

# Implémentez un Modèle de Scoring

PROJET 07/ Openclassrooms

Gulsum Kapanoglu



# Dans ce Project..

- ✓ **Problématique**
- ✓ **EDA (Analyse des données)**
- ✓ **Recherche de Meilleur Model**
- ✓ **Modélisation**
- ✓ **Choix de Model (LGBT)**
- ✓ **Feature Importance**
- ✓ **Api**
- ✓ **Dashboard & Streamlit**
- ✓ **Conclusion**

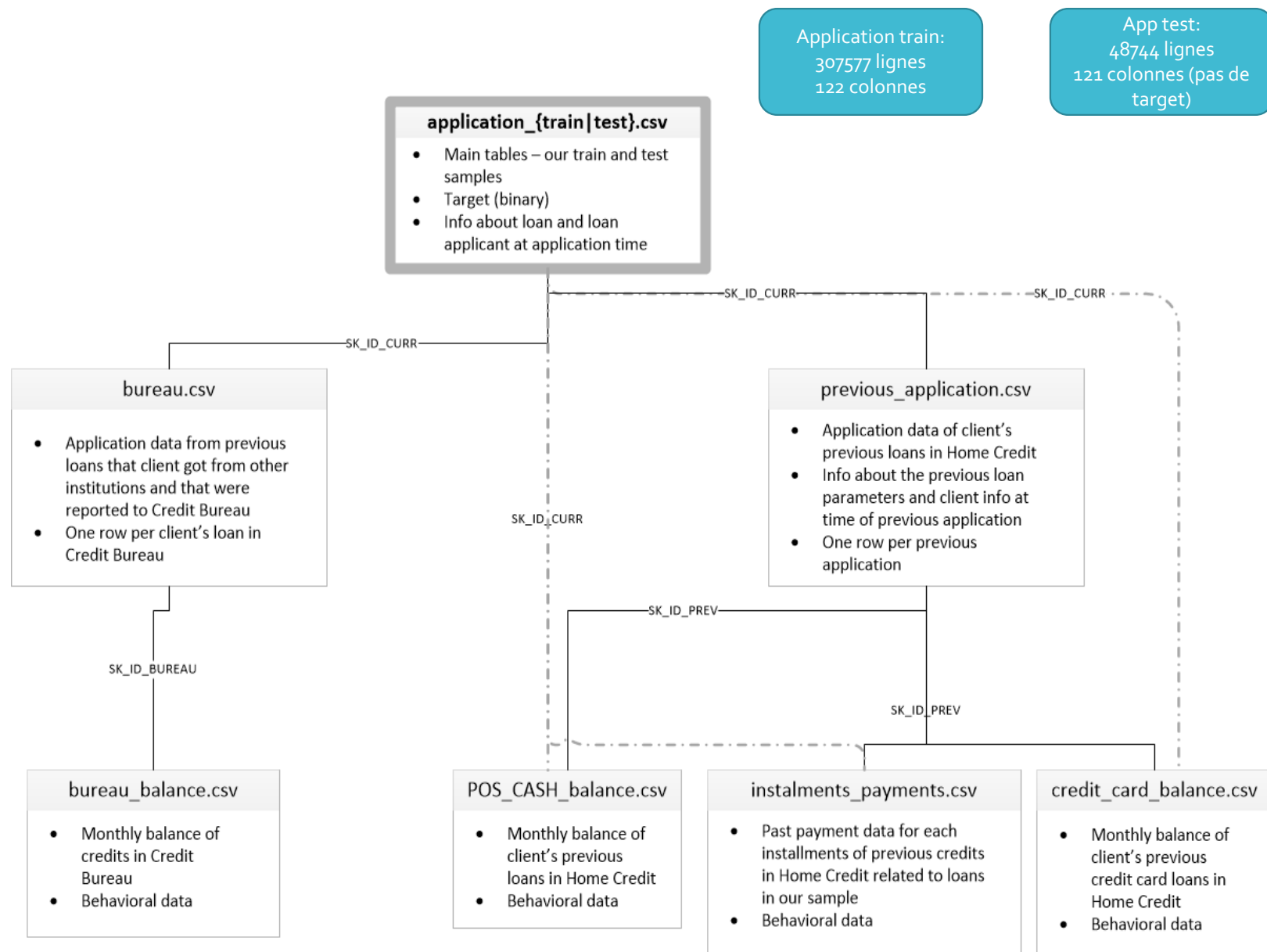
# Problématique

# Problématique

- Entreprise « Prêt à dépenser » :
  - **Prêt à dépenser** souhaite *développer un modèle de Scoring de la probabilité de défaut de paiement du client* pour étayer la décision d'accorder ou non un prêt à un client potentiel.
- Objectifs :
  - Construire un Modèle de scoring de la probabilité de défaut de paiement du client
  - Construire un Dashboard interactif à destination des chargés de relation client



