Analyse

Prêt à dépenser

Implémentez un modèle de scoring

Sofiane Mouhab

Septembre 2021

Sommaire

1 - Généralités

1.1 - Problématique

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.

1.2 - Objectif

L’objectif est l’implémentation d'un modèle d'apprentissage supervisé pour une application de Crédit Scoring suivant plusieurs parametres:

- Le modèle doit permettre de définir la probabilité de défaut de remboursement d'un crédit à partir des informations sur le client

- Il doit également offrir un certain niveau de transparence afin de permettre aux conseillers de justifier la réponse

- 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.

1.3 - Condition de mise en oeuvre

Pour pouvoir sereinement réaliser ses quatres objectifs, il nous faut donc diverses informations qui pourrait se trouver dans notre base de données.

À nous donc, d'examiner celle-ci, de déterminer à quel point les informations sont viables, ou perfectible.

Il y a donc 2 grandes interrogations :

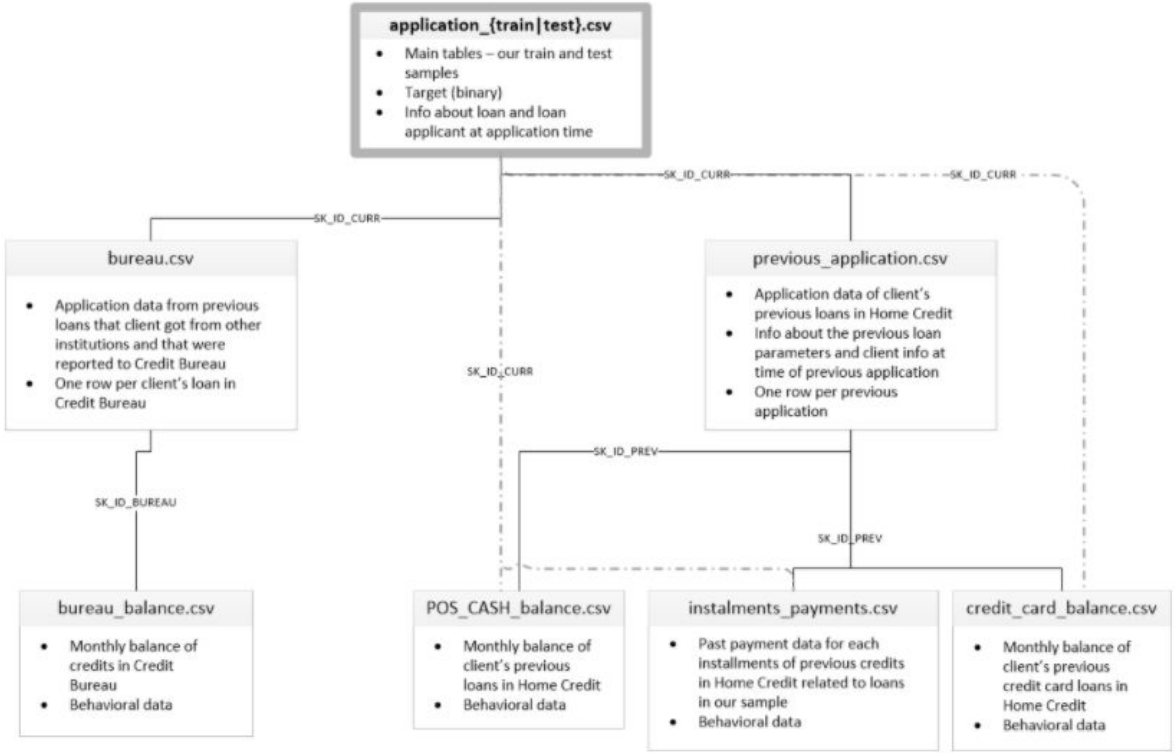
- A-t-on assez de données ?
- Peut-on faire des prévisions cohérente ?

Passons de suite à ce travail, en commençant par rapidement prendre connaissance des données en présence...

2 - Les données

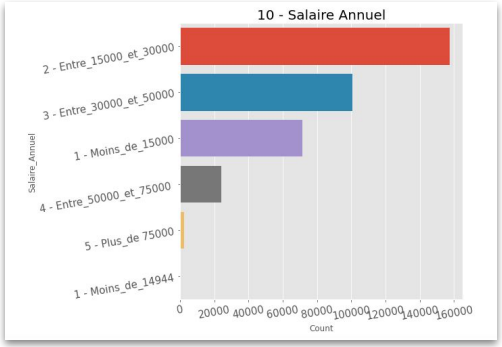
2.1 - Description

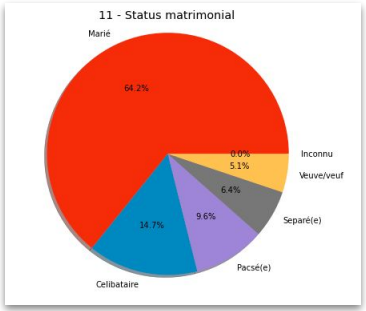
Dans ce fichier volumineux nous comptons 2 Fichiers , les relevés pour 2015 et 2016, voyons ce qu'ils contiennent :

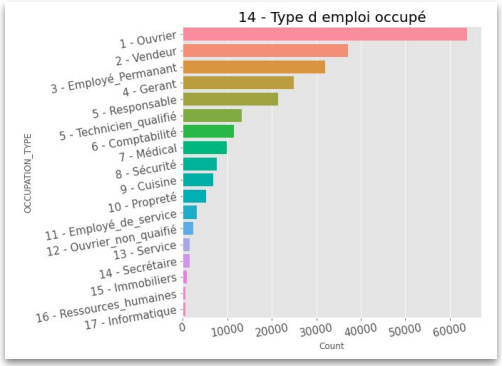


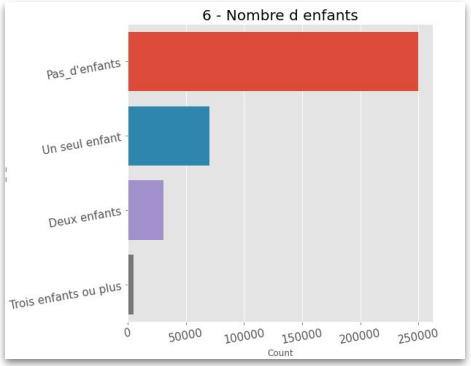
Nous sommes en présence de 7 datasets différents , pour cette étude préalable nous allons nous intéresser à la base de données "Application-Test-Train". Focus sur ce dernier :

- 307533 lignes de client
- 122 colonnes contenant des infos telle que :
  - Id du client
  - Difficulté de paiement (1 ou 0)
  - Somme empruntée
  - Âge ...

