PROJET 7
Implémentez un modèle de scoring

Problématique

Vous êtes Data Scientist au sein d'une société financière, nommée "Prêt à dépenser", qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.

L’entreprise 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 en s’appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

De plus, les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeurs de transparence vis-à-vis des décisions d’octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l’entreprise veut incarner.

Elle décide donc de développer un dashboard interactif pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d’octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

Données et Mission

Les données sont issues d'un dataset sur kaggle :

https://www.kaggle.com/c/home-credit-default-risk/data

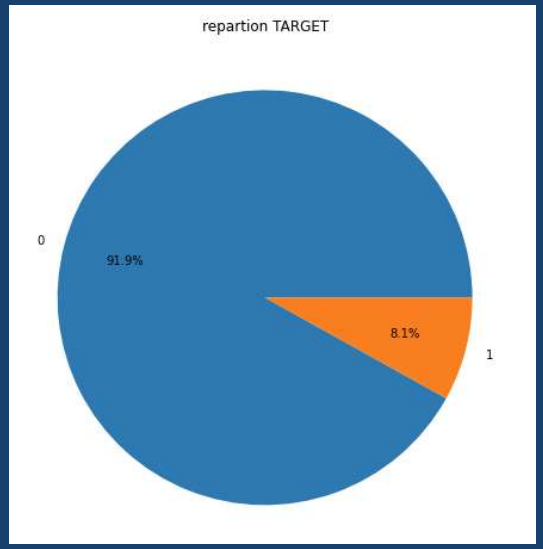
Votre mission

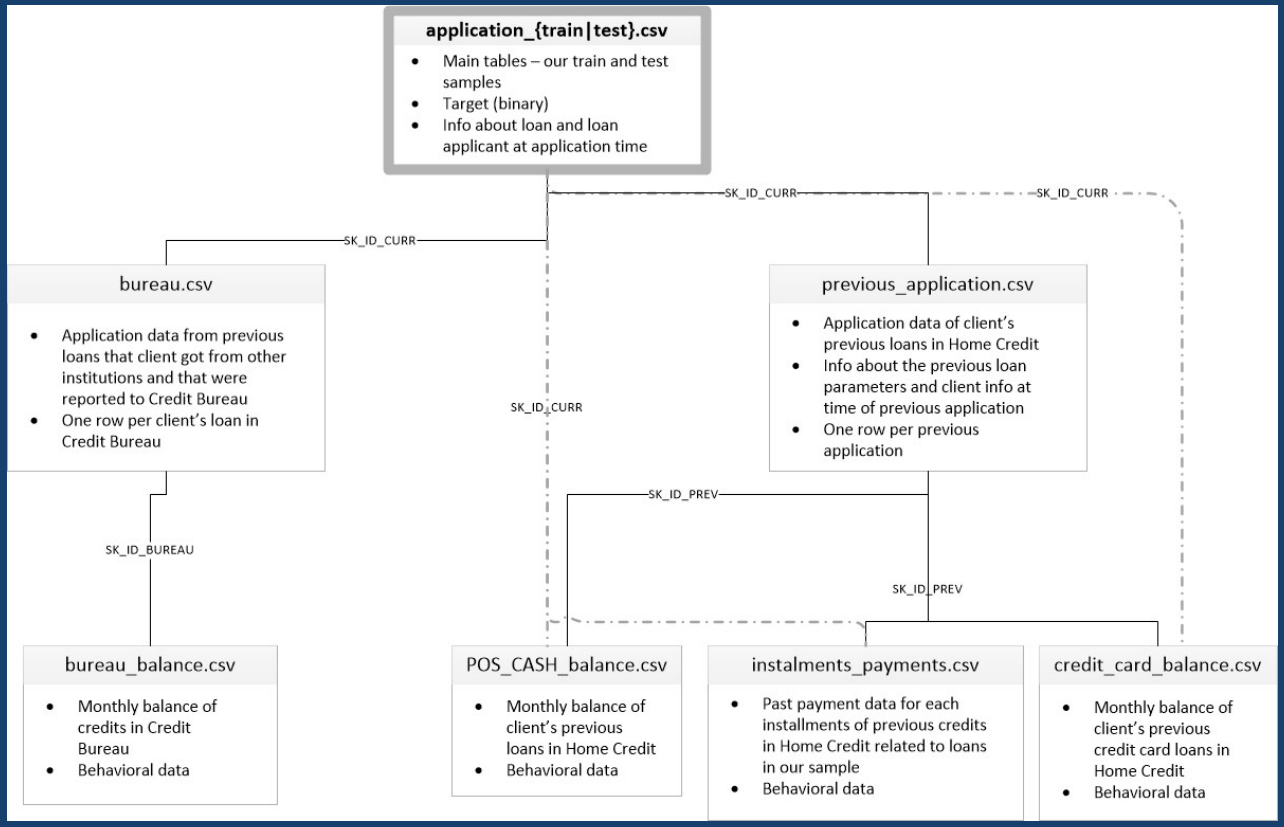
Construire un modèle de scoring qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique.