# Données



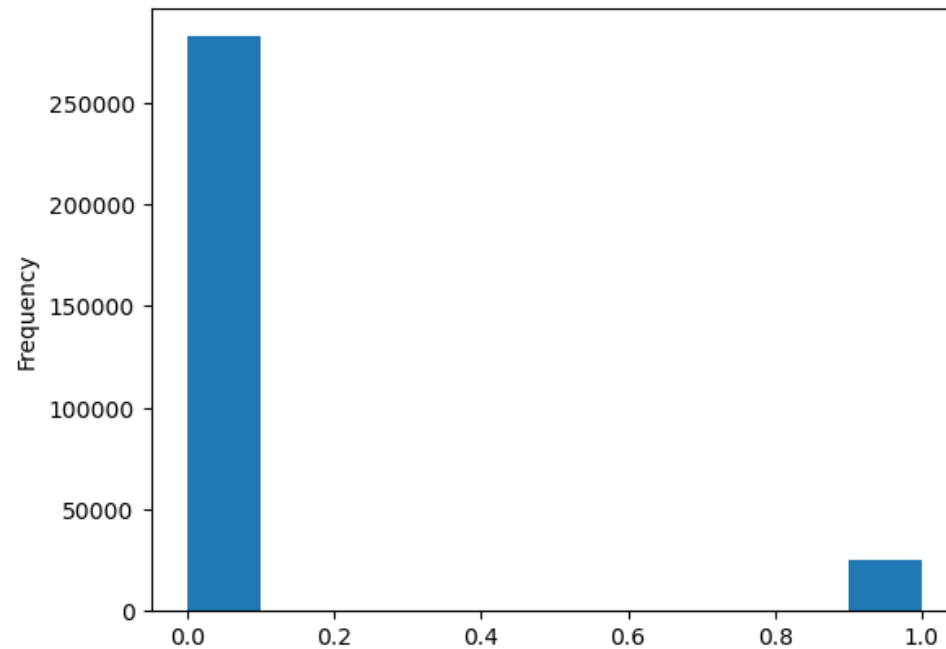
# Analyse des données

# Analyse des données

## Analyse de la TARGET

Problème de classification à 2 classes: 0 ou 1

- 0: Client rembourse son prêt
- 1: Client n'honore pas son prêt (totalement ou partiellement)



Classes déséquilibrées

- **91,2%** des clients en classe 0
- **8,1%** des clients en classe 1

# Préparation des Données avec un kernel

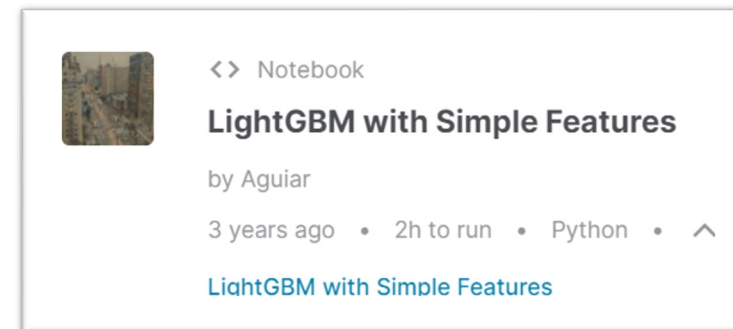
**Feature engineering** : Les features sont créées en appliquant les fonctions min, max, moyenne, somme, pourcentage, division, et variance à des tables groupées(PAYMENT\_RATE, ANNUITY\_INCOME\_PERC.....)

**Preprocessing** : Encodage des variables catégorielles avec One-hot encoding

- **Jointures** : Toutes les tables sont jointes à l'aide de la clé SK\_ID\_CURR

Source :

