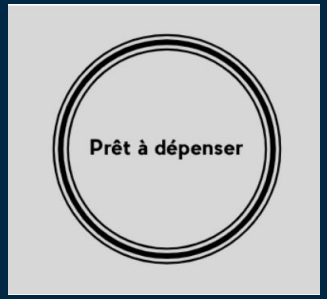
OpenClassrooms
Projet 7 : DATA SCIENCE

Implémentez un modèle de scoring

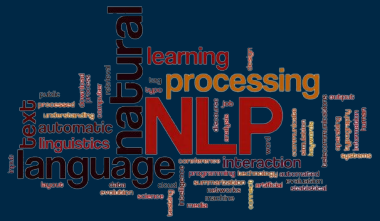
INTRODUCTION

Data Scientist pour la banque
"Prêt à dépenser"



Objectif : Catégoriser les demandes de prêt des nouveaux clients

Classification



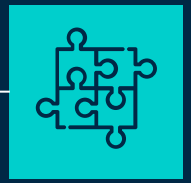
À partir des données clients

Pour répondre à leur demande

Prêt accordé

Prêt refusé

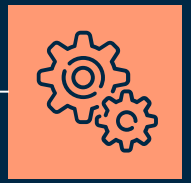
SOMMAIRE



01

ANALYSE EXPLORATOIRE

Découverte du fichier de données



02

ENTRAINEMENT DES MODELES

Choix des modèles, de la métrique et de la technique d'échantillonnage

03

DASHBOARD ET API

Présentation du dashboard Streamlit et de l'API FastAPI

ANALYSE EXPLORATOIRE

01

Découverte du jeu de données

DATAFRAME INITIAL

2 Dataframes :

- Train : 307 511 lignes et 122 colonnes

SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	A
100002	1	Cash loans	M	N	Y	0	202500.0	
100003	0	Cash loans	F	N	N	0	270000.0	
100004	0	Revolving loans	M	Y	Y	0	67500.0	
100006	0	Cash loans	F	N	Y	0	135000.0	

- Target : 48 744 lignes et 121 colonnes

SK_ID_CURR	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT
100001	Cash loans	F	N	Y	0	135000.0	568800.0
100005	Cash loans	M	N	Y	0	99000.0	222768.0
100013	Cash loans	M	Y	Y	0	202500.0	663264.0
100028	Cash loans	F	N	Y	2	315000.0	1575000.0
100038	Cash loans	M	Y	N	1	180000.0	625500.0

ANALYSE EXPLORATOIRE

Dataframe : 307 511 lignes et 122 colonnes

Données dupliquées	Données manquantes
0	25% au total

Colonnes utiles pour classification :

Corrélations positives

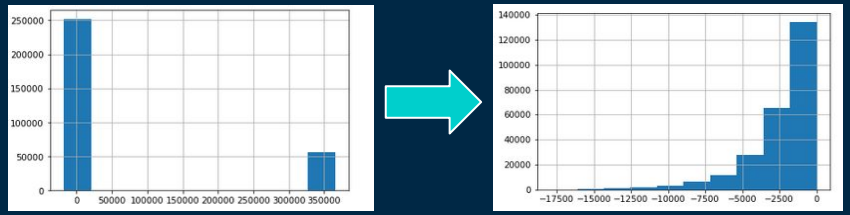
- DAYS_BIRTH
- DAYS_EMPLOYED
- REGION_RATING_CLIENT_W_CITY
- REGION_RATING_CLIENT
- NAME_INCOME_TYPE_Working
- DAYS_LAST_PHONE_CHANGE

Corrélations négatives

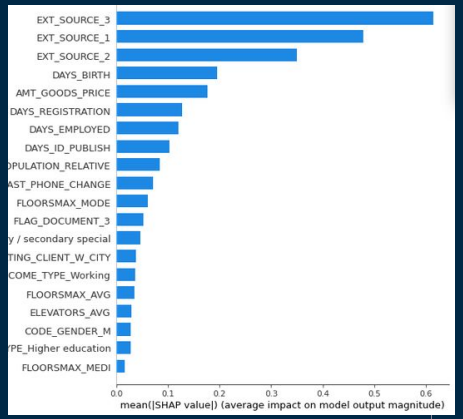
- EXT_SOURCE_3
- EXT_SOURCE_2
- EXT_SOURCE_1
- NAME_EDUCATION_TYPE_Higher education
- CODE_GENDER_F
- NAME_INCOME_TYPE_Pensioner