Construire un dashboard interactif à destination des gestionnaires de la relation client permettant d'interpréter les prédictions faites par le modèle et d'améliorer la connaissance client des chargés de relation client.

Le jeu de donnée

Issu de kaggle





Le jeu de donnée

Sélection des variables à partir d'un kernel kaggle :

https://www.kaggle.com/hikmetsezen/micro-model-174-features-0-8-auc-on-home-credit

Feature engineering :

- one hot encoding pour les variables catégorielles
- Création de variables supplémentaires (ratio divers)
- Agrégation des autres tables par numéro de client (somme, moyenne, max, min, count)

Cela donne un jeu de données de 1242 variables.

Sélection dans celle-ci de 173 variables explicables pour le client et contribuant au modele.

Le jeu de donnée

Sélection des variables à partir d'un kernel kaggle :

https://www.kaggle.com/hikmetsezen/micro-model-174-features-0-8-auc-on-home-credit

Feature engineering :

- one hot encoding pour les variables catégorielles
- Création de variables supplémentaires (ratio divers)
- Agrégation des autres tables par numéro de client (somme, moyenne, max, min, count)

Cela donne un jeu de données de 1242 variables.

Sélection dans celle-ci de 173 variables explicables pour le client et contribuant au modele.

Modélisation

Jeu de données déséquilibré 92% de 0 pour 8% de 1.

Utilisation de la librairie imblearn pour sur-échantillonner, sous-échantilloner.

Séparation en jeu d'entrainement (246004 lignes) et validation (61502 lignes)

Essai avec régression logistique, Arbres de décisions, Random Forest et Gradient Boosting(lightGBM)

Utilisation d'une métrique F2-score qui permet de minimiser les Faux négatifs qui sont les plus pénalisant dans notre modélisation pour la validation des modèles.

Pour entrainer les modèles j'ai utilise la fonction de cout AUC Score (Area Under the ROC Curve) pour le gradient boosting.

J'ai affiché aussi la matrice de confusion et differents scores (precision, recall, f1, FMI) pour chaque modèle.

Modélisation

Rappel matrice confusion :

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

Rappel precision, recall, Fbeta-measure :

- Precision = TruePositives / (TruePositives + FalsePositives)
- Recall = TruePositives / (TruePositives + FalseNegatives)
- Fbeta = ((1 + beta^2) * Precision * Recall) / (beta^2 * Precision + Recall)

3 valeurs communes pour Fbeta :

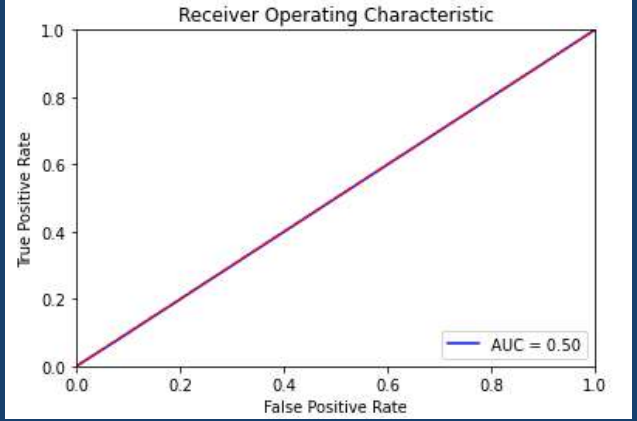
- F0.5-Measure (beta=0.5): precision plus important que recall
- F1-Measure (beta=1.0): équilibre entre precision et recall.
- F2-Measure (beta=2.0): recall plus d'importance que precision

Dans notre cas on recherche a minimiser les FN et donc maximisé le recall tout en essayant de garder une précision acceptable

