

---

# MISE EN PLACE D'UN OUTIL DE SCORING

CRÉATION D'UN MODÈLE DE CLASSIFICATION ET DÉPLOIEMENT SUR LE CLOUD



**Prêt à dépenser**



# INTRODUCTION

- 1) CONTEXTE
  - 2) DÉMARCHE DE LA CONCEPTION
- 

# CONTEXTE GÉNÉRAL

- **Contexte :**

L'entreprise ***Prêt à dépenser*** 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.

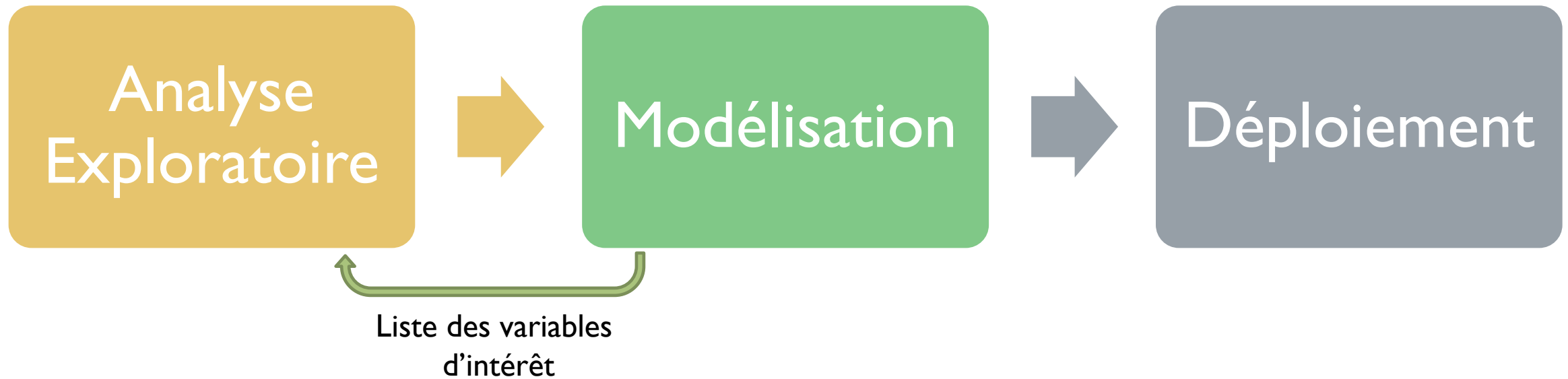
- **Mission :**

Réaliser une première étude de faisabilité d'un moteur de classification

1. Construire **un modèle de scoring** qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique.
2. Déployer le modèle sous forme **d'API** sur le cloud
3. Construire un **Dashboard interactif** (prédictions + explications) utilisant l'API