Préparation du dataframe

Gestion des anomalies



Réduction de dimension



Label Encoding

State (Nominal Scale)	State (Label Encoding)
Maharashtra	3
Tamil Nadu	4
Delhi	0

One-Hot Encoding

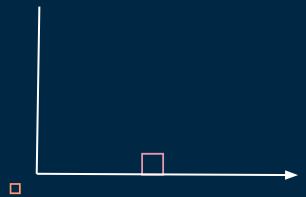
Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

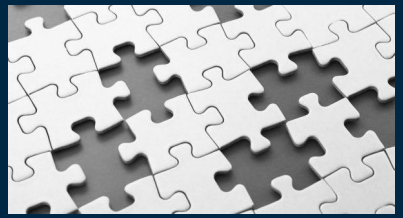
DATAFRAME FINAL

DAYS_BIRTH	DAYS_EMPLOYED	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	NAME_INCOME_TYPE_Working
-19241	-2329.0	2	2	1
-18064	-4469.0	2	2	1
-20038	-4458.0	2	2	1
-13976	-1866.0	2	2	1
-13040	-2191.0	2	2	1

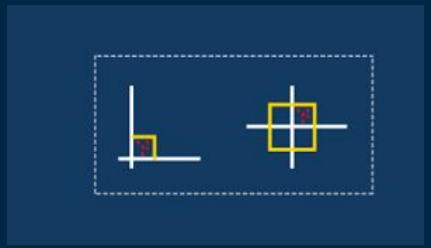
48 744 lignes et 31 colonnes



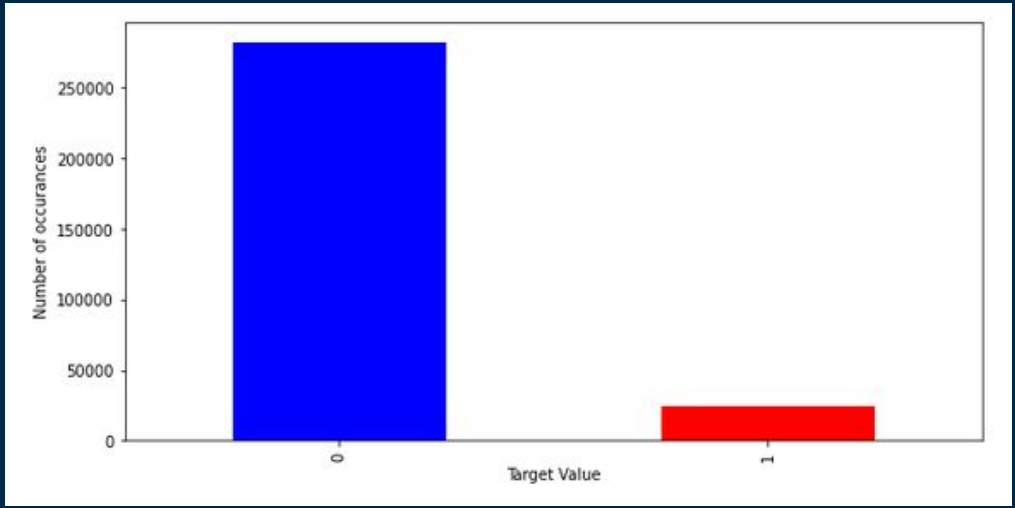
Imputer



Scaler



VISUALISATION DE L'OBJECTIF



La variable 'TARGET' est fortement déséquilibrée

Métrique : ~~Accuracy~~

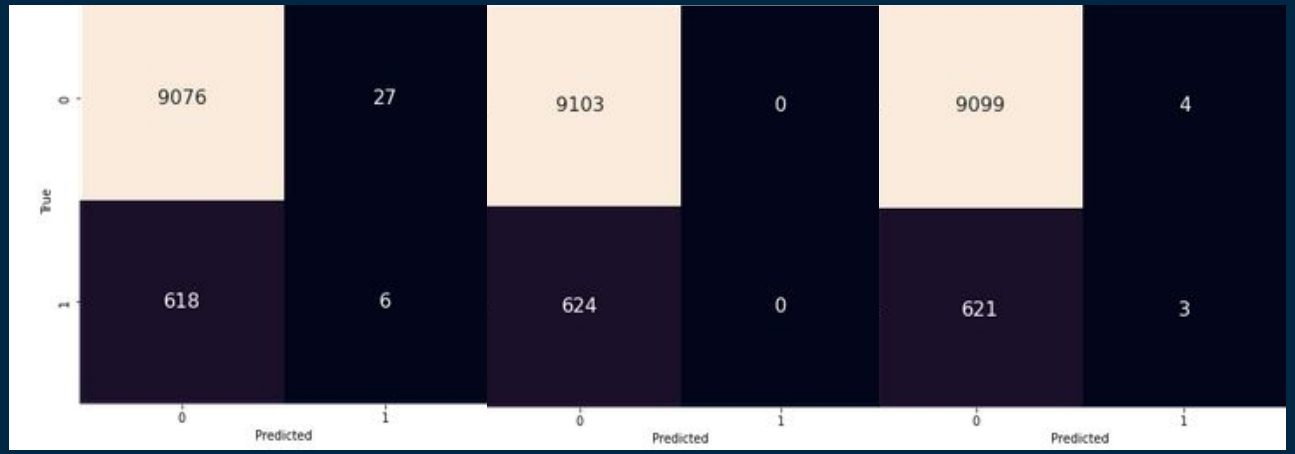
ENTRAINEMENT DES MODELES

02

Choix des modèles, de la métrique et de la technique d'échantillonnage

PREMIERS RESULTATS

LGBM | Ridge | Random Forest



Précision	0.182	0	0.429
