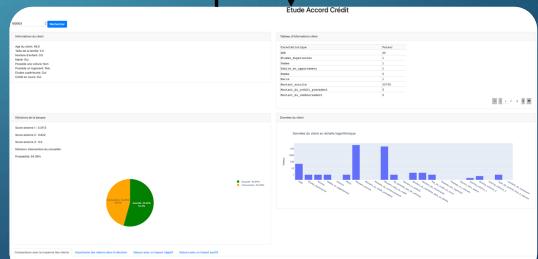
Tracking et stockage  
des modèles



Non-automatisation du  
déploiement des  
modèles par MLFLOW.



Requête API      Réponse JSON



Ajout note methodologique a finir	c56e256
xparisot committed 2 days ago	
ajout du data drift et de son html	0a0bfdb
xparisot committed 2 days ago	
maj fichier pour test yaml	b829730
xparisot committed 2 days ago ✓	
Update main.yml	Verified b31a8ed
xparisot committed 2 days ago	
installation des fichiers Procfiles, requierement et runtime a la rac... ...	e6c25e5
xparisot committed 2 days ago	
Mise a jour fichier .yml : modifications secret et version heroku, ma... ...	f0fcf30d
xparisot committed 2 days ago ✘	
Update main.yml	Verified 3b1ff21
xparisot committed 2 days ago	
Update main.yml	Verified 8480762
xparisot committed 2 days ago	
Update main.yml	Verified 3345232

```
DC) mbp-de-xavier:Projet_7 xparisot$ git commit -m "ajout du data drift et de son html"
[main 0a0bfdb] ajout du data drift et de son html
2 files changed, 2652 insertions(+)
create mode 100644 data_drift/data_drift.html
create mode 100644 data_drift/data_drift.ipynb
DC) mbp-de-xavier:Projet_7 xparisot$ git push origin main
  numérotation des objets: 6, fait.
  décompte des objets: 100% (6/6), fait.
  compression par delta en utilisant jusqu'à 12 fils d'exécution
  compression des objets: 100% (5/5), fait.
  écriture des objets: 100% (5/5), 4.48 Mio | 3.45 Mio/s, fait.
  total 5 (delta 1), réutilisés 0 (delta 0), réutilisés du pack 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xparisot/Implementez_un_modele_de_scoring.git
 b829730..0a0bfdb  main -> main
```

# PIPELINE DE DÉPLOIEMENT

- Git, GitHub et GitHub Actions
  - Commit via ligne de commande
  - Suivi des commit sur GitHub

```
1 name: Deploy API to Heroku
2
3 on:
4   push:
5     branches:
6       - main
7     paths:
8       - 'api/**'
9
10 jobs:
11   deploy:
12     runs-on: ubuntu-latest
13
14     steps:
15       - name: Checkout code
16         uses: actions/checkout@v2
17
18       - name: Setup Python
19         uses: actions/setup-python@v2
20         with:
21           python-version: '3.x'
22
23       # Install dependencies
24       - name: Install dependencies
25         run: pip install -r api/requirements.txt
26
27       # Check if api2.py was modified
28       - name: Check if api2.py was modified
29         id: checkfile
30         run: echo "::set-output name=api2_modified::$(git diff --name-only HEAD~ HEAD | grep 'api/api2.py')"
31
32       # Run tests on api2.py only if api2.py was modified
33       - name: Run tests on api2.py
34         if: steps.checkfile.outputs.api2_modified == 'api/api2.py'
35         run: pytest api/api2.py
36
37       # Add Heroku remote
38       - name: Add Heroku remote
39         env:
40           HEROKU_API_TOKEN: ${{ secrets.HEROKU_API_TOKEN }}
41         run: git remote add heroku https://heroku:$HEROKU_API_TOKEN@git.heroku.com/api2.git
42
43       # Deploy to Heroku
44       - name: Push API folder to Heroku
45         run: git subtree push --prefix api heroku main
```

# PIPELINE DE DÉPLOIEMENT SCRIPT YAML GIT ACTION

deploy  
succeeded 2 days ago in 2m 16s

- > ✓ Set up job
- > ✓ Checkout code
- > ✓ Setup Python
- > ✓ Install dependencies
- > ✓ Check if api2.py was modified
- > ✓ Run tests on api2.py
- > ✓ Add Heroku remote
- > ✓ Push API folder to Heroku
- > ✓ Post Setup Python
- > ✓ Post Checkout code
- > ✓ Complete job