# LES TROIS ÉTAPES DE LA MISE EN PLACE DE L'OUTIL DE SCORING





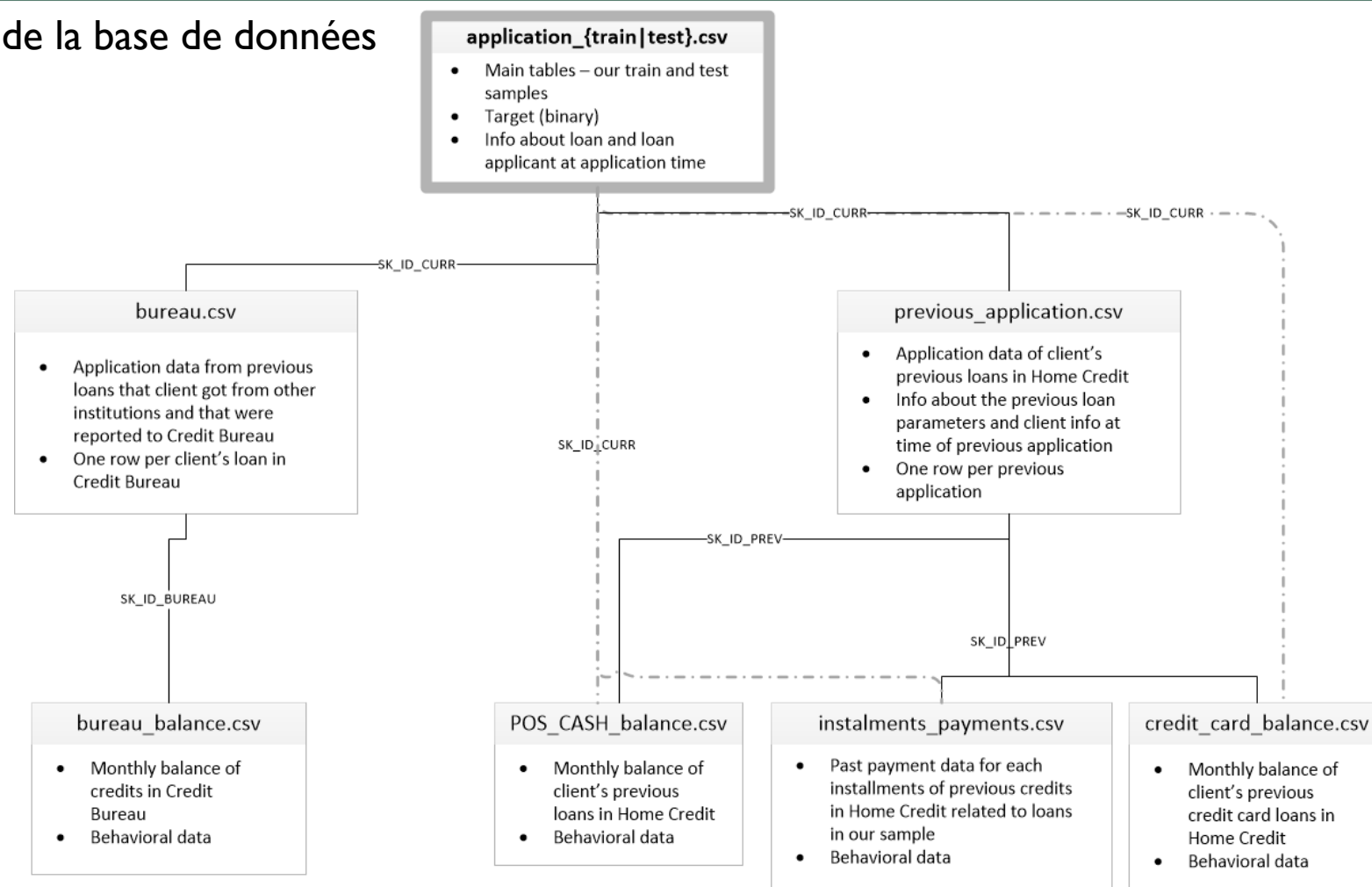
# I) ANALYSE EXPLORATOIRE

- 1) PRÉSENTATION DES DONNÉES
  - 2) CRÉATION DE VARIABLES (KERNEL KAGGLE)
  - 3) GESTION DES VALEURS MANQUANTES
  - 4) SÉLECTION DE VARIABLES
- 