Modélisation

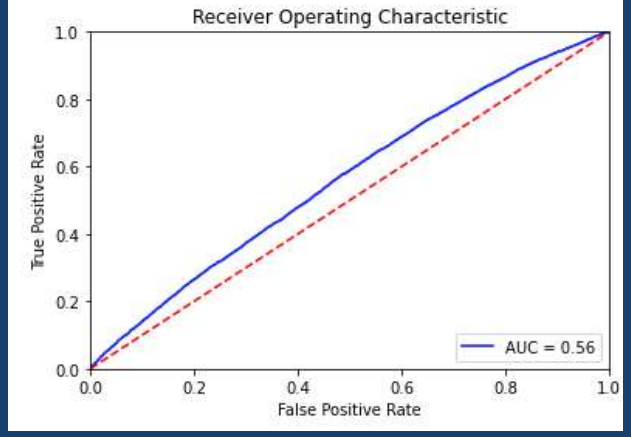
DummyClassifier

```
[[51928  4609]
 [ 4610   355]]
Accuracy score    : 0.85
precision score   : 0.0715
recall score      : 0.0715
F1 score          : 0.0715
F2 score          : 0.0715
ROCAUC score      : 0.495
custom metric FMI : 0.0715
```



Regression Logistique

```
[[56528     9]
 [ 4965     0]]
Accuracy score    : 0.919
precision score   : 0.0
recall score      : 0.0
F1 score          : 0.0
F2 score          : 0.0
ROCAUC score      : 0.5
custom metric FMI : 0.0
```

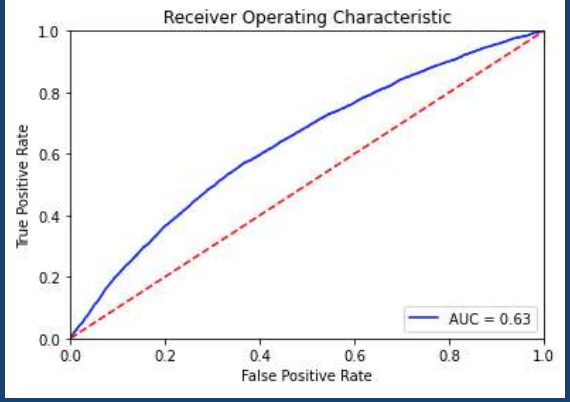


Modélisation LOGISTIC REGRESSION

Essai sous-échantillonage, sur-échantillonage départ (0: 226145, 1: 19859)

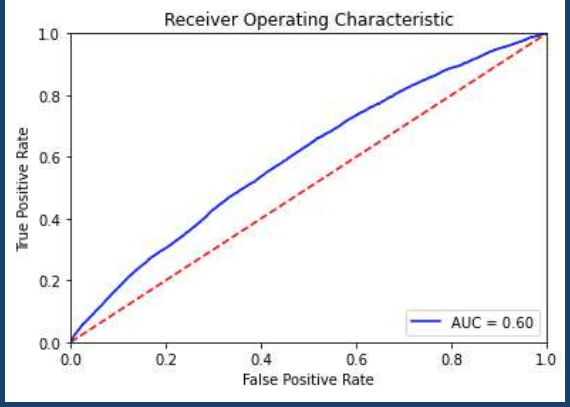
Under (0: 19859, 1: 19859)

```
[[32362 24175]
 [ 1882  3083]]
Accuracy score    : 0.576
precision score   : 0.113
recall score      : 0.621
F1 score          : 0.191
F2 score          : 0.327
ROCAUC score      : 0.597
custom metric FMI : 0.265
```



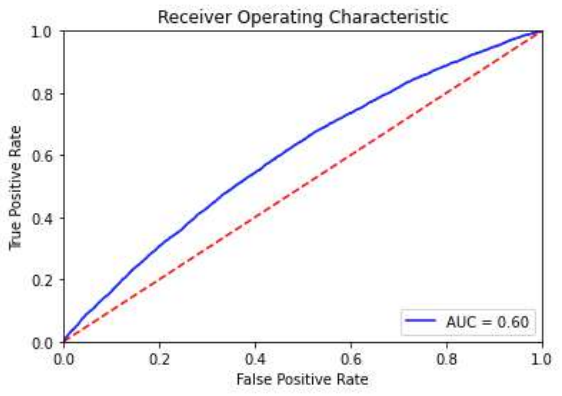
Over SMOTE(0: 226145, 1: 113072)

```
[[56240   297]
 [ 4890    75]]
Accuracy score    : 0.916
precision score   : 0.202
recall score      : 0.0151
F1 score          : 0.0281
F2 score          : 0.0185
ROCAUC score      : 0.505
custom metric FMI : 0.0552
```