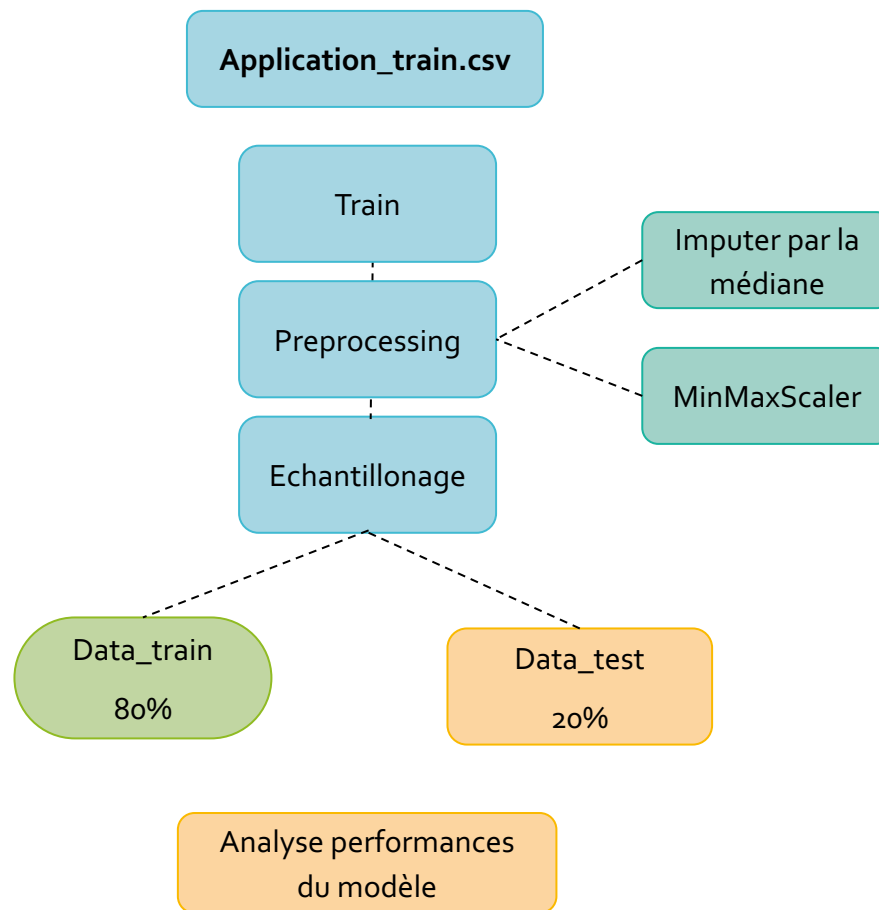
- Kernel LightGBM with Simple Features
- Lien : <https://www.kaggle.com/jsaguiar/lightgbm-with-simple-features>





# Modélisation

# Modelisation



**Application\_test.csv**

Ce dataset ne contenant pas de target sera utilisé dans la partie dashboard pour simuler des nouveaux clients.

# ENTRAINEMENT ET OPTIMISATION

1

Equilibrage des données

- Utilisation de la librairie **imblearn**
- Under-Sampling

2

Choix de la meilleure  
Hypothèse

- Utilisation d'une régression logistique comme baseline
- Hypothèse retenue : *Domain Features*

3

Essais de modélisation

- ✓ DummyClassifier (Baseline)
- ✓ Light Gradient Boosting Machine
- ✓ Régression Logistique
- ✓ RandomForestClassifier

4

Optimisation du modèle  
le plus prometteur

- Modèle retenu : XGBoost
- Optimisation par RandomizedSearchCV

# Processus de rééquilibrage



On va utiliser 3 approches et comparer les résultats pour l'ensemble des modèles.

- **Undersampling** : supprimer des observations de la classe majoritaire afin de rééquilibrer le jeu de données
- **Oversampling** : répéter des observations de la classe minoritaire afin de rééquilibrer le jeu de données
- **Weight\_balanced** : indiquer au modèle le déséquilibre afin qu'il en tienne compte directement

```
#Undersampling
rus = RandomUnderSampler(random_state=42,sampling_strategy=0.4)
xtrain_us, ytrain_us = rus.fit_resample(xtrain , y_train)
```