# I) ANALYSE EXPLORATOIRE

## Présentation des données

### Schéma de la base de données



# I) ANALYSE EXPLORATOIRE

## Présentation des données

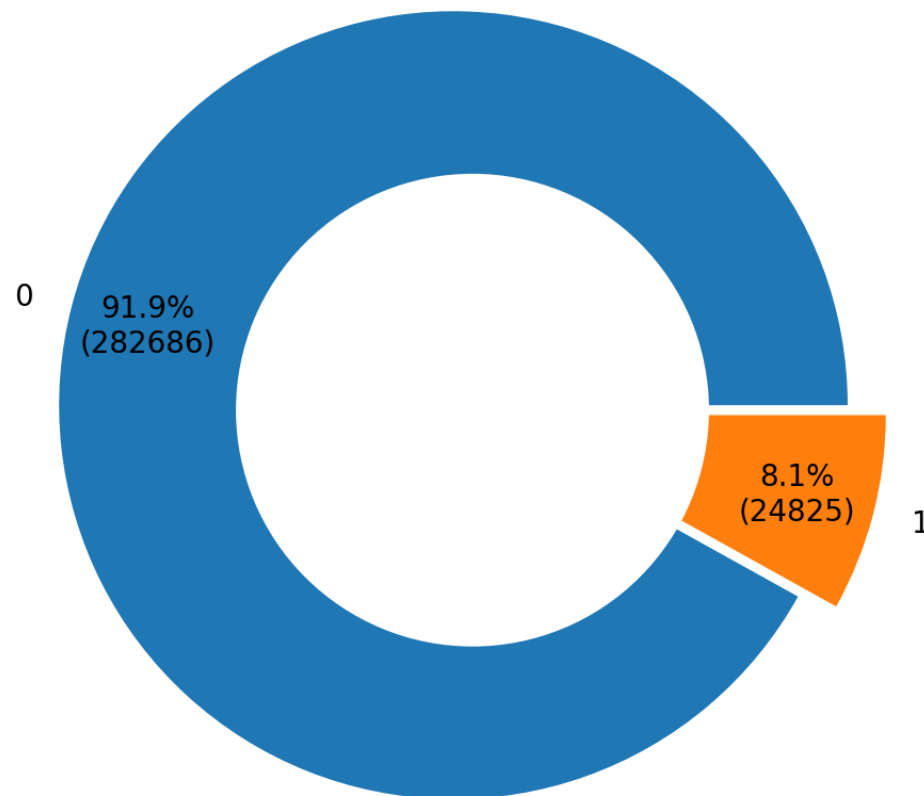
Nombre de clients : 307511

Variable cible à prédire : TARGET

- 0 > Client pouvant rembourser le crédit
- 1 > Client ayant des difficultés de remboursement

> Il y a un déséquilibre de classe entre les deux catégories.

Number of clients with difficulties (1) or not (0) to repay the loan



# I) ANALYSE EXPLORATOIRE

Présentation des  
données



Analyse et création  
de variables  
explicatives

Création de variables :

Kernel Kaggle de J.Aguiar (<https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script>)

Pour chaque table (bureau, previous application ...) :

- Création d'une colonne pour chaque variable : min, max, moyenne, médiane, comptage
- Transformation de certaines variables en pourcentage
- Transformation des variables catégorielles (One Hot Encoding)
- Jointure aux données principales (application\_train) par la clé SK\_ID\_CURR

➤ 769 variables explicatives

/!\ la variable Code-Gender a été retirée pour éviter de la discrimination de genres



# I) ANALYSE EXPLORATOIRE

## Présentation des données



## Analyse et création de variables explicatives



## Gestion des valeurs manquantes

[illegible]

- Retrait de variables avec 80% + de valeurs manquantes (37 var.)

# I) ANALYSE EXPLORATOIRE

Présentation des données



Analyse et création de variables explicatives



Gestion des valeurs manquantes



Sélection de variables

- Classification linéaire SVC avec pénalité L1  
172 variables sélectionnées

Modélisation

- « *Feature importance* » du meilleur modèle : 62 variables affectant le plus la décision (60% de la somme cumulée de l'importance des features)

# I) ANALYSE EXPLORATOIRE

Présentation des données



Analyse et création de variables explicatives



Gestion des valeurs manquantes



Sélection de variables

Pour de nouvelles données ( application – test) :

Création d'une fonction externe qui reprend l'ensembles des précédentes étapes :


- 1) Création de features
- 2) Spécification de l'index (SK\_ID\_CURR)
- 3) Changement de noms de certaines variables (regex)
- 4) Sélection des variables d'intérêts

La fonction demande 3 inputs :

- 1) Select the path of the data
- 2) Name of the csv file to prepare
- 3) Name of the output file (with extension)



## II) MODÉLISATION

- 1) PRÉSENTATION DES ÉTAPES DE LA MODÉLISATION
  - 2) PRÉSENTATION DES MÉTRIQUES UTILISÉES
  - 3) DÉTERMINATION DU MEILLEUR MODÈLE
  - 4) OPTIMISATION ET FINALISATION DU MODÈLE
  - 5) DÉRIVE DES DONNÉES
  - 6) EXPLICATION DU MODÈLE
- 

# I) LES TROIS ÉTAPES DE LA MODÉLISATION

## Initialisation de Pycaret

- Variable cible = 'TARGET'
- Autres variables = numériques
- Imputation données manquantes = remplacement par 0
- Gestion du déséquilibre des données : Non (sous-ech.) / SMOTE (sur-ech.)
- Validation croisée = 4 fold