SMOTEENN (0: 133326, 1: 217036)

```
[[10206 46331]
 [  496  4469]]
Accuracy score    : 0.239
precision score   : 0.088
recall score      : 0.9
F1 score          : 0.16
F2 score          : 0.316
ROCAUC score      : 0.54
custom metric FMI : 0.281
```

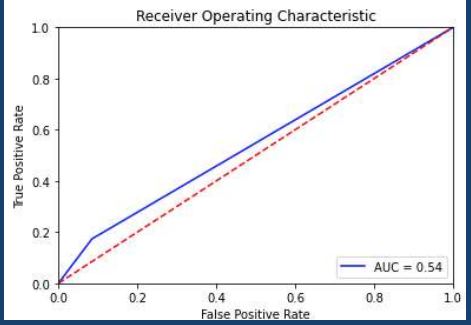


Modélisation DECISION TREE

sous-échantillonage, sur-échantillonage départ (0: 226145, 1: 19859)

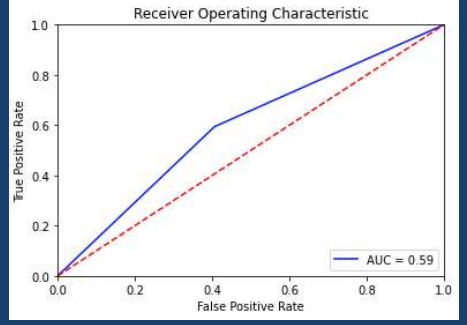
Normal

```
[[51737  4800]
 [ 4104   861]]
Accuracy score    : 0.855
precision score   : 0.152
recall score      : 0.173
F1 score          : 0.162
F2 score          : 0.169
ROCAUC score      : 0.544
custom metric FMI : 0.162
```



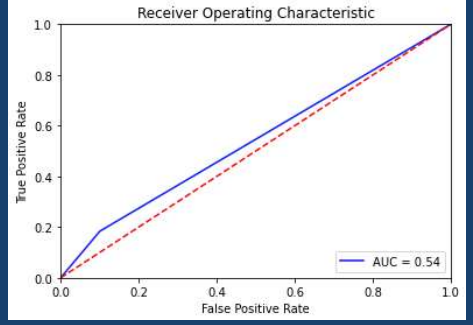
Under (0: 19859, 1: 19859)

```
[[33582 22955]
 [ 2015  2950]]
Accuracy score    : 0.594
precision score   : 0.114
recall score      : 0.594
F1 score          : 0.191
F2 score          : 0.322
ROCAUC score      : 0.594
custom metric FMI : 0.26
```



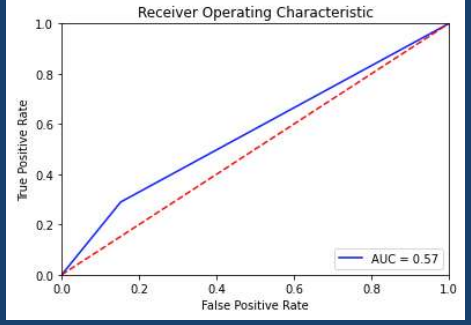
Over SMOTE(0: 226145, 1: 113072)

```
[[50854  5683]
 [ 4052   913]]
Accuracy score    : 0.842
precision score   : 0.138
recall score      : 0.184
F1 score          : 0.158
F2 score          : 0.173
ROCAUC score      : 0.542
custom metric FMI : 0.16
```



SMOTEENN (0: 133326, 1: 217036)

```
[[47914  8623]
 [ 3527  1438]]
Accuracy score    : 0.802
precision score   : 0.143
recall score      : 0.29
F1 score          : 0.191
F2 score          : 0.24
ROCAUC score      : 0.569
custom metric FMI : 0.203
```

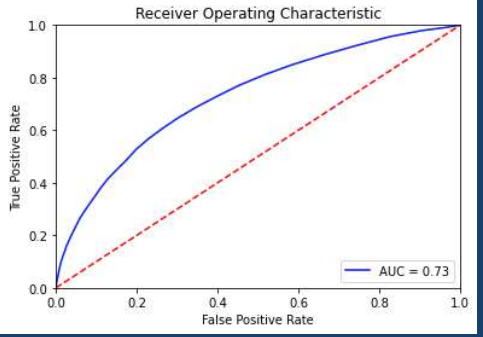


Modélisation RANDOM FOREST

sous-échantillonage, sur-échantillonage départ (0: 226145, 1: 19859)