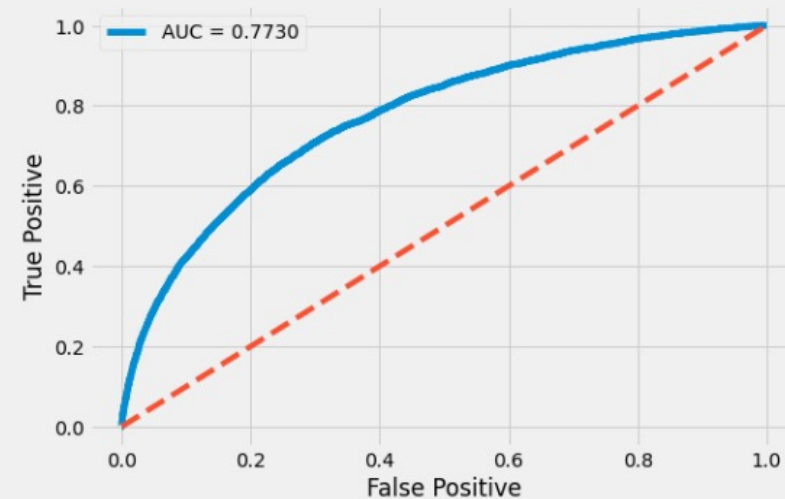
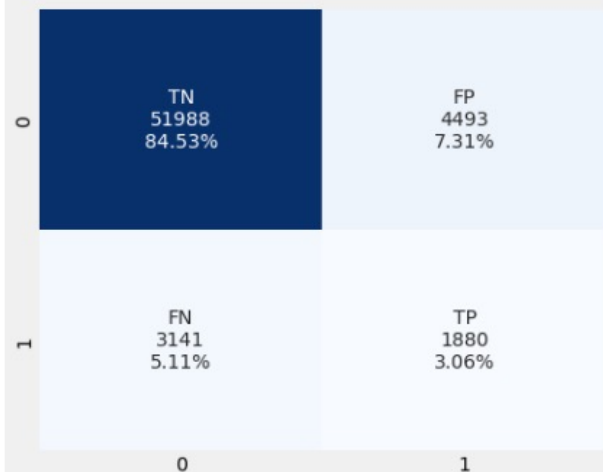
```
#Oversampling
smote = SMOTE(random_state=42,sampling_strategy=0.4)
xtrain_os, ytrain_os = smote.fit_resample(xtrain , y_train)
```

# Choix du Modèle

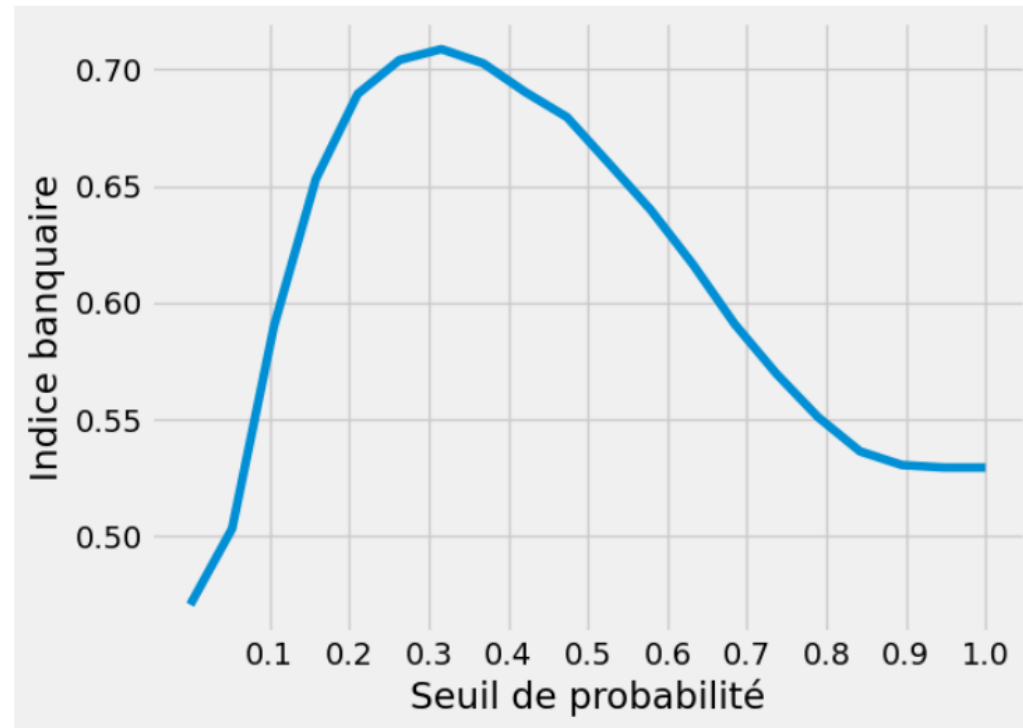
# LGBT & Undersampling

	Algorithm	Balancing_method	AUC	AUC_test	Time
0	Baseline	Undersampling	0.500	0.500	0.016013
1	Baseline	Oversampling	0.500	0.500	0.040002
2	Baseline	Balanced	0.499	0.506	0.032003
3	LGBM	Undersampling	0.773	0.771	106.270308
4	LGBM	Oversampling	0.758	0.757	986.328474
5	LGBM	Balanced	0.774	0.773	303.566205
6	LogisticRegression	Undersampling	0.740	0.740	111.869798
7	LogisticRegression	Oversampling	0.728	0.730	1643.591990
8	LogisticRegression	Balanced	0.738	0.739	300.206112
9	RandomForest	Undersampling	0.740	0.738	154.615660
10	RandomForest	Oversampling	0.665	0.663	2058.111072
11	RandomForest	Balanced	0.732	0.733	826.940315