Rappel	0.010	0	0.005

- Accuracy élevée

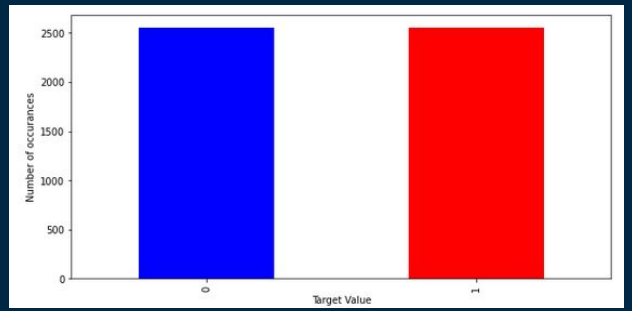
⇒ 90% en moyenne

- Précision et rappel très faibles

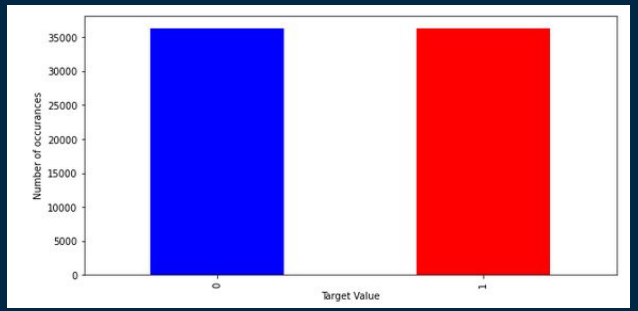
Mauvaise prédiction

TECHNIQUE D'ÉCHANTILLONNAGE

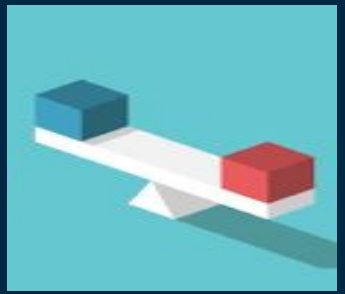
Under-sampling



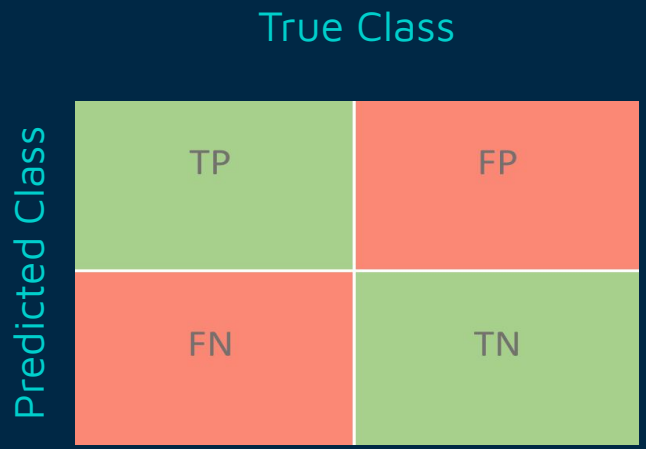
Over -sampling



Class-Weight



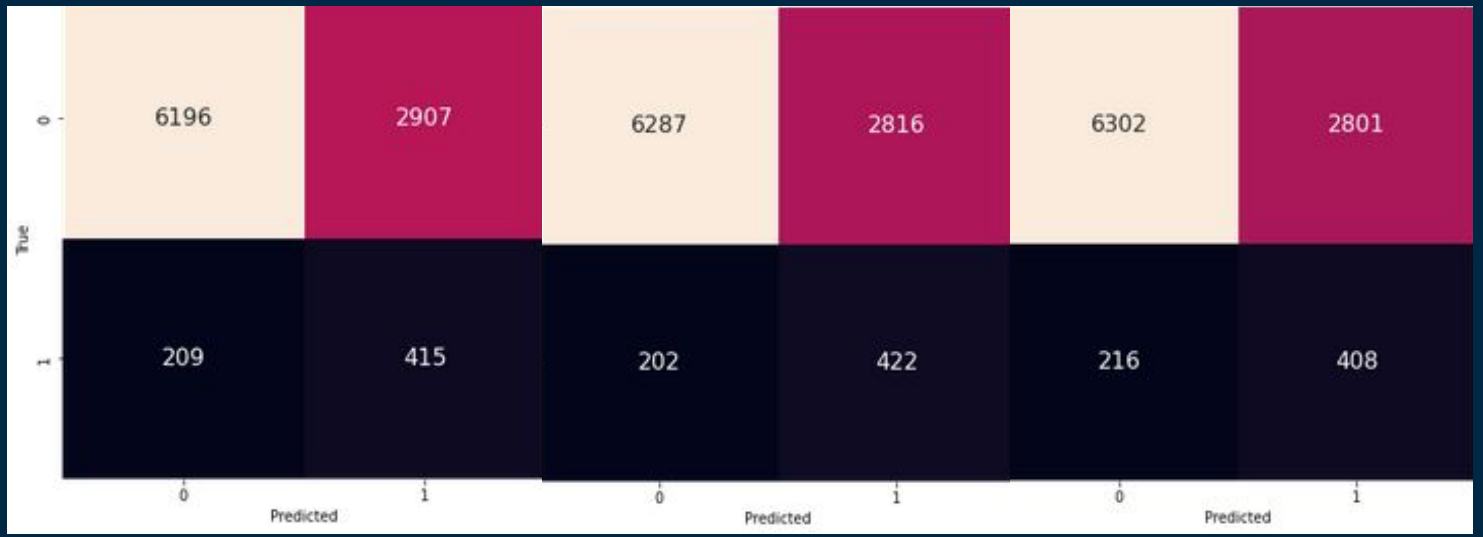
CHOIX DE LA MÉTRIQUE



- Minimiser le taux de mauvais payeurs
- Maximiser le taux de bons payeurs