## I) Détermination du meilleur modèle

Under-sampling

Over-sampling

Pycaret

MLFlow tracking

## II) Optimisation du meilleur modèle

Hyper-parameter tuning  
(Optuna)

## III) Finalisation du meilleur modèle

Entrainement sur les  
données totales

## II) PRÉSENTATION DES MÉTRIQUES UTILISÉES

Valeurs vraies	0	Vrais négatifs (TN)	Faux positifs (FP)
	1	Faux négatifs (FN)	Vrais positifs (TP)
		0	1

Valeurs prédites

Métrique à optimiser :  
Indice de Profit

$$\frac{(TP \times 0) + (TN \times 3) + (FP \times -1) + (FN \times -10)}{TP + TN + FP + FN}$$

Métrique	Description
<b>AUC</b>	Aire sous la courbe ROC construite avec le couple « taux de vrais positifs (TPR = Recall) » et « taux de faux positifs (FPR) » à différents seuils de décision. $TPR = \frac{TP}{TP+FN}$ $FPR = \frac{FP}{FP+TN}$
<b>Accuracy</b>	Prédictions correctement identifiées $\frac{TP+TN}{TP+TN+FP+FN}$
<b>Recall</b>	Proportion de vrais positifs identifiés correctement $\frac{TP}{TP+FN}$
<b>Precision</b>	Proportion de prédiction positives étant correcte : $\frac{TP}{TP+FP}$
<b>F1</b>	Combinaison de la Precision et du Recall : $2 \times \left( \frac{Precision * Recall}{Precision + Recall} \right)$
<b>TT(sec)</b>	Temps d'entrainements

### III) DÉTERMINATION DU MEILLEUR MODÈLE

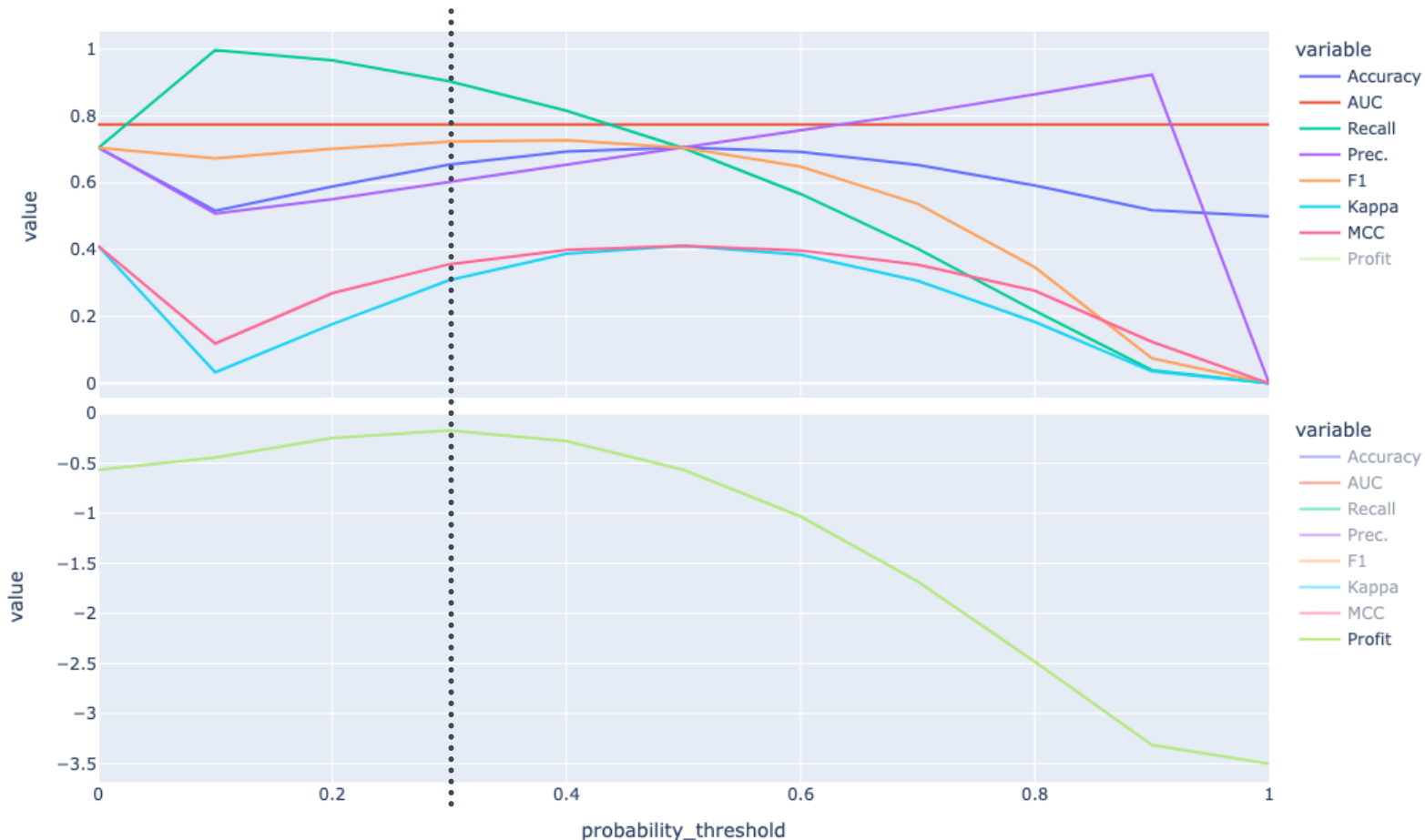
Sous échantillonnage (undersampling) : échantillon de 24825 clients de classe 1 et de classe 0

