IMPLÉMENTEZ UN MODÉLE DE SCORING

Présenté par Souleymane Camara

Contexte de l'étude

L'ENTREPRISE "PRÊT À DÉPENSER"

- 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
- Pour réaliser cette étude on s'appuie sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).
- On doit fournir un outil de compréhension et d'explication du résultat car les clients sont de plus en plus demandeurs de transparence vis-à-vis des décisions d'octroi de crédit .
- Pour cela nous allons mettre en place 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.

Présentation des données

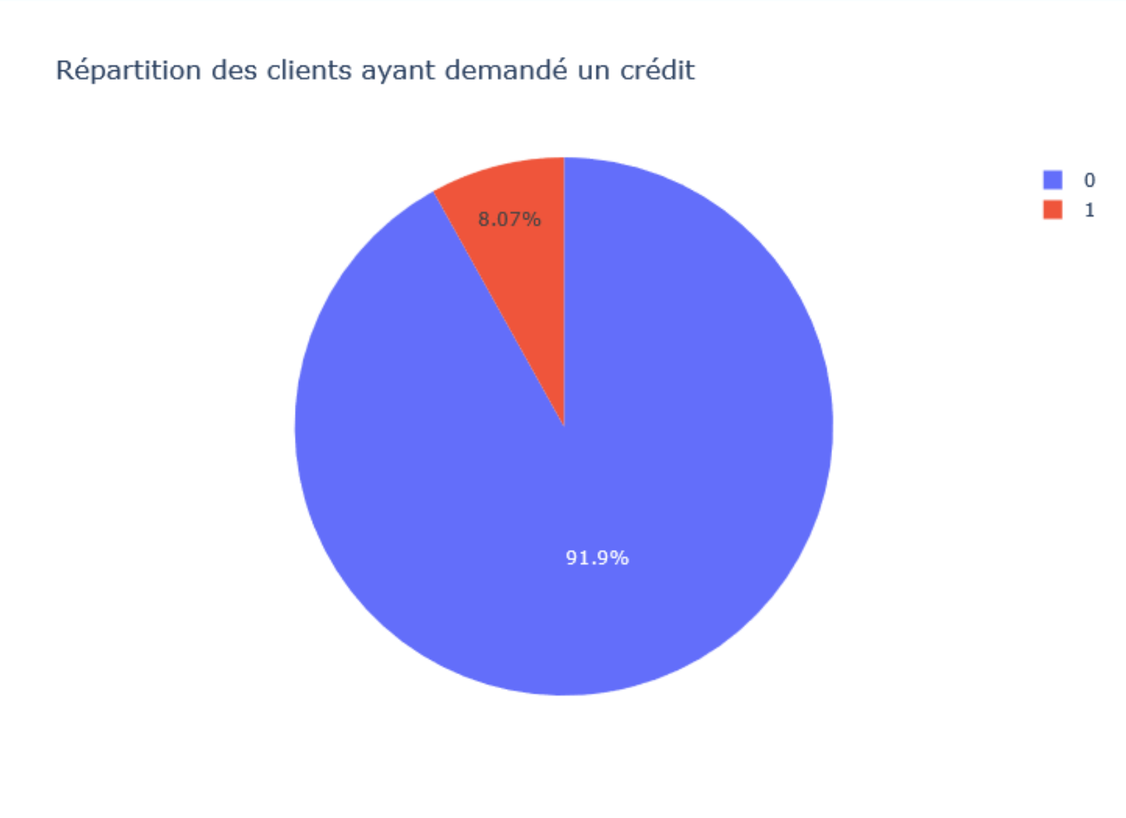
NOUS AVONS 10 TABLE DE DONNÉES:

- Nous allons nous intéresser à trois tables : la table de description des features, la table TRAIN et la table TEST.
- La table **TRAIN** contient 307511 clients avec 122 informations les concernant.
- Parmi les informations on peut en citer certains :
    - La target (0: crédit accepté et 1 crédit rejeté)
    - le sexe du client, situation maritale, nombre d'enfants
    - le montant de ces revenus, le montant de la demande, la mensualité souhaitée, etc.....
- la table **TEST** contient 48744 clients avec 121 features, la colonne Target n'est pas disponible.