**Light Gradient Boosting M (undersampling)**



# Optimisation du Seuil



le seuil est fixé à 0,32

2 coûts à optimiser

Prêter le moins possible à des « mauvais clients »

➤ FN (Faux Négatif: Faux Bon Client)

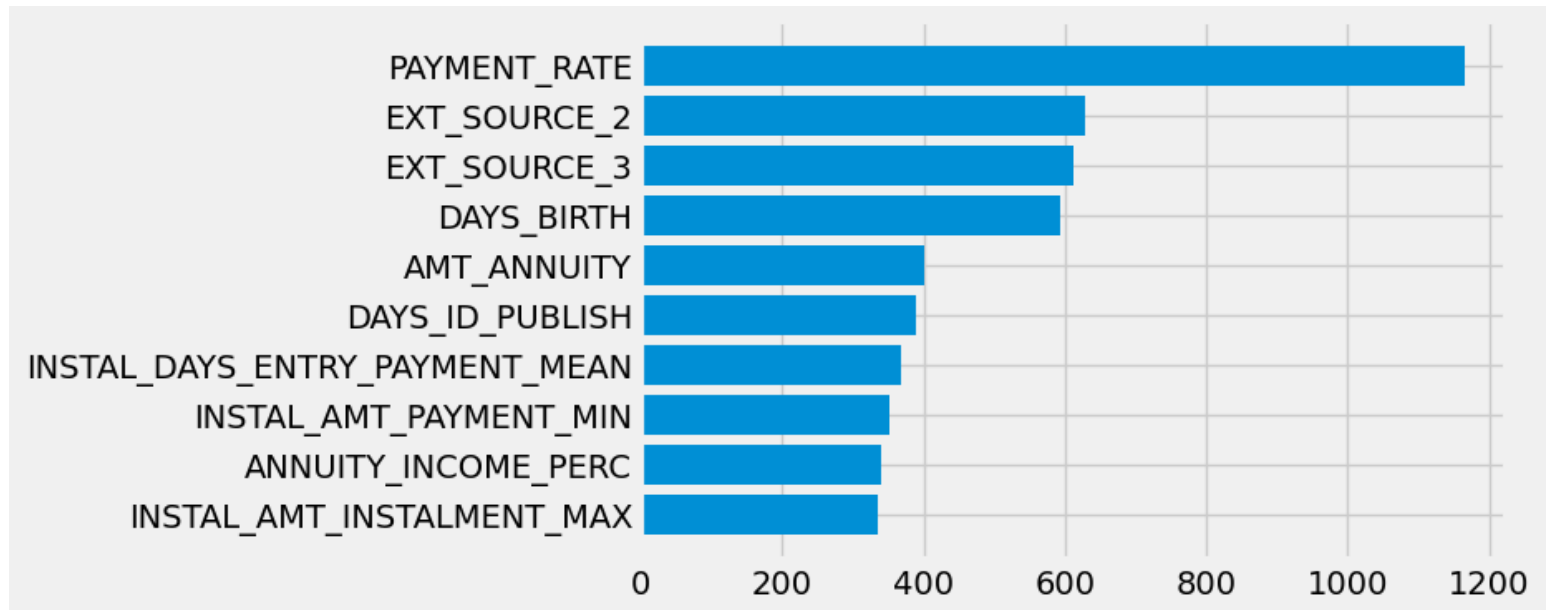
Prêter le plus possible aux « bon clients »

➤ FP (Faux Positif: Faux Mauvais Client)