Résultats :

Model	Accuracy	AUC	Recall	Prec.	FI	Profit	TT (Sec)
Light Gradient Boosting Machine	0.7055	0.7747	0.7039	0.7062	0.705	-0.566	0.9025
Gradient Boosting Classifier	0.6998	0.7677	0.7064	0.6971	0.7017	-0.5817	15.58
Ada Boost Classifier	0.6899	0.7539	0.6925	0.6889	0.6907	-0.6628	3.8025
Random Forest Classifier	0.6927	0.7556	0.6776	0.6987	0.688	-0.6961	8.125
Linear Discriminant Analysis	0.6822	0.7438	0.6821	0.6823	0.6821	-0.7248	0.435
Ridge Classifier	0.6818	0.0	0.6816	0.6819	0.6817	-0.7278	0.1575
Extra Trees Classifier	0.6872	0.7508	0.6677	0.6948	0.681	-0.7478	3.095
Naive Bayes	0.5608	0.6217	0.807	0.541	0.6473	-0.8359	0.1975
Quadratic Discriminant Analysis	0.6356	0.6775	0.5746	0.6592	0.6099	-1.2337	0.23
Logistic Regression	0.6062	0.646	0.5742	0.6136	0.5932	-1.3523	9.9525
Decision Tree Classifier	0.5908	0.5908	0.5887	0.5911	0.5899	-1.3707	1.8825
SVM - Linear Kernel	0.5113	0.0	0.6627	0.5309	0.5542	-1.4667	0.7525
K Neighbors Classifier	0.5495	0.5661	0.5592	0.5485	0.5538	-1.6242	4.38
Dummy Classifier	0.5	0.5	0.0	0.0	0.0	-3.4998	0.105

### III) DÉTERMINATION DU MEILLEUR MODÈLE

Light Gradient Boosting Machine Probability Threshold Optimization (default = 0.5)



Seuil de décision :  
Prédiction = classe I si probabilité  
d'appartenir à la classe I  $> 0.5$

➤ Meilleur seuil de décision : 0.3



### III) DÉTERMINATION DU MEILLEUR MODÈLE

Sur-échantillonnage (overampling) : **S**ynthetic **M**inority **O**versampling **T**echnique (Chawla et al. *J. Artif. Intell. Res.* 2022)