Etude des deux tables

RÉPARTITION DES CLIENTS

Nous avons une table avec 91.9 % de clients qu'on accepte leur crédit et 8.07 % qu'on rejette.



VALEURS MANQUANTES

la table **TRAIN** a 67 features (122 features) qui contiennent des valeurs manquantes allant de 0.1 % à 69 %.

la table TEST a 64 features (121 features) qui contiennent des valeurs manquantes allant de 0.1 % à 69.9%.

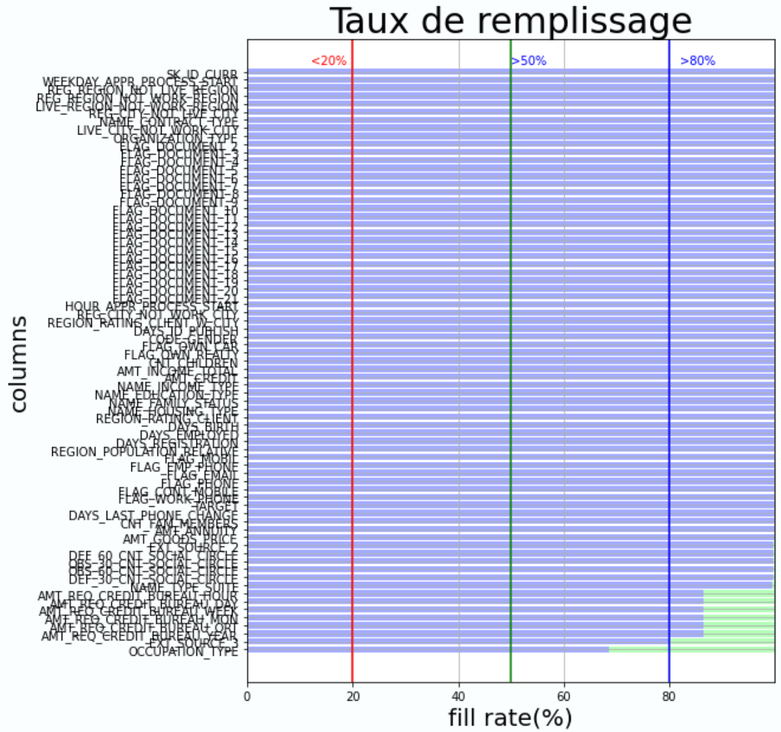
Etude des deux tables

VALEUR MANQUANTE

Nous allons supprimer toutes les features ayant plus de 45% de valeurs manquantes.
On se retrouve avec 73 features pour la table TRAIN et 72 features pour la table TEST.

DIMINUTION DES MODALITÉS

Nous avons 12 variables catégorielles parmi elles nous en avons avec beaucoup de modalités.




```
# number of unique classes in each object column
TRAIN.select_dtypes('object').apply(pd.Series.nunique, axis = 0)
```

```
NAME_CONTRACT_TYPE             2
CODE_GENDER                    3
FLAG_OWN_CAR                   2
FLAG_OWN_REALTY                2
NAME_TYPE_SUITE                7
NAME_INCOME_TYPE               8
NAME_EDUCATION_TYPE            5
NAME_FAMILY_STATUS             6
NAME_HOUSING_TYPE              6
OCCUPATION_TYPE               18
WEEKDAY_APPR_PROCESS_START     7
ORGANIZATION_TYPE             58
dtype: int64
```

```
TRAIN.select_dtypes('object').apply(pd.Series.nunique, axis = 0)
```

```
NAME_CONTRACT_TYPE             2
CODE_GENDER                    2
FLAG_OWN_CAR                   2
FLAG_OWN_REALTY                2
NAME_TYPE_SUITE                2
NAME_INCOME_TYPE               6
NAME_EDUCATION_TYPE            5
NAME_FAMILY_STATUS             6
NAME_HOUSING_TYPE              6
OCCUPATION_TYPE                3
WEEKDAY_APPR_PROCESS_START     7
ORGANIZATION_TYPE             11
dtype: int64
```