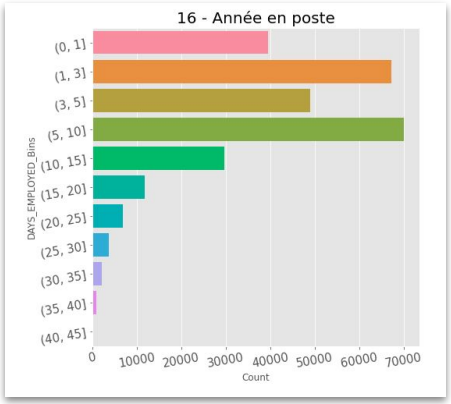


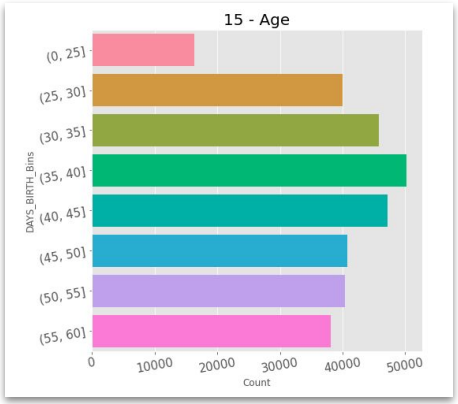


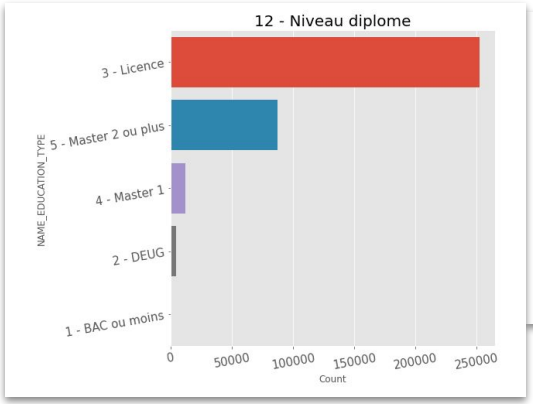


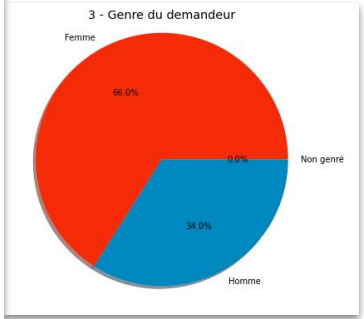


2.2 - Profil des clients

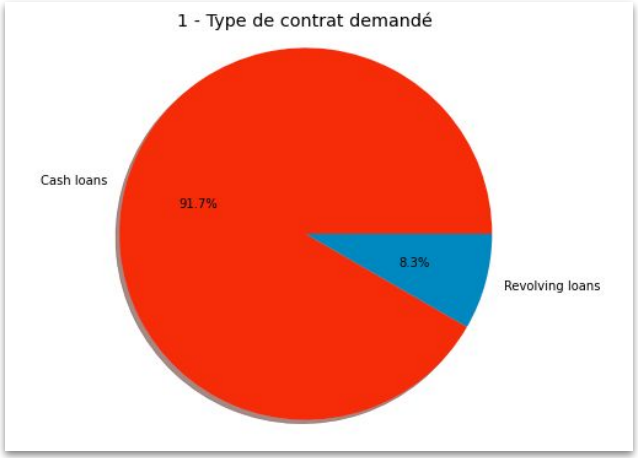


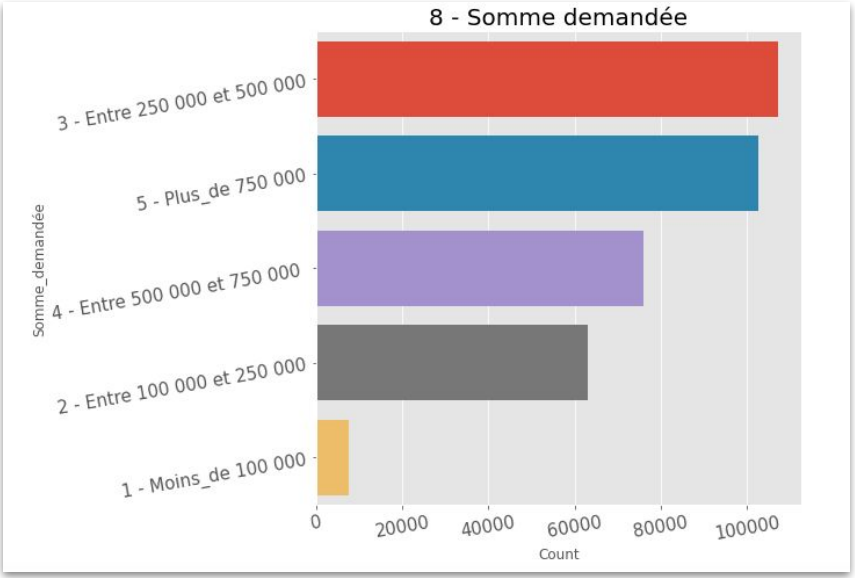


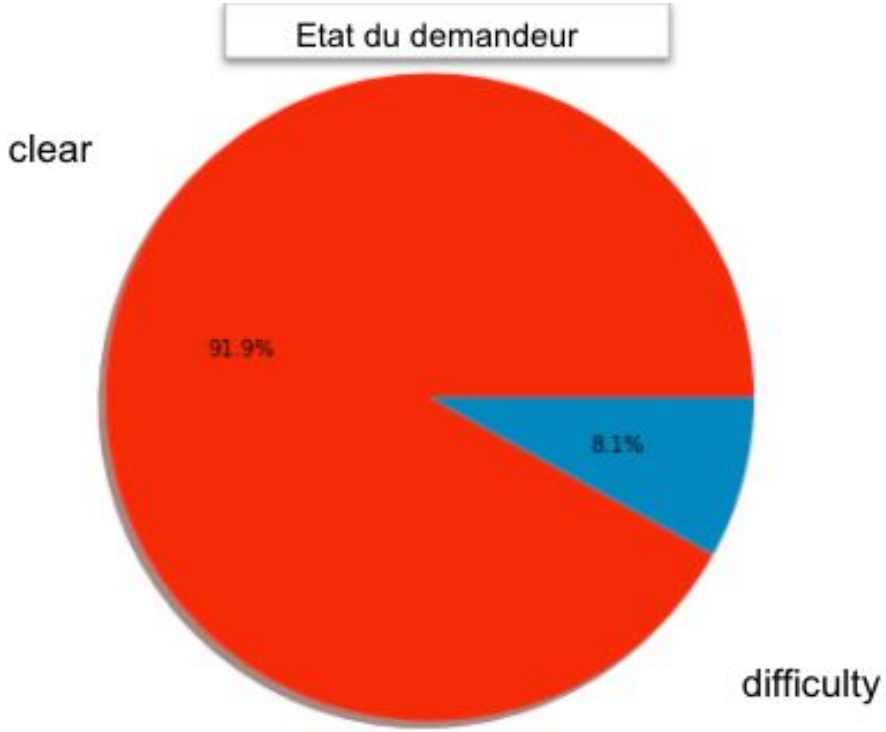




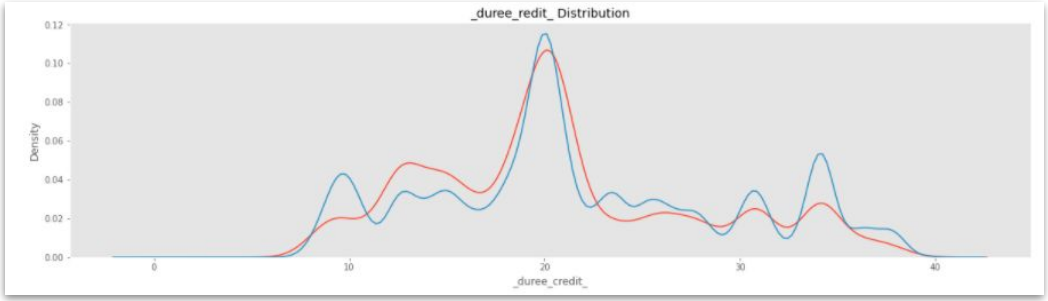
2.2 - Profil des demandes

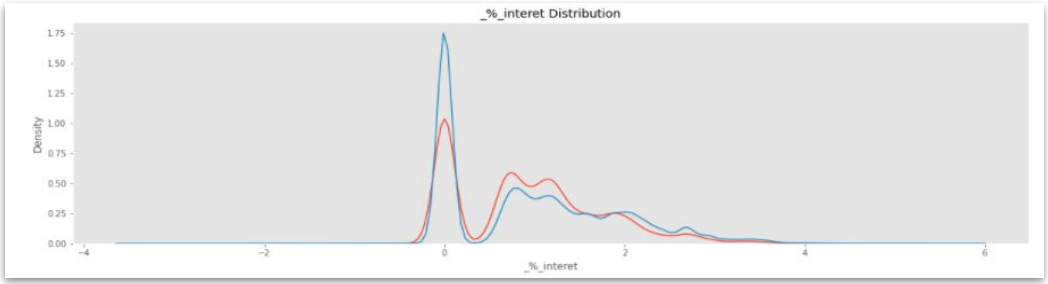


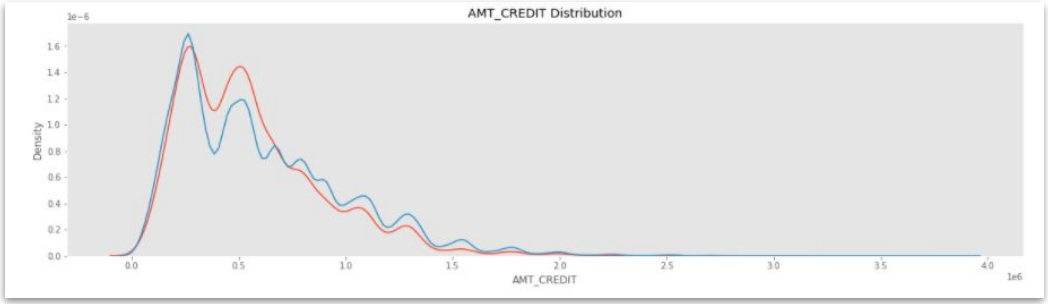


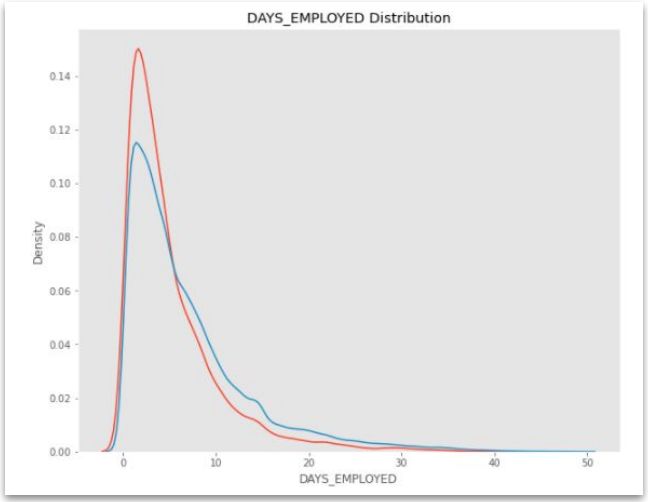


2.4 - Analyse bi-varié en fonction de la target

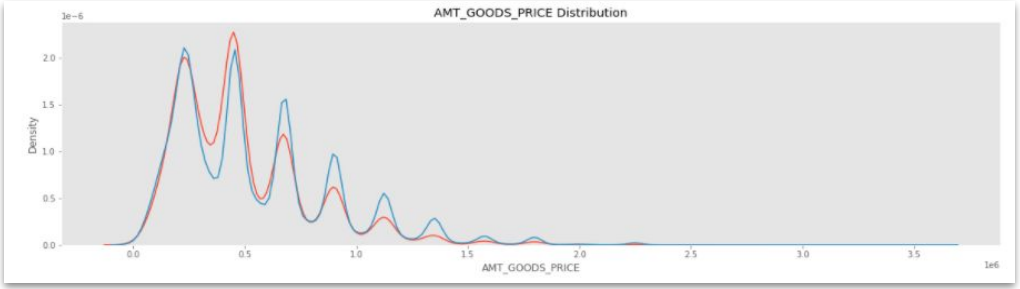


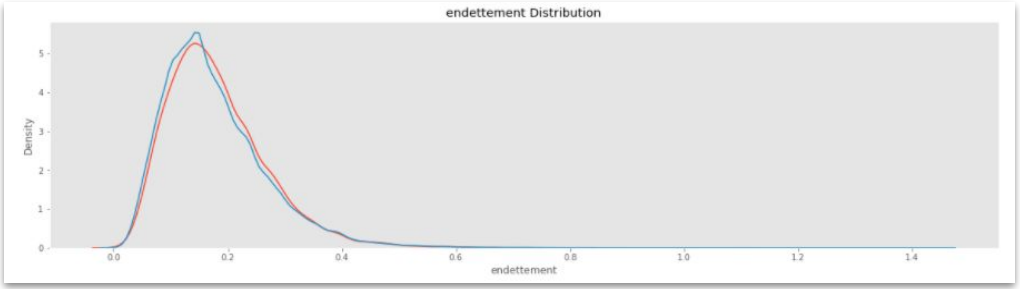


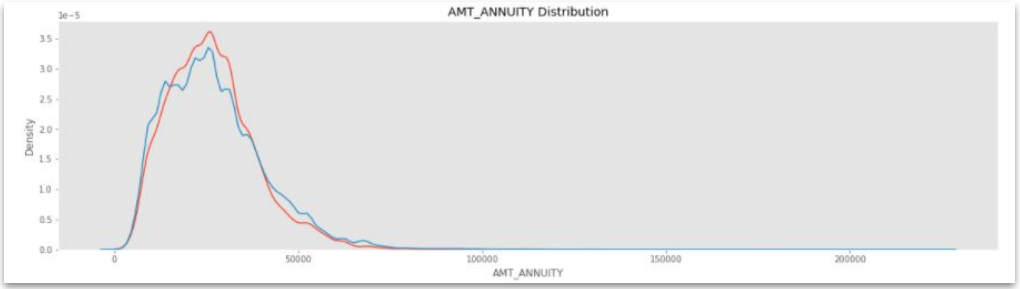