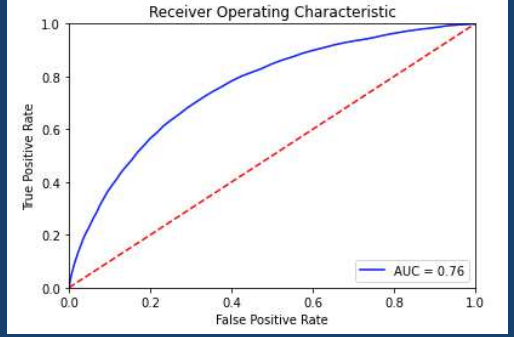
Normal

```
[[56533     4]
 [ 4955    10]]
Accuracy score     : 0.919
precision score    : 0.714
recall score       : 0.00201
F1 score           : 0.00402
F2 score           : 0.00252
ROCAUC score       : 0.501
custom metric FMI : 0.0379
```



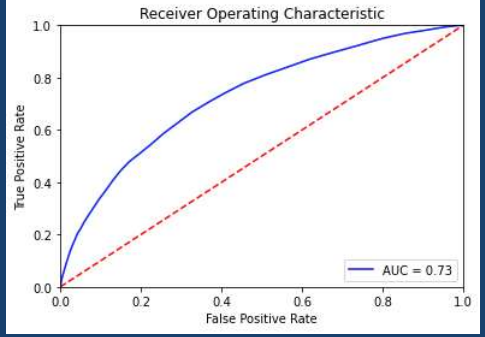
Under (0: 19859, 1: 19859)

```
[[40137 16400]
 [ 1590  3375]]
Accuracy score     : 0.707
precision score    : 0.171
recall score       : 0.68
F1 score           : 0.273
F2 score           : 0.426
ROCAUC score       : 0.695
custom metric FMI : 0.341
```



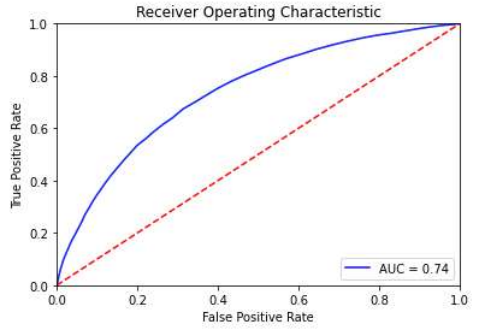
Over SMOTE(0: 226145, 1: 113072)

```
[[56523    14]
 [ 4945    20]]
Accuracy score     : 0.919
precision score    : 0.588
recall score       : 0.00403
F1 score           : 0.008
F2 score           : 0.00503
ROCAUC score       : 0.502
custom metric FMI : 0.0487
```



SMOTEENN (0: 133326, 1: 217036)

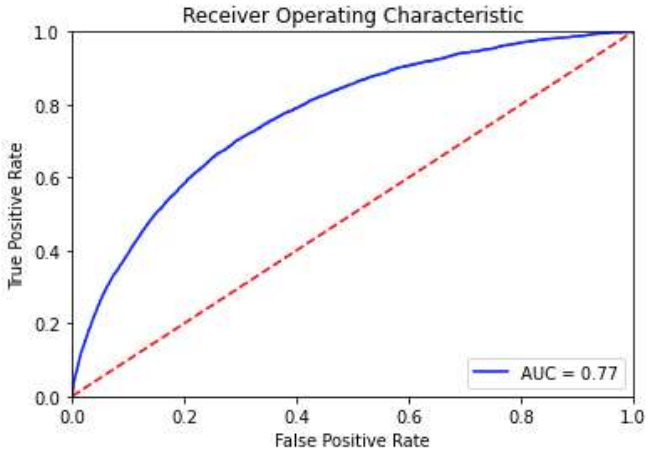
```
[[55799   738]
 [ 4573   392]]
Accuracy score     : 0.914
precision score    : 0.347
recall score       : 0.079
F1 score           : 0.129
F2 score           : 0.0934
ROCAUC score       : 0.533
custom metric FMI : 0.165
```