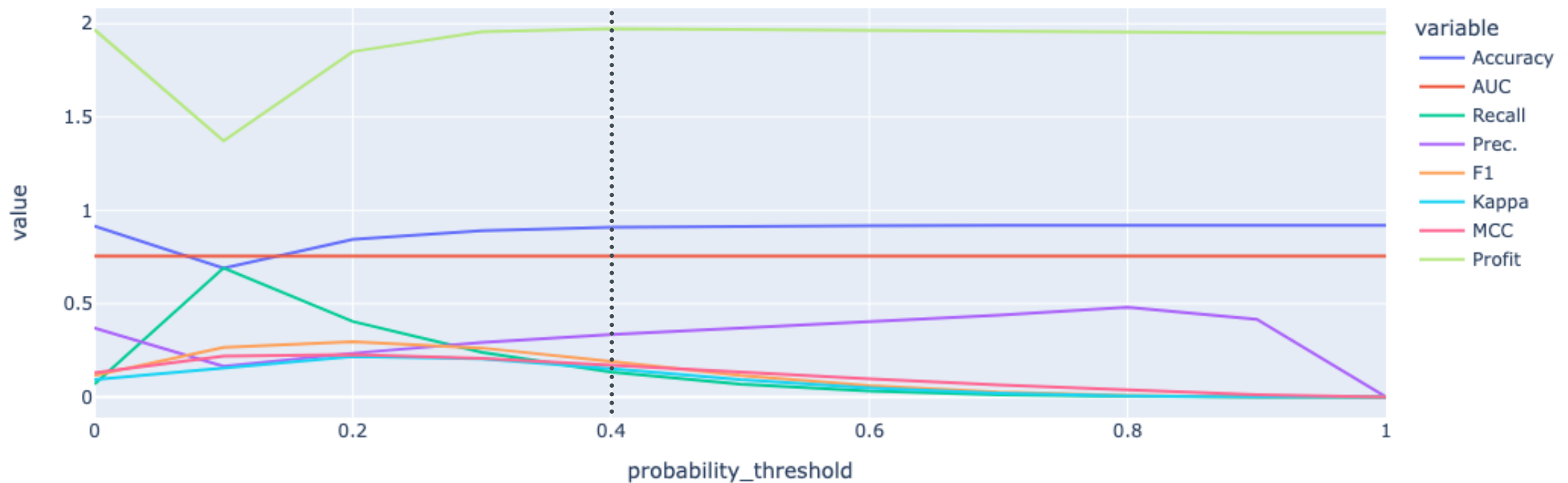
➤ Création de points basés sur les k plus proches voisins + du bruit de fond

Résultats :

Model	Accuracy	AUC	Recall	Prec.	F1	Profit	TT (Sec)
Extra Trees Classifier	0.9145	0.7397	0.0769	0.3616	0.1269	1.9688	40.6367
Light Gradient Boosting Machine	0.9154	0.7558	0.0684	0.3706	0.1154	1.9683	7.5267
Random Forest Classifier	0.9125	0.7275	0.0684	0.3093	0.112	1.9565	106.4533
Dummy Classifier	0.9193	0.5	0.0	0.0	0.0	1.9506	0.7033
Gradient Boosting Classifier	0.9049	0.7225	0.1118	0.2787	0.1596	1.9473	256.9267
Ada Boost Classifier	0.8765	0.711	0.2039	0.2175	0.2105	1.8783	64.93
Decision Tree Classifier	0.8246	0.549	0.2202	0.1365	0.1685	1.6786	12.4633
SVM - Linear Kernel	0.8076	0.0	0.2303	0.1525	0.1341	1.6156	3.2533
Linear Discriminant Analysis	0.6872	0.7446	0.6785	0.1603	0.2594	1.351	3.7567
Ridge Classifier	0.6872	0.0	0.6785	0.1603	0.2594	1.3507	1.0167
K Neighbors Classifier	0.6804	0.5539	0.3711	0.1002	0.1578	1.1747	220.4467
Logistic Regression	0.6361	0.6462	0.5706	0.1228	0.202	1.0943	30.1133
Quadratic Discriminant Analysis	0.4526	0.6485	0.747	0.1029	0.1808	0.4457	5.1333
Naive Bayes	0.3018	0.6141	0.8418	0.0902	0.1629	-0.1116	1.22