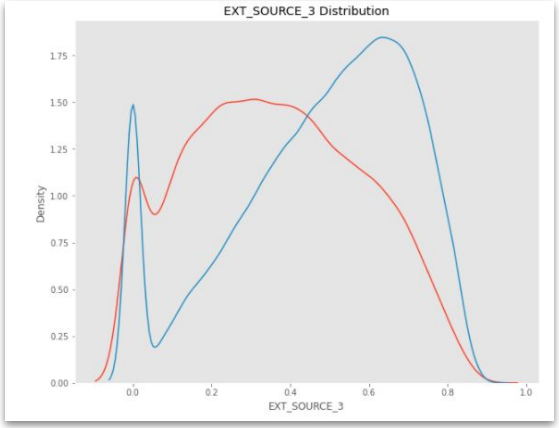


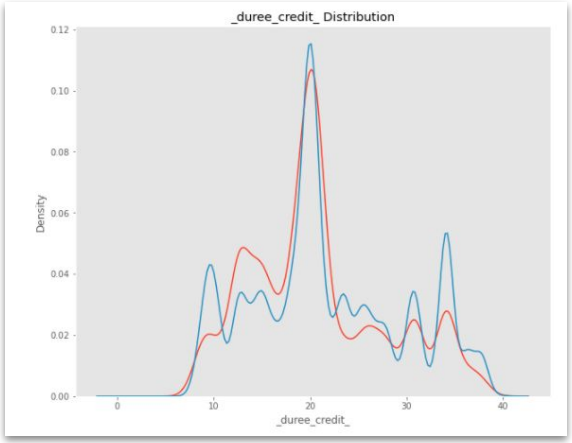
2.4bis - Analyse bi-varié en fonction de la target











3 - Pré-Process

3.1 Préalable

3.1.1 Cleaning

- Tri par ratio de colonnes vides
- Tri par ratio de lignes vides
- Suppression des outliers
- Nettoyage et simplification
    - Niveau de diplôme
    - Métier
    - Statut professionnel

3.1.2 Features engineering

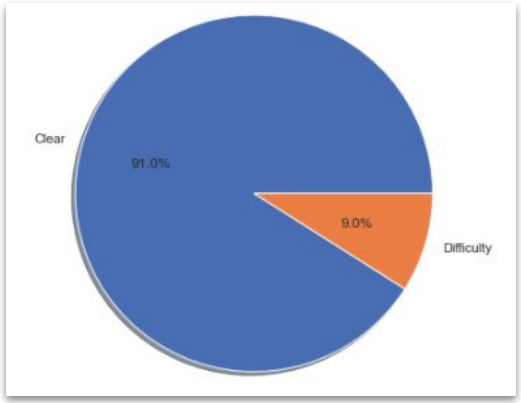
Création de nouvelles variables

- Durée du crédit
- Taux d'intérêt annuel
- Part fiscale
- Taux endettement
- Document reçu (clustering)
- Score du cercle social (clustering)
- Données calendaires

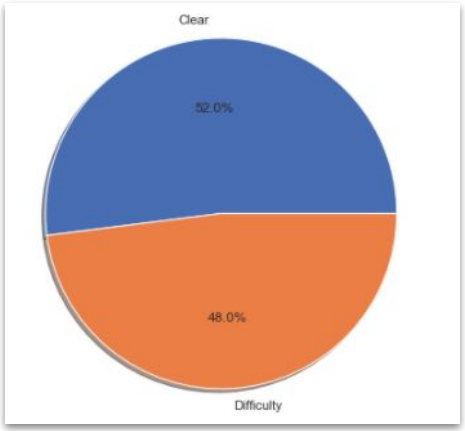
3.1.3 Random Over Sampling

Comme on le constate les données à déterminer sont déséquilibrés. On procède donc à un ajustement pour créer des données équivalente via la méthode Random Over Sampling. Ce dernier va créer un ratio plus facilement interprétable

Avant



Apres



3.2 Variables numériques

- **SimpleImputer**

  SimpleImputer est une classe scikit-learn qui est utile pour gérer les données manquantes dans l'ensemble de données du modèle prédictif. Il remplace les valeurs NaN par un espace réservé spécifié dans ce cas la valeur moyenne

- **Quantile Transform**

  La transformation quantile est une technique de transformation de données non paramétrique pour transformer votre distribution de données numériques en suivant une distribution de données normale.