# DEPLOIEMENT DE L'API ET DU DASHBOARD

## TEST DE L'API VIA PYTEST

```
(OC) mbp-de-xavier:api xparisot$ pytest
----- test session starts -----
platform darwin -- Python 3.10.12, pytest-7.4.0, pluggy-1.2.0
rootdir: /Users/xparisot/Formation_OpenClassroom/Projets/Projet_7/api
plugins: dash-2.11.1, anyio-3.7.1
collected 8 items

test/test_api.py ......

----- 8 passed in 2.09s -----
[100%]
(OC) mbp-de-xavier:api xparisot$ █
```

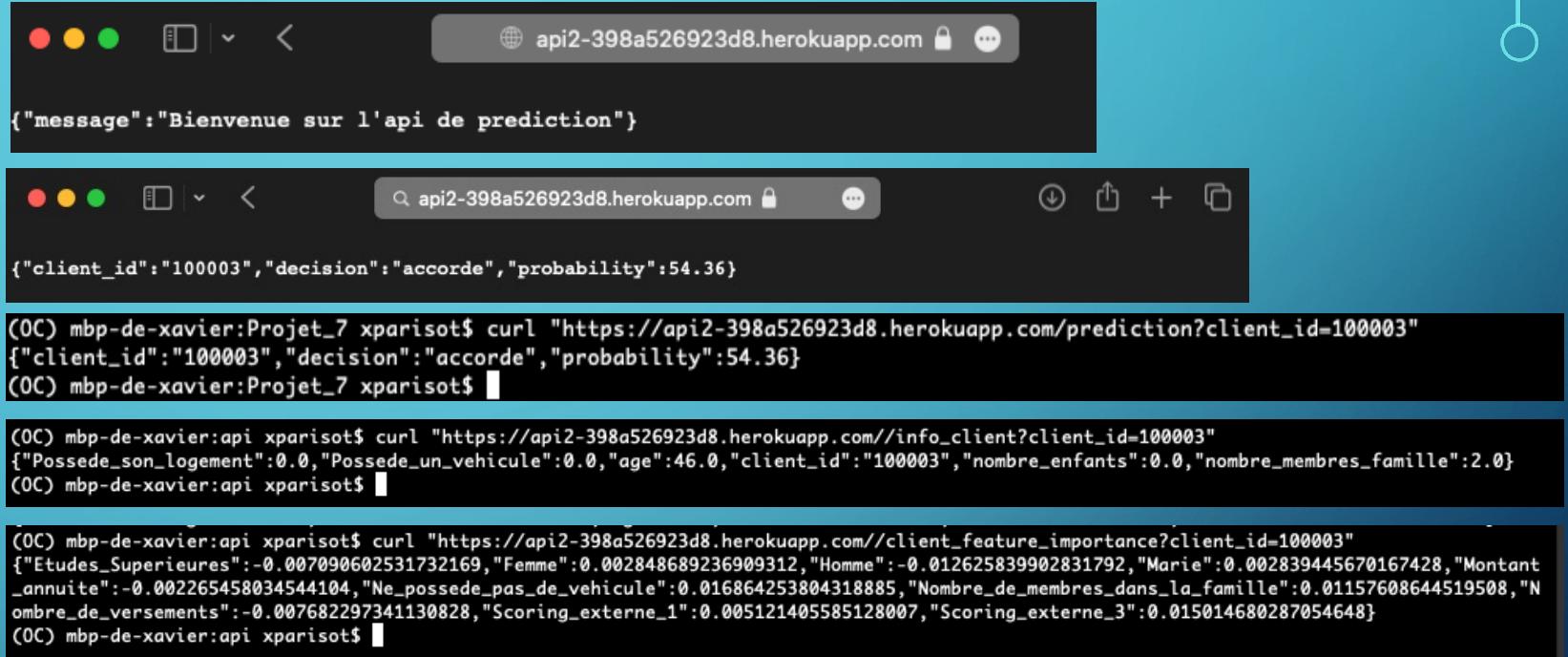
Lien GitHub : [https://github.com/xparisot/Implementez\\_un\\_modele\\_de\\_scoring](https://github.com/xparisot/Implementez_un_modele_de_scoring)

Lien api : <https://api2-398a526923d8.herokuapp.com>

Lien dashboard : <https://dashboard2-8d637637c5a8.herokuapp.com>

# DEPLOIEMENT DE L'API ET DU DASHBOARD REQUÊTE PRÉDICTION

Mise en œuvre de  
l'API et prédiction



The screenshot shows a terminal window with four distinct sections of text output:

- Section 1:** Displays the response to a general API call: 

```
{"message": "Bienvenue sur l'api de prediction"}
```
- Section 2:** Displays the response to a prediction request for client\_id 100003: 

```
{"client_id": "100003", "decision": "accorde", "probability": 54.36}
```
- Section 3:** Displays the response to an info\_client request for client\_id 100003: 

```
(0C) mbp-de-xavier:Projet_7 xparisot$ curl "https://api2-398a526923d8.herokuapp.com/prediction?client_id=100003"
{"client_id": "100003", "decision": "accorde", "probability": 54.36}
(0C) mbp-de-xavier:Projet_7 xparisot$ curl "https://api2-398a526923d8.herokuapp.com/info_client?client_id=100003"
{"Possede_son_logement": 0.0, "Possede_un_vehicule": 0.0, "age": 46.0, "client_id": "100003", "nombre_enfants": 0.0, "nombre_membres_famille": 2.0}
```
- Section 4:** Displays the response to a feature\_importance request for client\_id 100003: 

```
(0C) mbp-de-xavier:api xparisot$ curl "https://api2-398a526923d8.herokuapp.com/client_feature_importance?client_id=100003"
{"Etudes_Superieures": -0.007090602531732169, "Femme": 0.002848689236909312, "Homme": -0.012625839902831792, "Marie": 0.002839445670167428, "Montant_versements": -0.002265458034544104, "Ne_possee_pas_de_vehicule": 0.016864253804318885, "Nombre_de_membres_dans_la_famille": 0.01157608644519508, "Nombre_de_versements": -0.007682297341130828, "Scoring_externe_1": 0.005121405585128007, "Scoring_externe_3": 0.015014680287054648}
(0C) mbp-de-xavier:api xparisot$
```

Requêtes possibles :

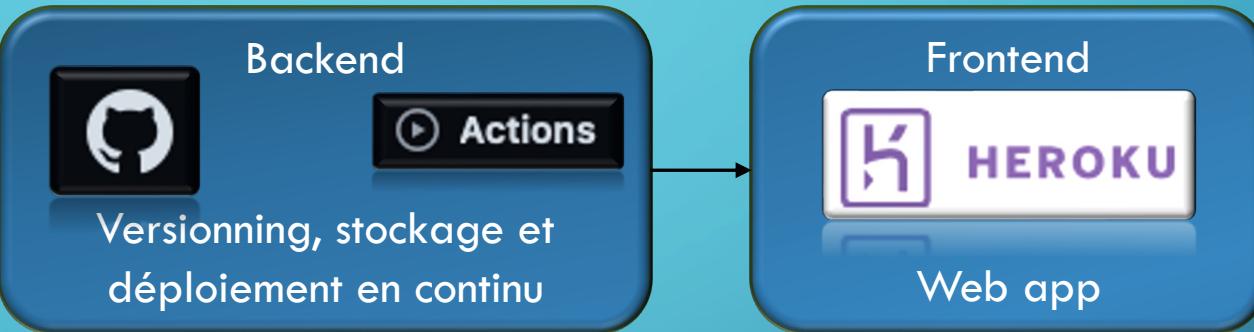
- /full\_client\_data
- /average\_values\_all
- /info\_client
- /info\_banque
- /feature\_importance
- /average\_values
- /client\_feature\_importance

# PIPELINE DE DEPLOIEMENT

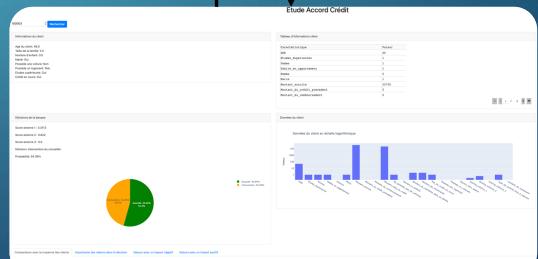
Tracking et stockage  
des modèles



Non-automatisation du  
déploiement des  
modèles par MLFLOW.