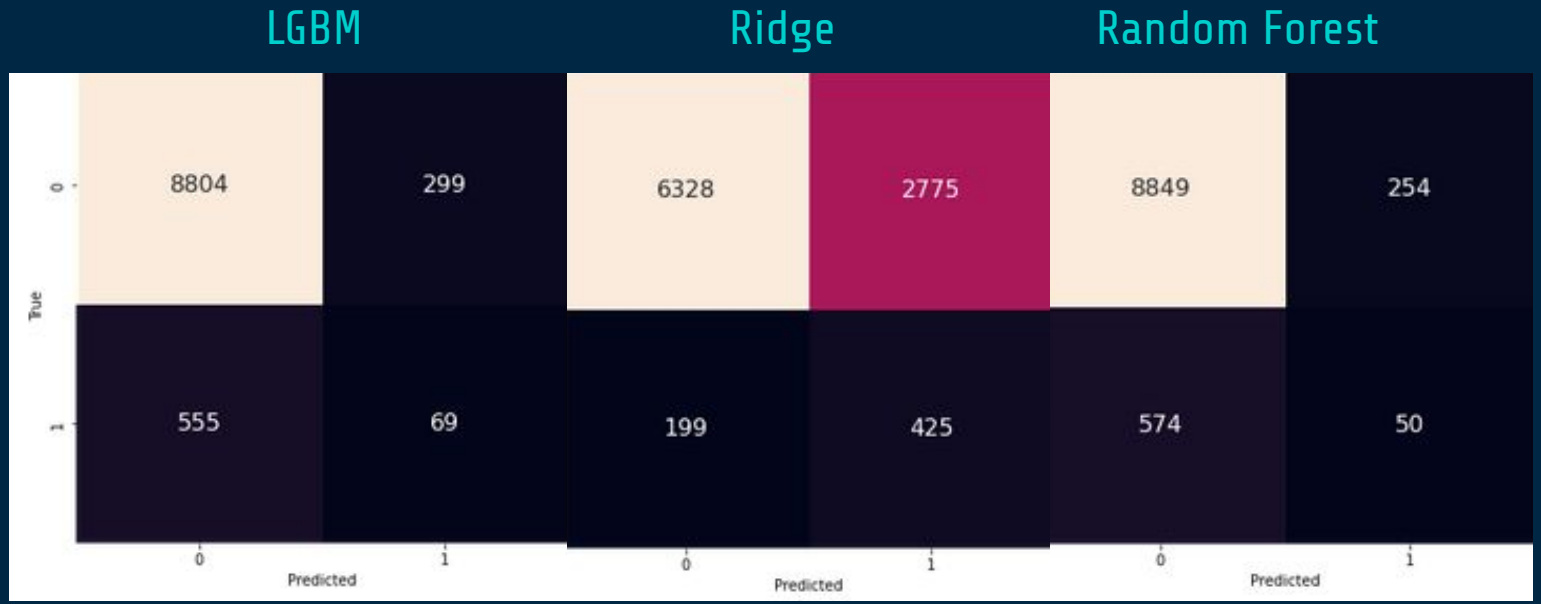
Choix de la métrique : Fβ-Score
avec β = 3

RESULTATS Under-Sampling



FB-Score	0.464	0.477	0.462

RESULTATS SMOTE



FB-Score	0.115	0.482	0.084

RESULTATS CLASS-WEIGHT

FB-Score	0.447	0.482	0.002

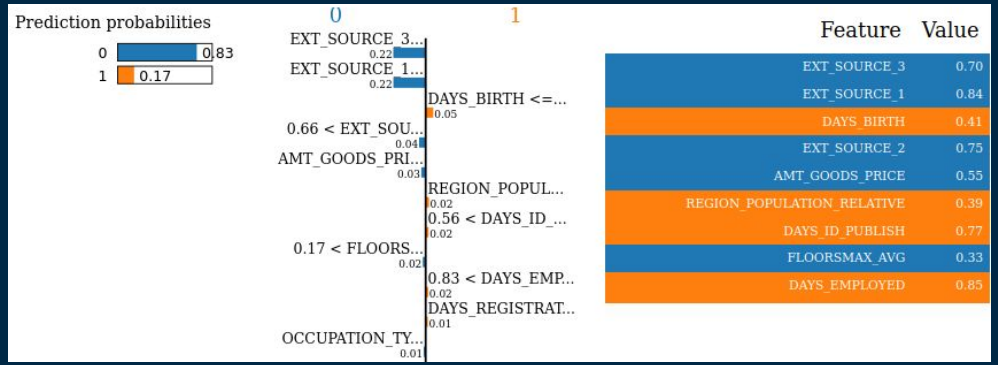
MODELE CHOISI

Modèle	Under-sampling	SMOTE	Class-Weight
LGBM	0.464	0.115	0.447
Random Forest	0.462	0.084	0.002
Ridge	0.477	0.482	0.482

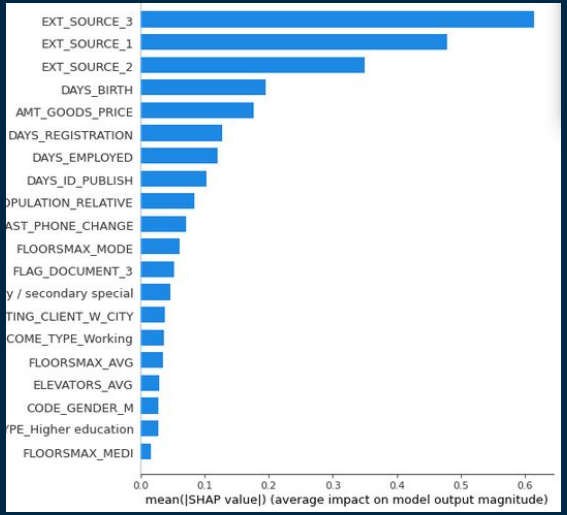
Ridge meilleur score mais LGBM meilleure interprétabilité