3.3 Variables catégorielles

- **SimpleImputer**

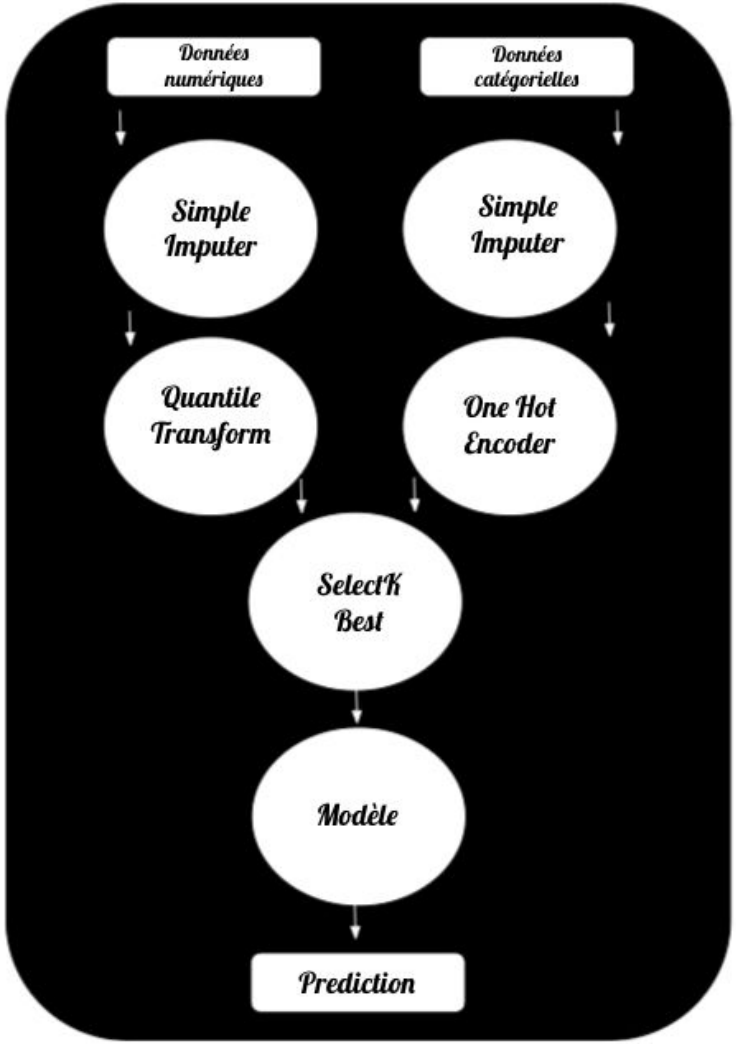
  Dans ce cas on remplace les valeurs NaN par l'item la plus présente dans la colonne

- **OneHotEncoder**

  Un encodage à chaud est un processus de traitement des données appliqué aux données catégorielles, pour les convertir en une représentation vectorielle binaire à utiliser dans les algorithmes d'apprentissage automatique.

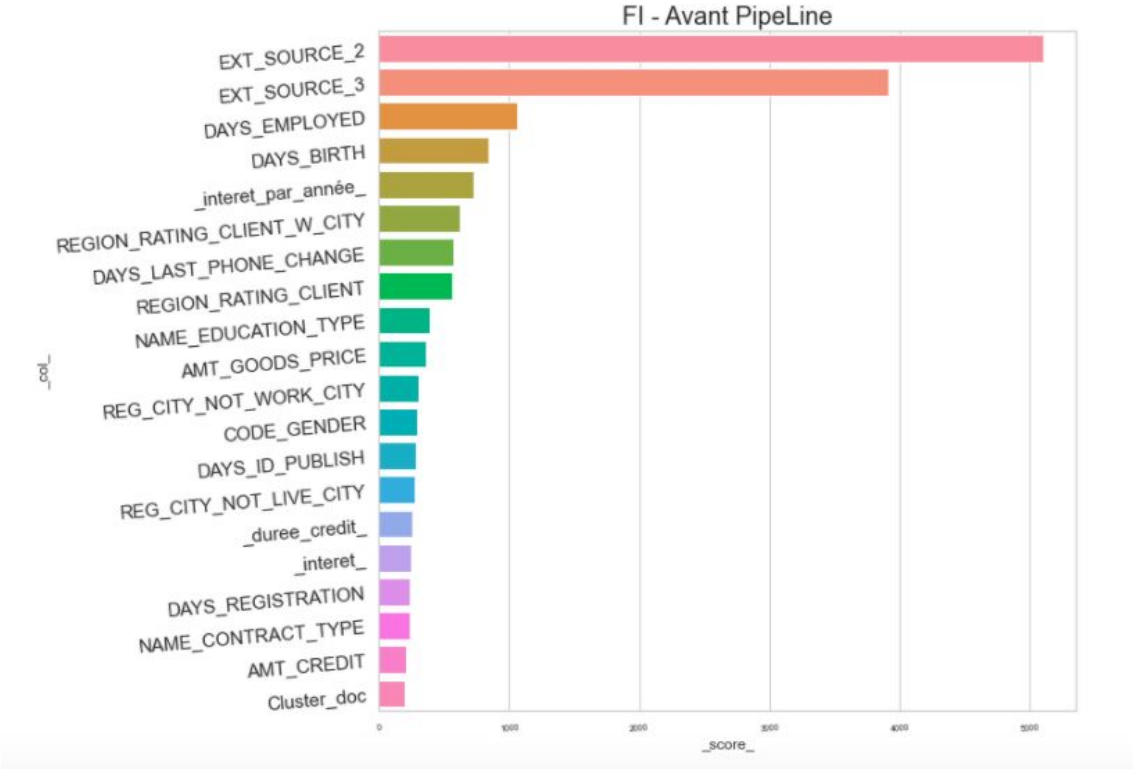
3.4 Pipeline

- ➔ Preprocess
- ➔ SelectKBest
- ➔ Model
  - Logistic
  - RandomForest
  - XGBoost