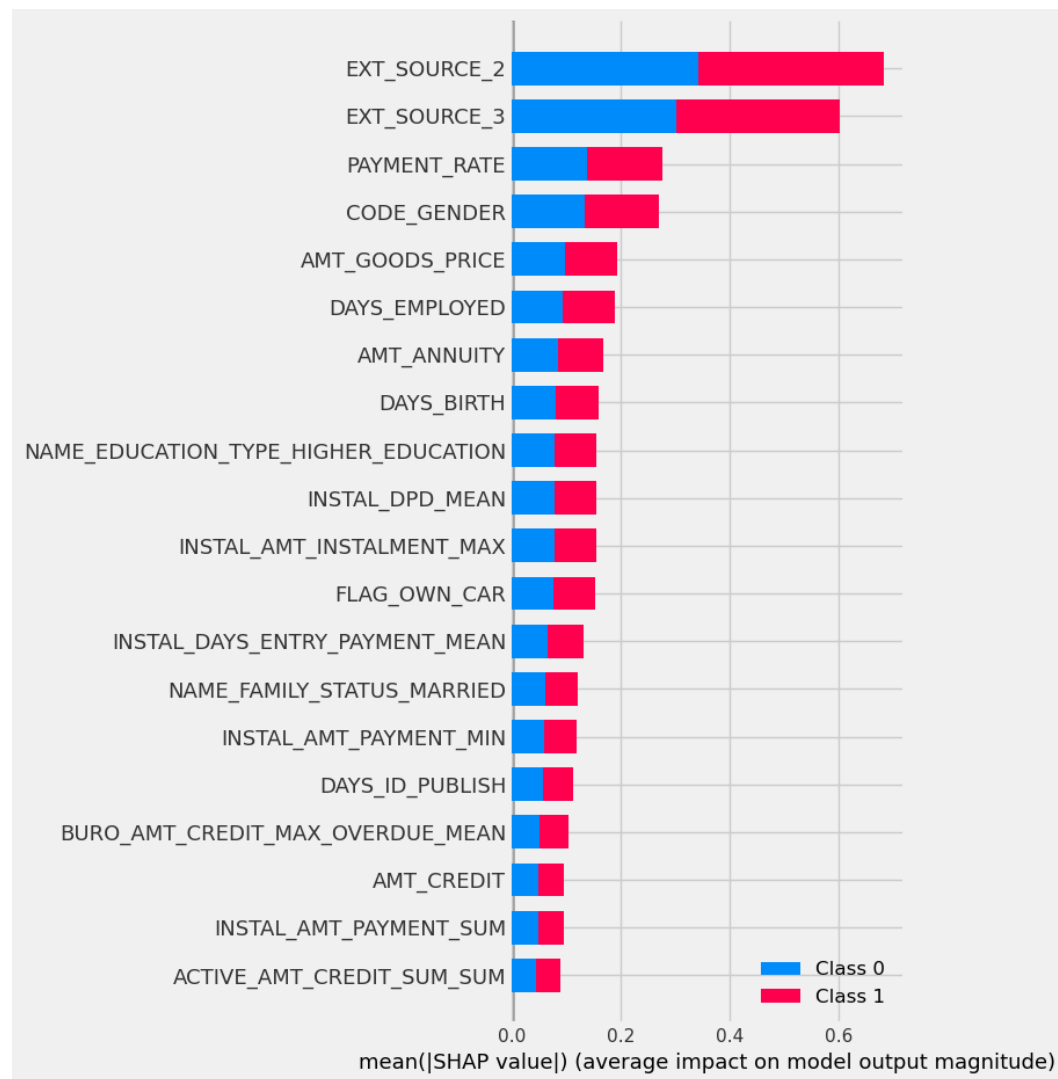
# Feature Importance



# Importance des variables du modèle



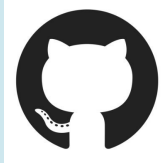


# Analyse SHAP



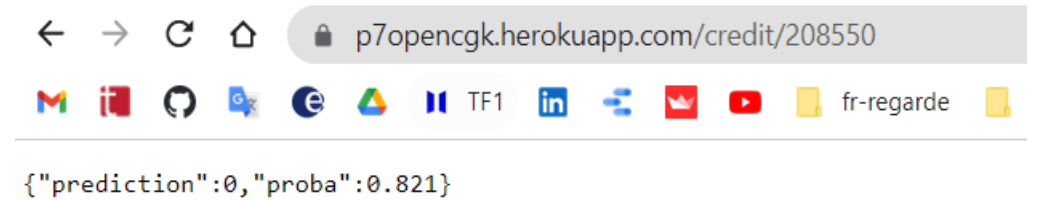
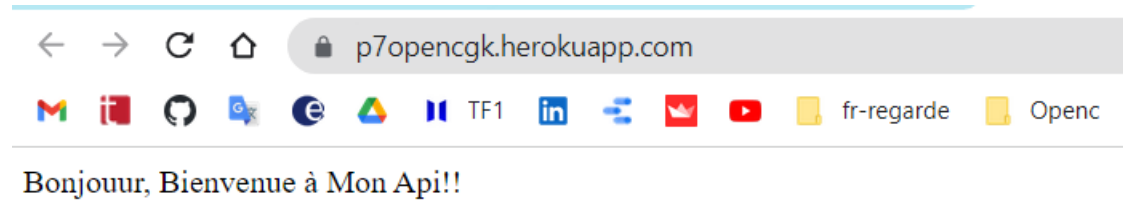
# API