Requête API      Réponse JSON



# PIPELINE DE DÉPLOIEMENT DASHBOARD

**Etude Accord Crédit**

**Informations du client**

Age du client: 46.0  
 Taille de la famille: 2.0  
 Nombre d'enfant: 0.0  
 Marié: Oui  
 Possède une voiture: Non  
 Possède un logement: Non  
 Etudes supérieures: Oui  
 Crédit en cours: Oui

**Tableau d'informations client**

Caractéristique	Valeur
AGE	46
Etudes_Supérieures	1
Femme	1
Habite_en_appartement	1
Homme	0
Marie	1
Montant_annuite	35700
Montant_du_crédit_precedent	0
Montant_du_remboursement	0

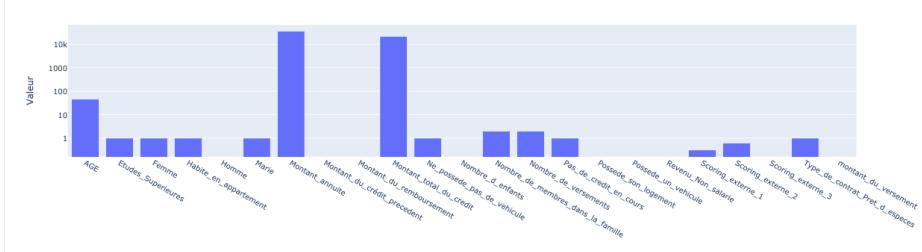
**Décisions de la banque**

Score externe 1: 0.3113  
 Score externe 2 : 0.622  
 Score externe 3 : 0.0  
 Décision: Intervention du conseiller  
 Probabilité: 54.36%