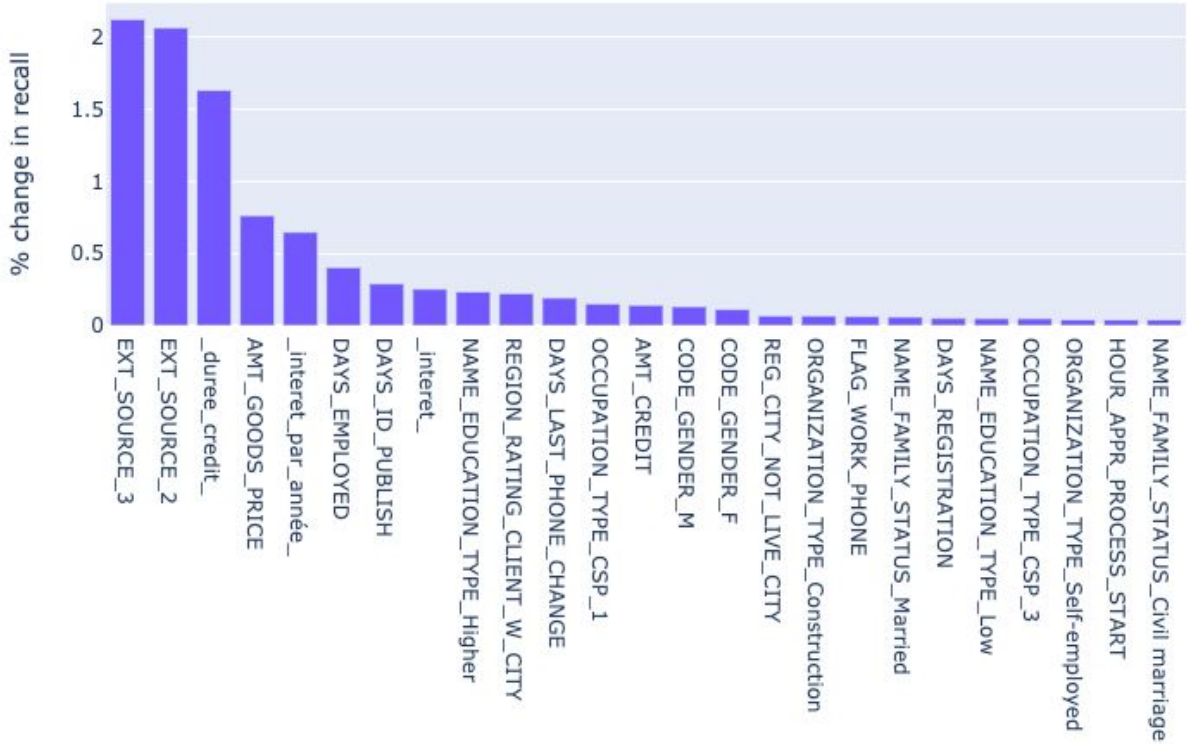


4 - Features importances

4.1 - Features importances avant Pipeline



4.2 - Features importances après PipeLine



5 - Scoring

5.1 - Metrics disponibles

- Accuracy

$$Accuracy = \frac{vrai\ positif + vrai\ négatif}{vrai\ positif + vrai\ négatif + faux\ positif + faux\ négatif}$$

- Confusion Matrix

- Recall

$$Recall = \frac{vrai\ positif}{vrai\ positif + faux\ négatif}$$

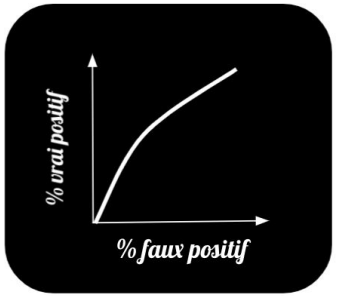
- Précision

$$Précision = \frac{vrai\ positif}{vrai\ positif + faux\ positif}$$

- F1

$$F1 = \frac{2\ x\ precision\ x\ recall}{precision + recall}$$

- Roc-Auc



5.2 - Création d'un score "sur-mesure"

$$\text{Score crée} = \frac{\text{vrai positif} + \text{faux positif}*25 + \text{vrai négatif} + \text{faux négatif}*100}{\text{vrai positif} + \text{faux positif} + \text{vrai négatif} + \text{faux négatif}}$$

6 - Prédiction

6.1 - Neutre

- Logistic :

  Roc = 0.6721
  score créé = 317

- RandomForest :

  Roc = 0.7284
  score crée = 105

- XGBoost :

  Roc = 0.7681
  score créé = 50

6.2 - Avec Hyperparamètres

(méthode Grid & Random)

- Logistic :

  Roc = 0.7335
  score créé = 305

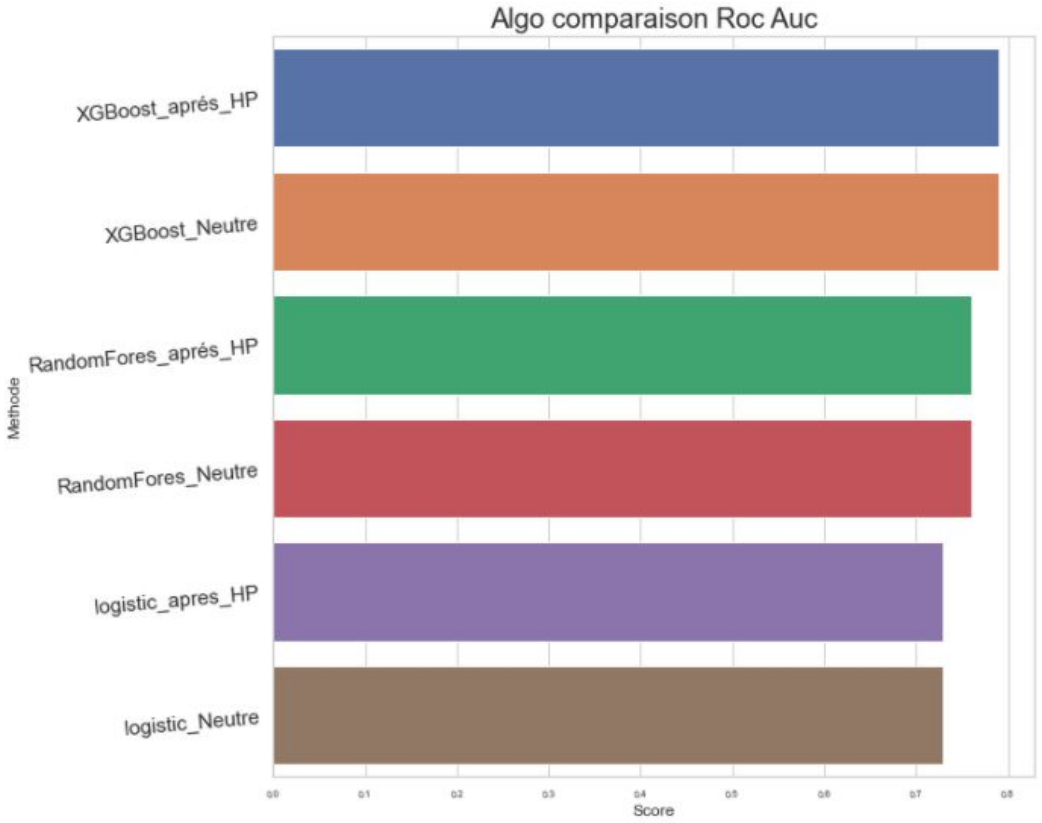
- RandomForest :