Deploy Cloud

# Outils utilisés

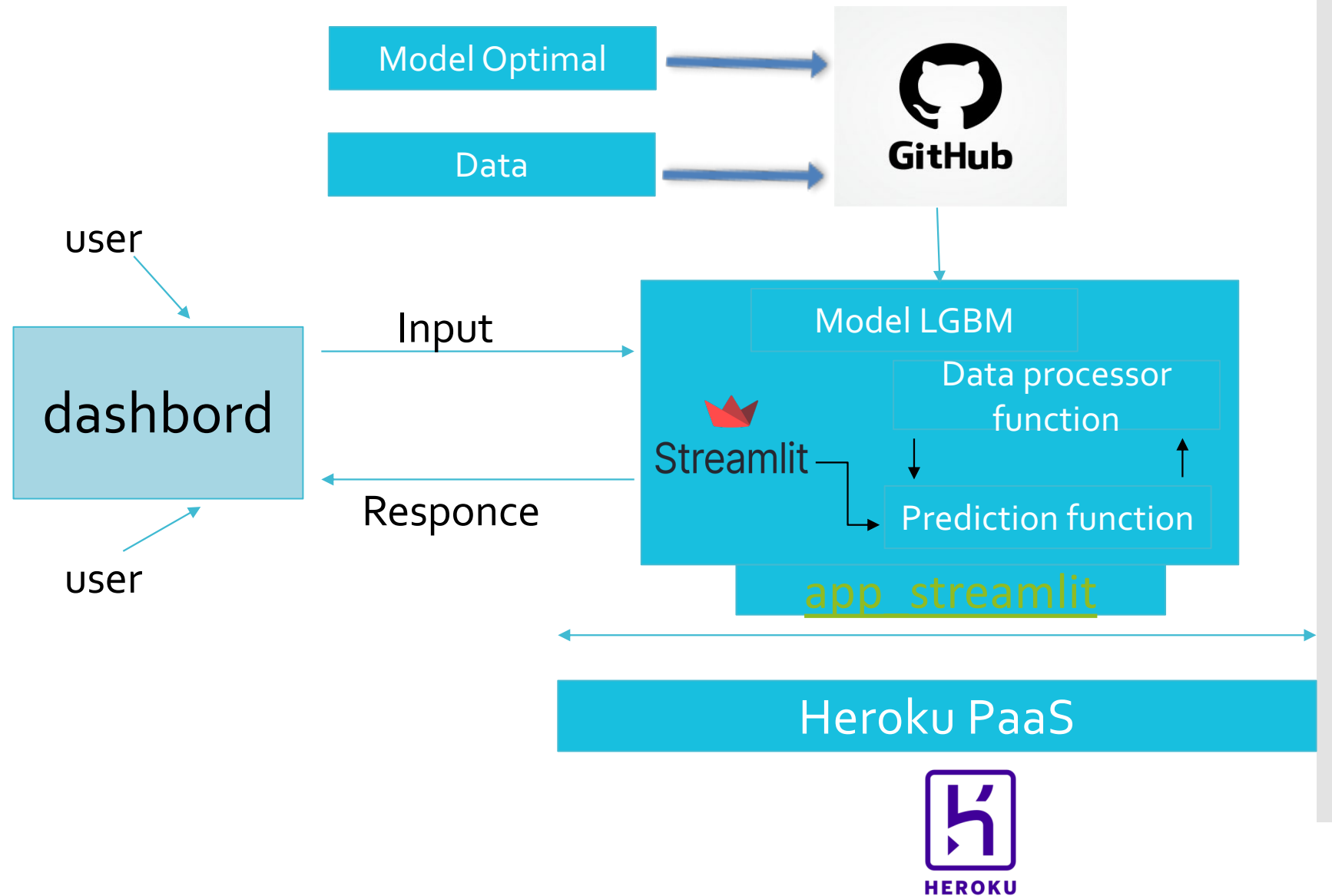
Solution	Description
	<b>Versioning</b>
<b>SHAP</b>	<b>Explicabilité de la prédiction</b>
	<b>API permettant d'appeler la prédiction à partir de l'ID du client</b> <code>127.0.0.1:5000/credit/413394</code>
 <b>Streamlit</b>	<b>Tableau de bord : Front</b>