**Données du client**

**Données du client en échelle logarithmique**

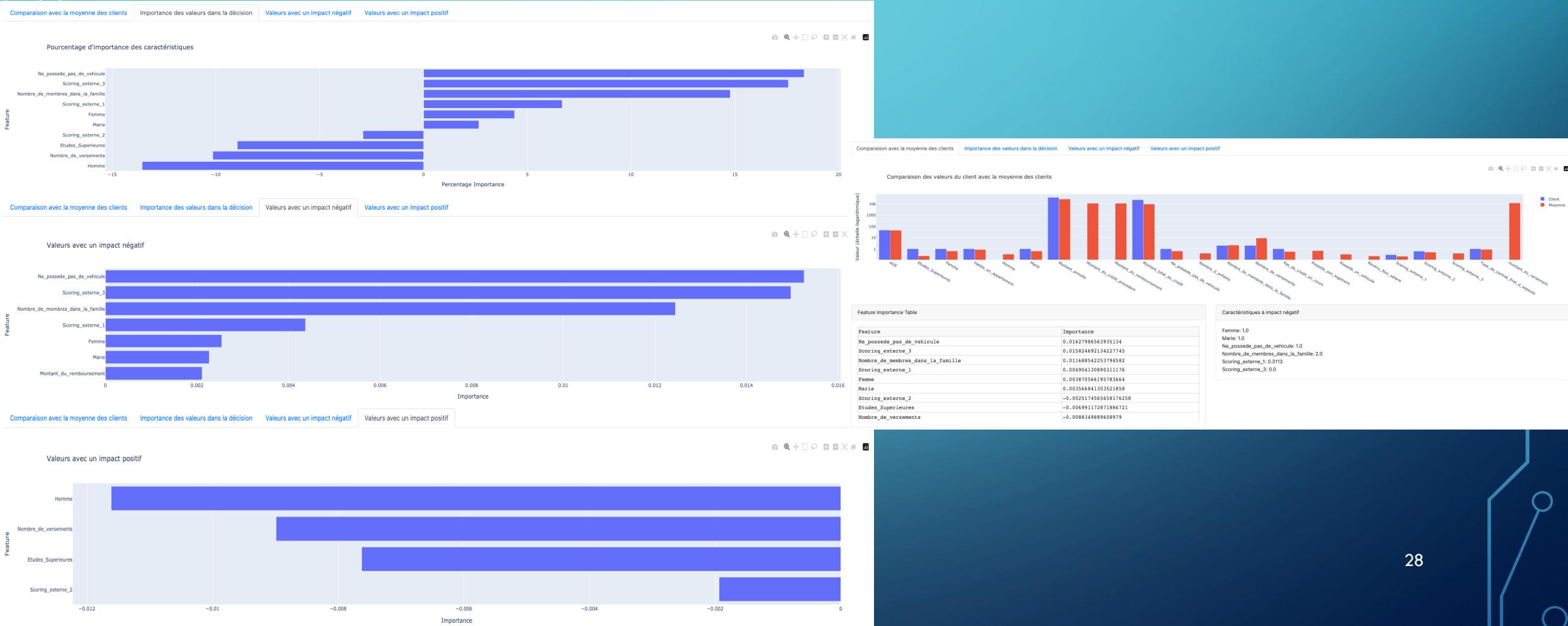


Caractéristique	Valeur
AGE	46
Etudes_Supérieures	1
Femme	1
Habite_en_appartement	1
Homme	0
Marie	1
Montant_annuite	35700
Montant_du_crédit_precedent	0
Montant_du_remboursement	0
Montant_total_du_credit	1000
Montant_total_du_remboursement	1
Nombre_d_enfants	1
Nombre_d_espaces	1
Nombre_d_membres_dans_la_famille	2
Nombre_de_versements	1
Possede_une_voiture	0
Possede_son_logement	1
Revenu	35700
Score_externe_1	0.3113
Score_externe_2	0.622
Score_externe_3	0.0
Type_de_contrat_Pret_d_espace	Crédit en cours
montant_du_verstement	35700

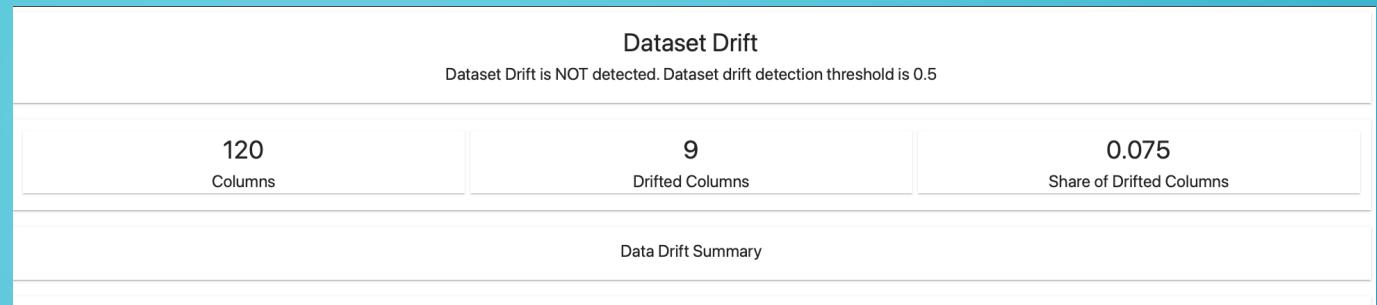
Comparaison avec la moyenne des clients    Importance des valeurs dans la décision    Valeurs avec un impact négatif    Valeurs avec un impact positif

Comparaison des valeurs du client avec la moyenne des clients

# PIPELINE DE DÉPLOIEMENT DASHBOARD



# DATA DRIFT



Drift is detected for 7.5% of columns (9 out of 120).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
AMT_REQ_CREDIT_BUREAU_QRT	num			Detected	Wasserstein distance (normed)	0.359052
AMT_REQ_CREDIT_BUREAU_MON	num			Detected	Wasserstein distance (normed)	0.281765
AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210785
AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334
AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161102
AMT_REQ_CREDIT_BUREAU_WEEK	num			Detected	Wasserstein distance (normed)	0.15426
NAME_CONTRACT_TYPE	cat			Detected	Jensen-Shannon distance	0.14755
DAYS_LAST_PHONE_CHANGE	num			Detected	Wasserstein distance (normed)	0.138977
FLAG_EMAIL	num			Detected	Jensen-Shannon distance	0.122121
FLAG_DOCUMENT_3	num			Not Detected	Jensen-Shannon distance	0.062496

## Qu'est-ce que le Data Drift ?

Le "data drift" fait référence à une modification ou un changement dans la distribution des données au fil du temps, ce qui peut affecter les performances d'un modèle de machine learning. Le modèle, en effet, est formé sur un ensemble de données spécifique, et si les données sur lesquelles il fait des prédictions changent, les performances du modèle peuvent en être affectées. Le data drift peut survenir pour diverses raisons, telles que les changements saisonniers, les modifications du comportement des utilisateurs, les changements de politique, etc.

# CONCLUSION

- Le modèle Light GBM est le modèle utilisé pour la prédiction
- Ce projet bien que très intéressant et permettant d'acquérir des connaissances complémentaires est très, trop ? touffu. Il donne l'impression d'être un amalgame de demandes. J'ai eu beaucoup de difficultés à le réaliser, ne connaissant pas de langages autres que python.
- Le dashboard correspond à ce que j'aurais souhaité avoir lors de mes années en banque.



MERCI POUR VOTRE  
ÉCOUTE