  Roc = 0.7656
  score crée = 52

- XGBoost :

  Roc = 0.7901
  score crée = 45

6.3- Récapitulatif



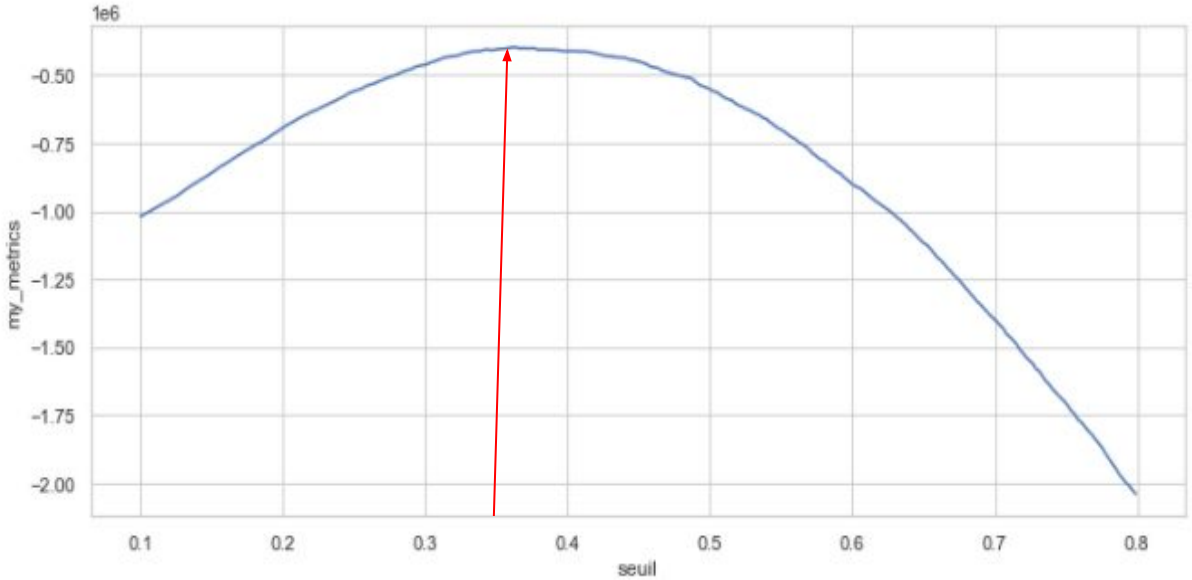
Meilleur Choix:

XGBOOST

7 - Optimisation / Fonction coût

7.1 - Recherche seuil optimal

	seuil	base_0	all_0	true_0	false_0	base_1	all_1	true_1	false_1	my_metrics
699	0.799	36389	64572	35412	29160	36077	7894	6917	977	-2056024
698	0.798	36389	64493	35404	29089	36077	7973	6988	985	-2049317
697	0.797	36389	64377	35384	28993	36077	8089	7084	1005	-2040781
696	0.796	36389	64301	35375	28926	36077	8165	7151	1014	-2034536
695	0.795	36389	64187	35359	28828	36077	8279	7249	1030	-2025566
...	...	...	...	...	...	...	...	...	...	...
289	0.389	36389	25657	21068	4589	36077	46809	31488	15321	-406305
284	0.384	36389	25203	20780	4423	36077	47263	31654	15609	-406243
292	0.392	36389	25939	21249	4690	36077	46527	31387	15140	-406008
291	0.391	36389	25834	21190	4644	36077	46632	31433	15199	-404784
290	0.390	36389	25741	21132	4609	36077	46725	31468	15257	-404613



Seuil optimal:
0.362

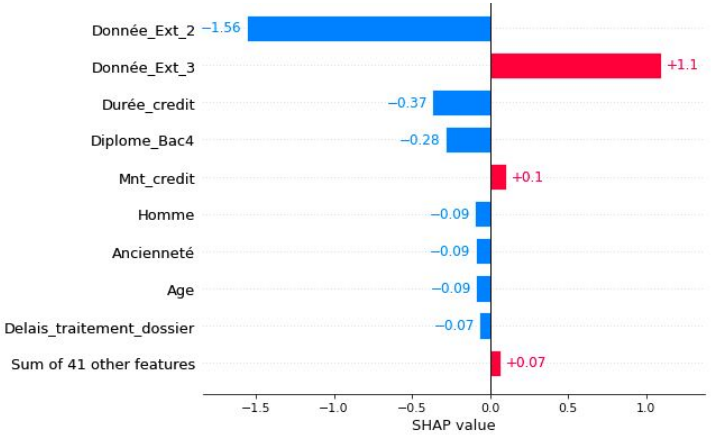
8 - Interprétation

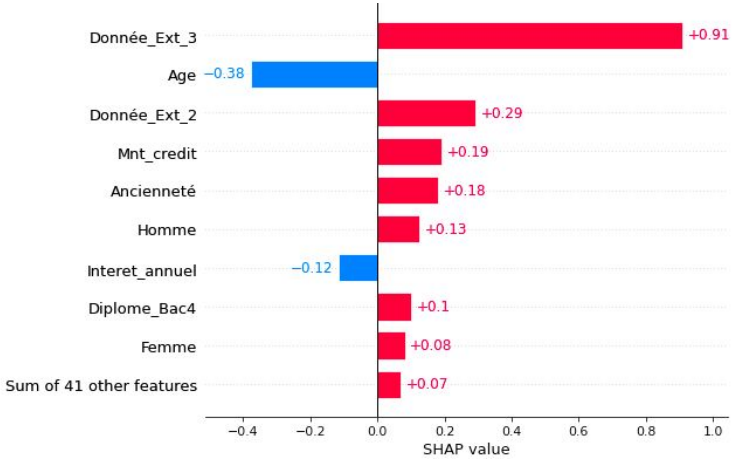
8.1 - SHAP

Les méthode SHAP calcule la moyenne de l'impact d'une variable sur la prédiction pour toutes les combinaisons de variables possibles.