# Deploy API avec Heroku



Lien de Heroku: <https://p7opencgk.herokuapp.com/>

# SCHÉMA FONCTIONNEL DE L'APPLICATION



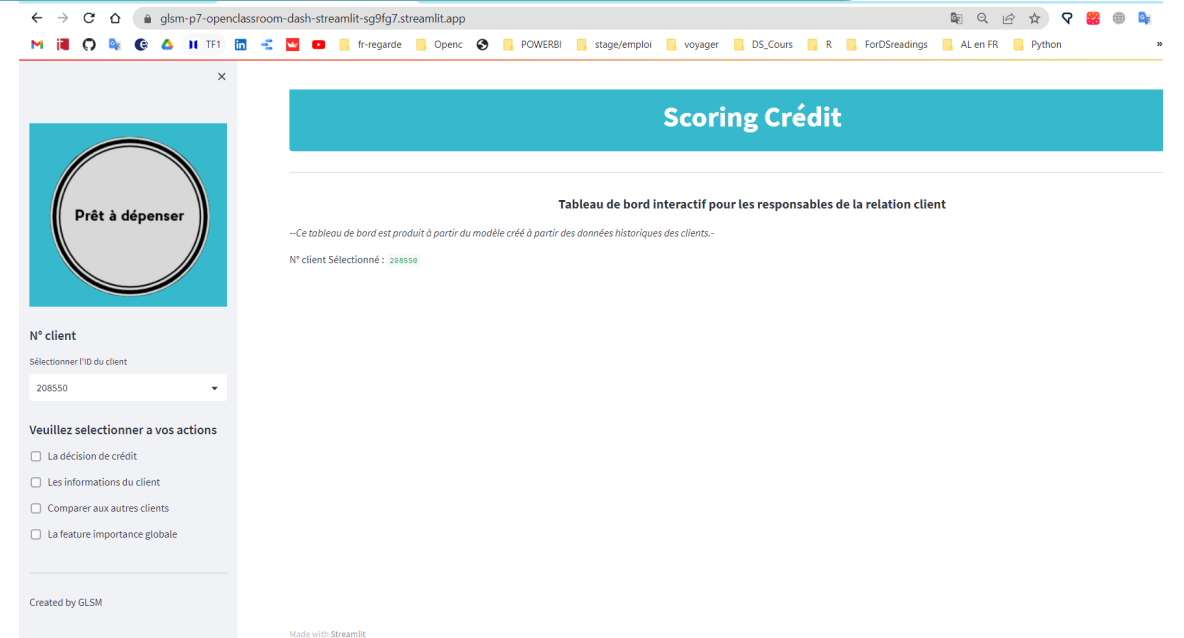
# Dashboard

## Dans ce Dashboard;

**Développement d'un Dashboard interactif** pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions de crédit.

### **Le Dashboard doit permettre de :**

1. Visualiser le score pour chaque client
2. Visualiser des informations descriptives relatives à un client
3. Comparer les informations descriptives relatives à un client à l'ensemble des clients ou à un groupe de clients similaires



# Dashboard & Streamlit

×



Prêt à dépenser

N° client

Sélectionner l'ID du client

365820

Veuillez selectionner a vos actions

☒ La décision de crédit

☐ Les informations du client

☐ Comparer aux autres clients

☐ La feature importance globale

Created by GLSM

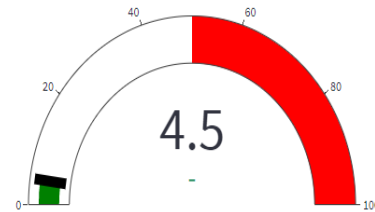
## Scoring Crédit

### Tableau de bord interactif pour les responsables de la relation client

--Ce tableau de bord est produit à partir du modèle créé à partir des données historiques des clients.--

N° client Sélectionné : 365820

### Le score et la décision du modèle de crédit



☐ Afficher les variables ayant le plus contribué à la décision du modèle ?

Risque de défaut: 4.5%

Décision: Acceptation de la demande de prêt pour le client

Made with Streamlit



# Dashboard

Prêt à dépenser

N° client

Sélectionner l'ID du client

365820

Veuillez selectionner a vos actions

☒ La décision de crédit

☒ Les informations du client

☐ Comparer aux autres clients

☐ La feature importance globale

Created by GLSM

4.5

Risque de défaut: 4.5%

Décision: **Acceptation de la demande de prêt pour le client**

☐ Afficher les variables ayant le plus contribué à la décision du modèle ?

Informations relatives au client

Choisir les informations à afficher

GENRE

AGE

STATUT FAMILIAL

NB ENFANTS

REVENUS

MONTANT CREDIT

	365820
GENRE	F
AGE	41
STATUT FAMILIAL	Married
NB ENFANTS	0
REVENUS	135000.0
MONTANT CREDIT	339948.0

☐ Afficher toutes les informations

Lien de Dashboard: <https://glsm-p7-openclassroom-dash-streamlit-mq31bu.streamlit.app/>

# Conclusion

# Conclusion

- i. Le modèle final est **LGBM**, nous avons utilisé le Under Sampling pour faire face au déséquilibre des classes
- ii. Création d'une API web avec Flask pour le côté serveur, et Streamlit pour le côté dashboard

Merci!