Etude des deux tables

IMPUTATION VARIABLES CATÉGORIELLES

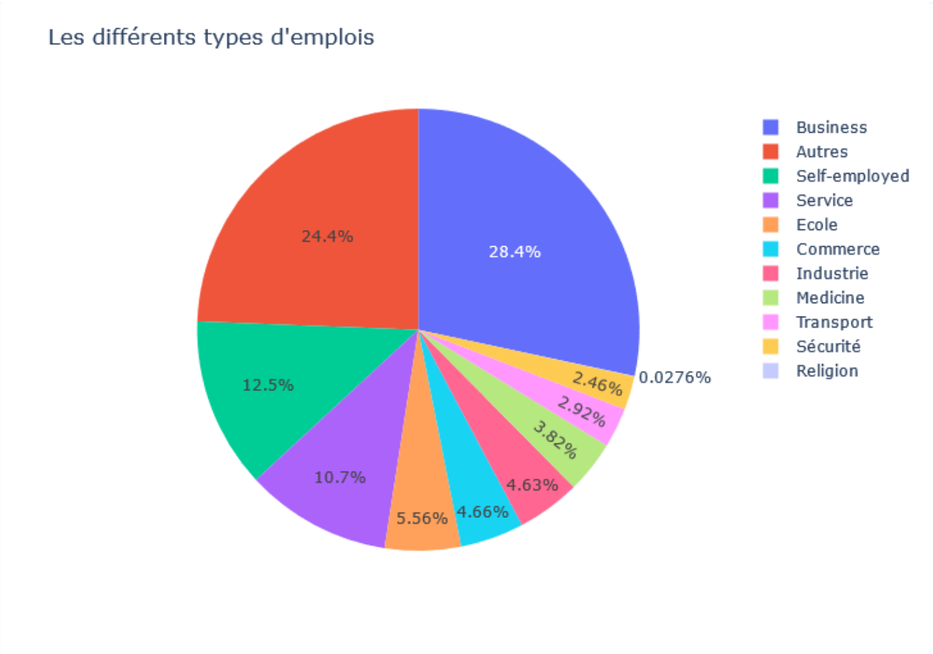
Toutes les valeurs manquantes des variables catégorielles ont été remplacés par la valeur la plus fréquente.

IMPUTATION VARIABLES NUMÉRIQUES

Tout d'abord nous avons supprimer les lignes dont les features n'ont pas beaucoup de valeurs manquantes. On a 305541 lignes soit 0.64 % de lignes perdus pour le TRAIN et 48683 soit 0.72 % de lignes supprimés pour le TEST.
Six variables donnent la même information on en supprime 5, on se retrouve avec 65 features et 64 features.

```
Missing(TRAIN.select_dtypes('object'))
```

```
NAME_CONTRACT_TYPE             0.000000
CODE_GENDER                    0.000000
FLAG_OWN_CAR                   0.000000
FLAG_OWN_REALTY                0.000000
NAME_TYPE_SUITE                0.420153
NAME_INCOME_TYPE               0.000000
NAME_EDUCATION_TYPE            0.000000
NAME_FAMILY_STATUS             0.000000
NAME_HOUSING_TYPE              0.000000
OCCUPATION_TYPE               31.345303
WEEKDAY_APPR_PROCESS_START     0.000000
ORGANIZATION_TYPE              0.878354
dtype: float64
```