Un graphique clair et précis, tout à fait intéressant à importer dans le DashBoard

Exemple de prédiction négative

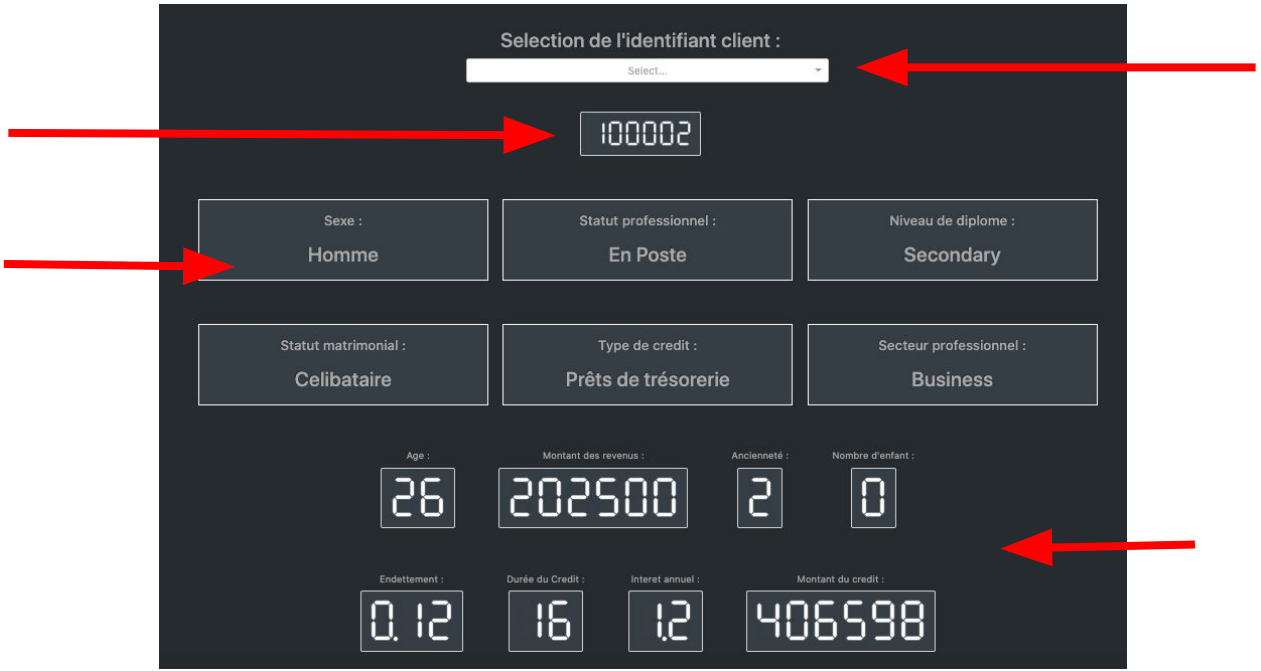




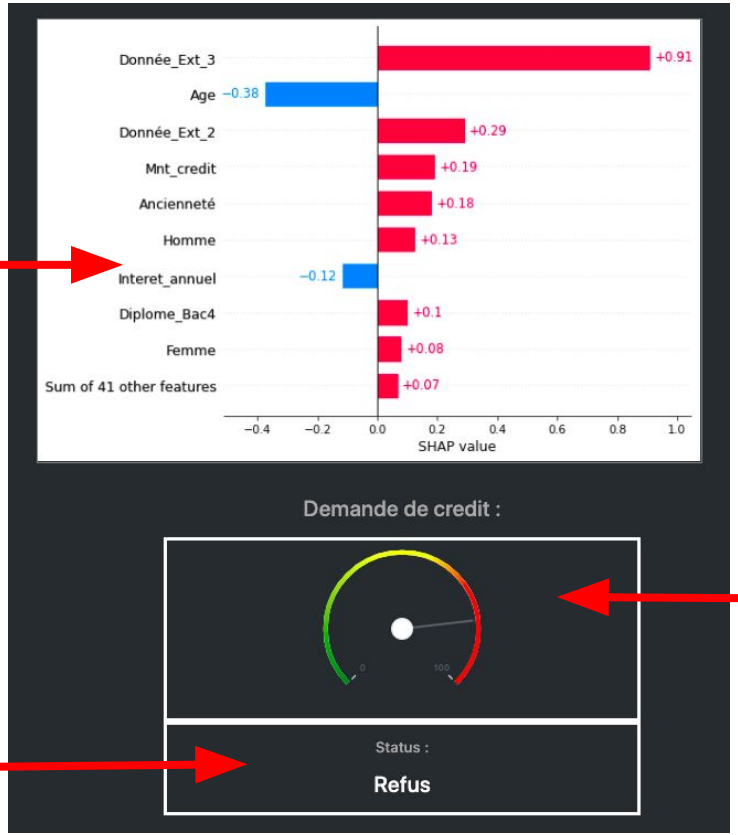
Exemple de prédiction positive

9 - DashBoard

9.1 - DashBoard information client



9.2 - DashBoard prédiction et décision



10 - Conclusion

Ce travail nous a permis de déterminer avec une bonne probabilité les difficultés de paiement d'un client, et de le présenter de manière compréhensible sur un dashboard.

Néanmoins quelques améliorations peuvent être apporté :

- La gestion du seuil a été décidé en fonction des valeurs trouvées néanmoins, il doit être ajusté en fonction de la temporalité et des décisions de la banque en fonction du marché
- Il est possible d'étudier l'intégralité de la base SQL pour parfaire les prédictions
- Améliorer le Feature engineering avec l'appui d'un consultant métier
- Travailler en accord avec les conseillers pour améliorer le dashboard

- Il faut effectuer une mise à jour afin de préserver une bonne prédiction dans le temps