### III) DÉTERMINATION DU MEILLEUR MODÈLE

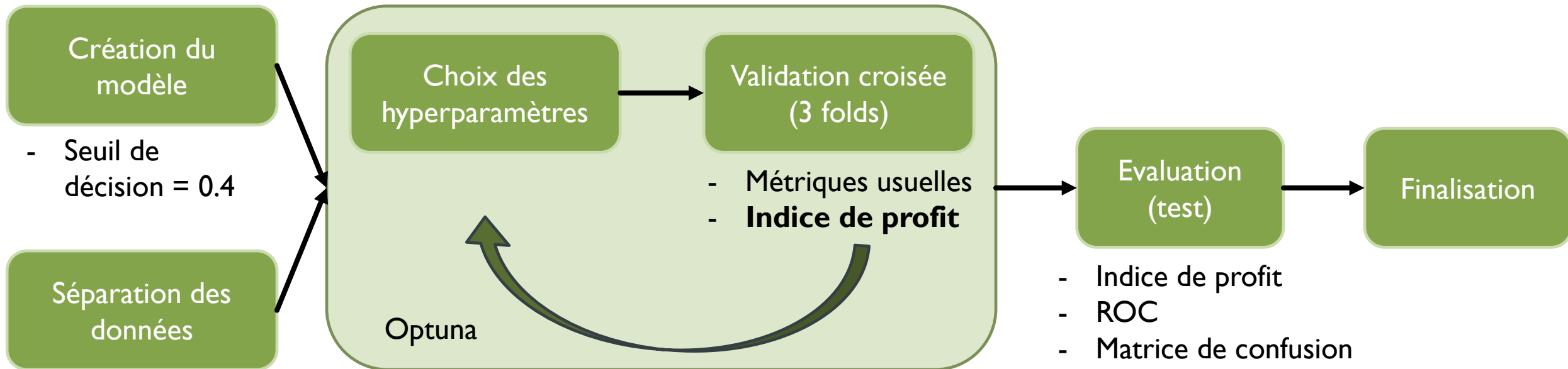
Light Gradient Boosting Machine Probability Threshold Optimization (default = 0.5)



➤ Meilleur seuil de décision : 0.4

## IV) OPTIMISATION PAR OPTUNA

- Optuna : Bibliothèque spécialisé dans l'optimisation des hyperparamètres du modèle
  - Algorithme de Tree-structured Parzen Estimator (TPESampler) > Approche bayésienne





# PRÉSENTATION DU TRACKING MLFLOW

[mlflow ui](#)

## V) LA DÉRIVE DES DONNÉES

- La dérive des données correspond à des changements de distributions de nos variables avec l'ajout ou modification de clients.
- Le modèle a été entraîné sur des données désormais obsolète
  - Diminution de ses performances au cours du temps
- Données de références (entraînement) VS Données cibles (ici données de test)



# PRÉSENTATION DU RAPPORT DE DÉRIVE DES DONNÉES (DATA DRIFT)

Report from evidently

## VI) EXPLICATION DU MODÈLE

- Problème :

Afin de faire preuve de transparence et de pouvoir justifier une décision (comme accorder un crédit), il est nécessaire de pouvoir interpréter les prédictions du modèle.

- Solution :