INTERPRETABILITE

LIME



SHAP



ENTRAINEMENT DE MODELE

```
my_estimator = lgbm.LGBMClassifier(random_state = 42)
my_params = {
    'learning_rate': [0.05, 0.1, 0.15],
    'n_estimators': [50, 100, 150],
    'num_leaves': [25, 31, 37], # large num_leaves helps improve accuracy but might lead to over-fitting
    'boosting_type' : ['gbdt', 'dart'], # for better accuracy -> try dart
    'objective' : ['binary', None],
    'max_bin':[255, 510], # large max_bin helps improve accuracy but might slow down training progress
    'random_state' : [42],
    'colsample_bytree' : [0.99, 1, 1.01],
    'subsample' : [0.9, 1.0]
    }
```

```
: my_gridsearch = GridSearchCV(estimator=my_estimator, param_grid=my_params, scoring=my_scorer, cv=cv)
  my_gridsearch.fit(X_resampled, y_resampled)
```

```
GridSearchCV(cv=RepeatedKFold(n_repeats=3, n_splits=10, random_state=42),
             estimator=LGBMClassifier(random_state=42),
             param_grid={'boosting_type': ['gbdt', 'dart'],
                         'colsample_bytree': [0.99, 1, 1.01],
                         'learning_rate': [0.05, 0.1, 0.15],
                         'max_bin': [255, 510], 'n_estimators': [50, 100, 150],
                         'num_leaves': [25, 31, 37],
                         'objective': ['binary', None], 'random_state': [42],
                         'subsample': [0.9, 1.0]},
             scoring=make_scorer(fbeta_score, beta=3))
```