Modélisation Random Forest hyperparamètres

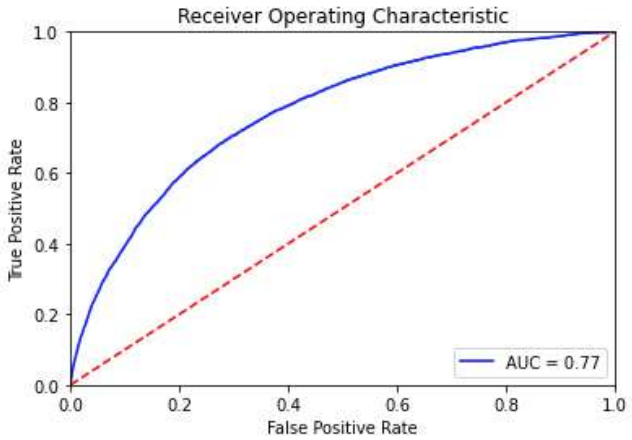
RandomizedSearchCV

```
[[40080 16457]
 [ 1495  3470]]
Accuracy score     : 0.708
precision score    : 0.174
recall score       : 0.699
F1 score           : 0.279
F2 score           : 0.436
ROCAUC score       : 0.704
custom metric FMI : 0.349
```



GridSearchCV

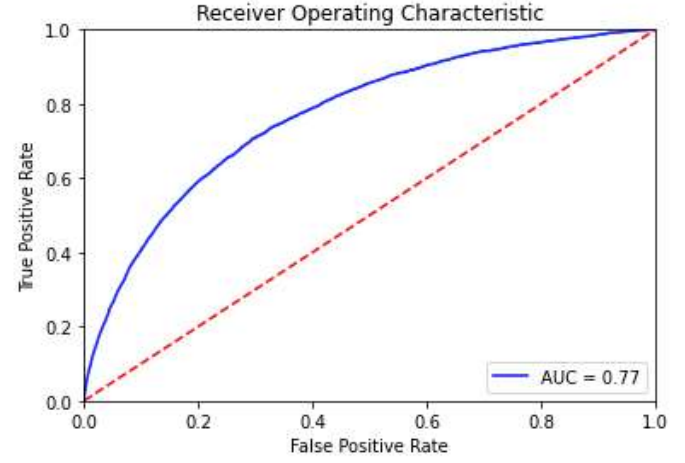
```
[[40016 16521]
 [ 1492  3473]]
Accuracy score     : 0.707
precision score    : 0.174
recall score       : 0.699
F1 score           : 0.278
F2 score           : 0.436
ROCAUC score       : 0.704
custom metric FMI : 0.349
```



Modélisation LightGBM hyperparamètres

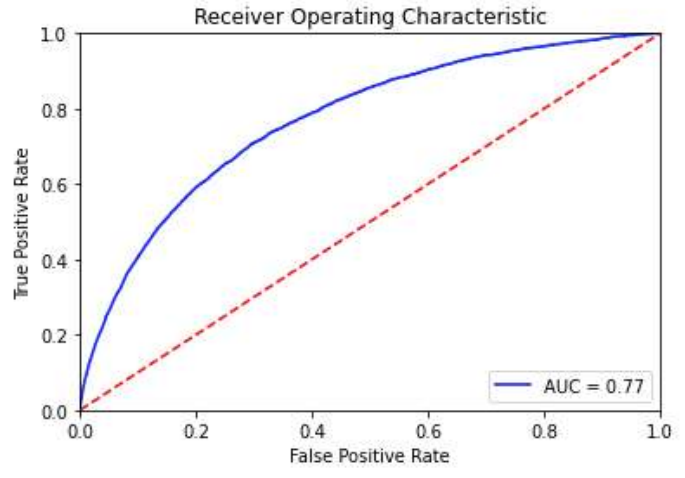
RandomizedSearchCV

```
[[40047 16490]
 [ 1488  3477]]
Accuracy score    : 0.708
precision score   : 0.174
recall score      : 0.7
F1 score          : 0.279
F2 score          : 0.437
ROCAUC score      : 0.704
custom metric FMI : 0.349
```



GridSearchCV

```
[[40047 16490]
 [ 1488  3477]]
Accuracy score    : 0.708
precision score   : 0.174
recall score      : 0.7
F1 score          : 0.279
F2 score          : 0.437
ROCAUC score      : 0.704
custom metric FMI : 0.349
```



API et DASHBOARD

Demonstration :

API FLASK :

https://p7api.herokuapp.com/predict/

DASHBOARD STREAMLIT :

https://limitless-springs-15363.herokuapp.com/

MERCI

Questions et Réponses