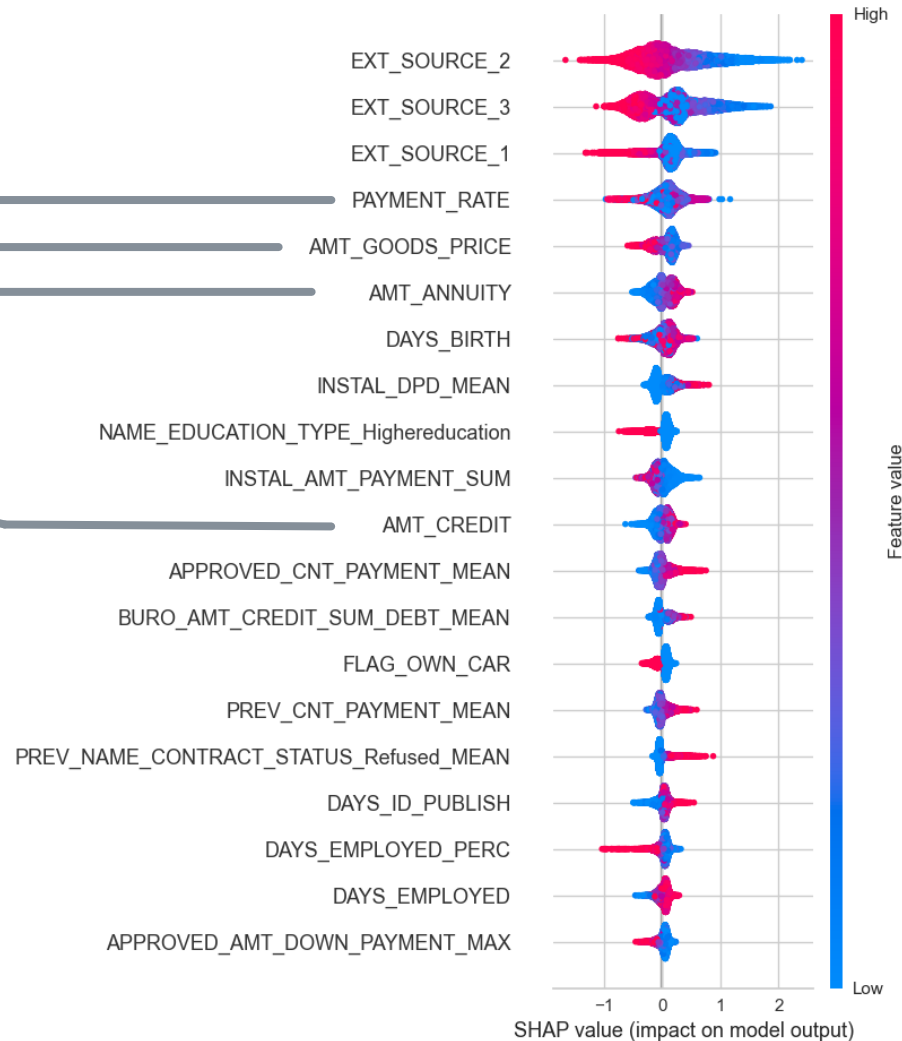
Calcule des valeur SHAP (SHapley Additive exPlanations) > quantification de l'influence de chaque variable sur la prédiction **au niveau local** (pour chaque client).

La moyenne des valeurs SHAP de tous les individus (clients) permettent de calculer l'effet de chaque variable sur la réponse, **au niveau global**.

## VI) EXPLICATION DU MODÈLE

Dérive de données :

PAYMENT\_RATE -  
INCOME\_CREDIT\_PERC -  
AMT\_GOODS\_PRICE -  
AMT\_CREDIT -  
AMT\_ANNUITY -  
NAME\_CONTRACT\_TYPE\_Cashloans -  
DAYS\_LAST\_PHONE\_CHANGE -



- 4 variables importantes évoluent avec de nouveaux clients.
- Il peut être nécessaire de ré-entraîner le modèle sur un nouveau jeu de données
- Création d'une alerte : réentraînement lorsque l'AUC global dépasse un certain seuil.





## III) DÉPLOIEMENT

- 1) PRÉSENTATION DU DÉPLOIEMENT GITHUB
  - 2) PRÉSENTATION DE L'API - FASTAPI
  - 3) PRÉSENTATION DU DASHBOARD - STREAMLIT
- 



## PRÉSENTATION DES DOSSIERS GITHUB

<https://github.com/Cdubois1992/OC-DS-P7-frontend>

<https://github.com/Cdubois1992/OC-DS-P7-backend>

## PRÉSENTATION DE L'API

<https://credit-fastapi-oc.herokuapp.com/>

## PRÉSENTATION DE L'APPLICATION STREAMLIT

<https://dubois-credit-frontend-home.streamlit.app/>



# CONCLUSION



# PISTES D'AMÉLIORATIONS

- Modélisation :
  - Création de variables : Prise en compte d'interactions entre les variables
  - Sélection de variables : Algorithmes performants et robustes (Boruta)
  - Modification de la fonction de « l' indice de Profit » avec des valeurs réelles
  - Augmentation du nombre d'itérations d'Optuna (optimisation des hyperparamètres)
- Optimisation du Dashboard et de l'API :
  - Intégration du calcul des valeurs SHAP par l'API
- Dérive des données :
  - Enregistrement des logs de l'API (requests input and output) et calcul de l'AUC afin de déterminer une alerte pour réentraîner le modèle