FB-Score = 0.464

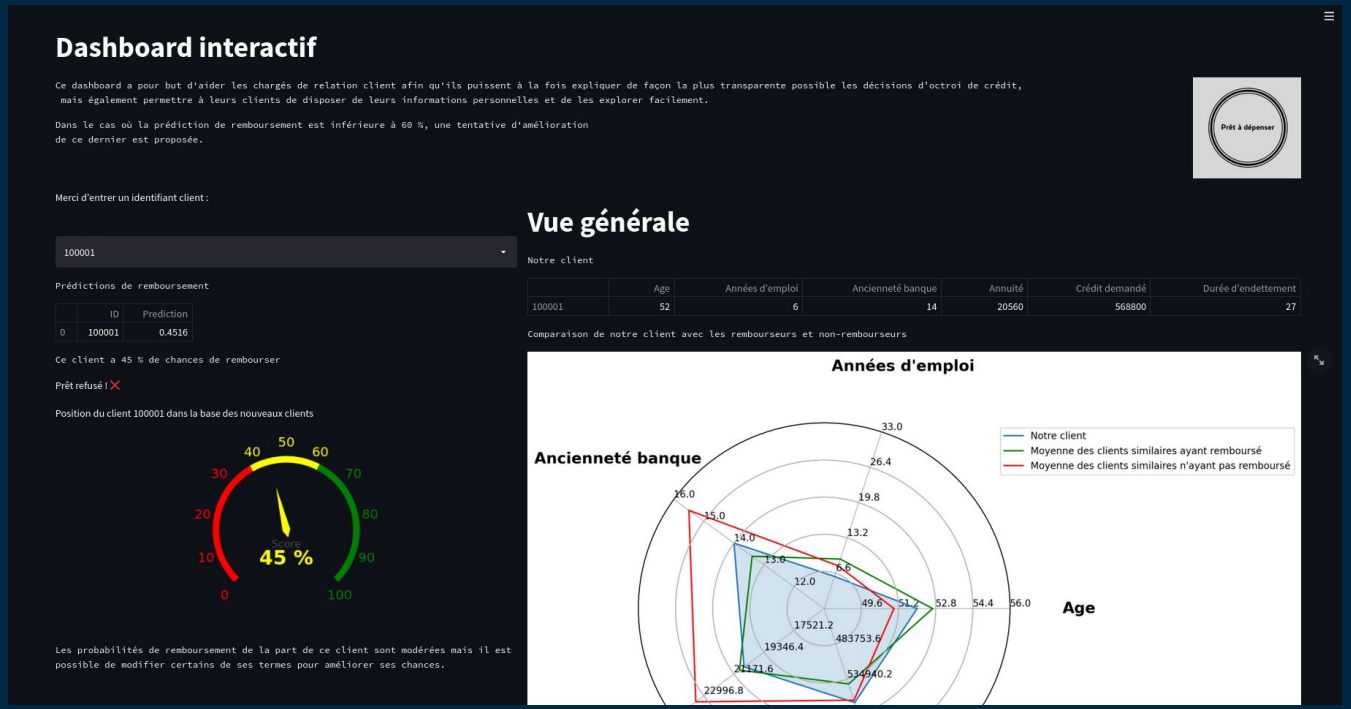
DASHBOARD ET API

03

Streamlit & FastAPI

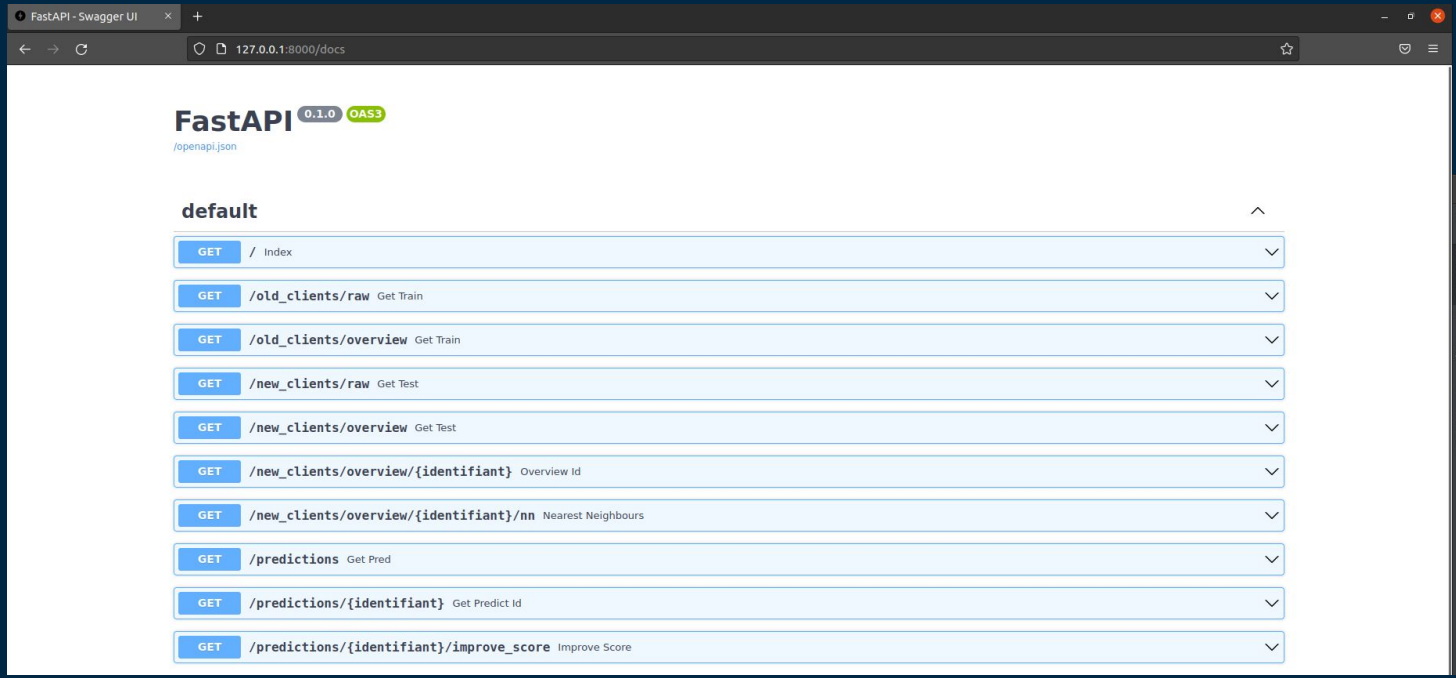
STREAMLIT

http://localhost:8501/



FastAPI

http://127.0.0.1:8000/



CONCLUSION

- β=3 à confirmer
- + de caractéristiques exploitables
- Limité par l'interprétabilité

MERCI POUR VOTRE ATTENTION !