Etude de la table TRAIN et TEST

IMPUTATION DES VARIABLES NUMÉRIQUES

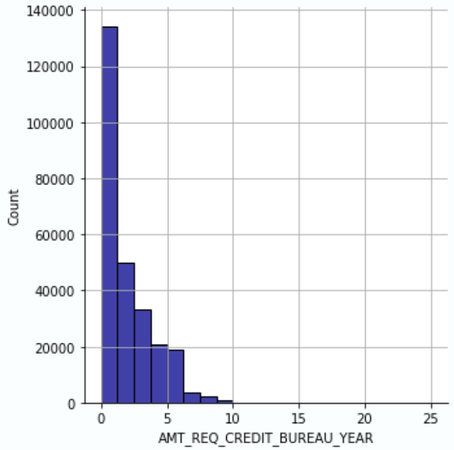
- On se retrouve avec deux variables numériques qui contiennent des NaN.
- Nous allons regarder leur distribution et le taux de remplissage 13 % et 12 % de NAN en vue de choisir la méthode d'imputation : la médiane

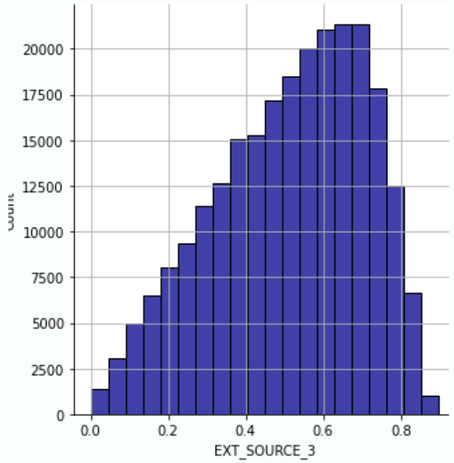
```
Missing(TEST.select_dtypes(['int64','float64'])).sort_values(ascending=False)
#Nous voyons que ces 6 variables sont à peu prés la meme chose
```

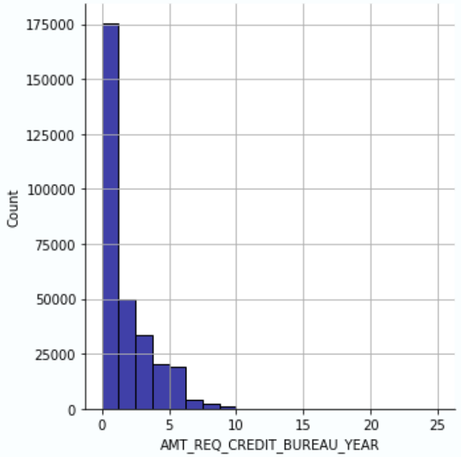
```
EXT_SOURCE_3                  17.768009
AMT_REQ_CREDIT_BUREAU_YEAR    12.410903
LIVE_CITY_NOT_WORK_CITY        0.000000
REG_CITY_NOT_LIVE_CITY         0.000000
LIVE_REGION_NOT_WORK_REGION    0.000000
REG_REGION_NOT_WORK_REGION     0.000000
REG_REGION_NOT_LIVE_REGION     0.000000
HOUR_APPR_PROCESS_START        0.000000
REGION_RATING_CLIENT_W_CITY    0.000000
REGION_RATING_CLIENT           0.000000
CNT_FAM_MEMBERS                0.000000
FLAG_EMAIL                     0.000000
FLAG_PHONE                     0.000000
FLAG_CONT_MOBILE               0.000000
FLAG_WORK_PHONE                0.000000
FLAG_EMP_PHONE                 0.000000
FLAG_MOBIL                     0.000000
DAYS_ID_PUBLISH                0.000000
DAYS_REGISTRATION              0.000000
DAYS_EMPLOYED                  0.000000
DAYS_BIRTH                     0.000000
REGION_POPULATION_RELATIVE     0.000000
AMT_GOODS_PRICE                0.000000
AMT_ANNUITY                    0.000000
```

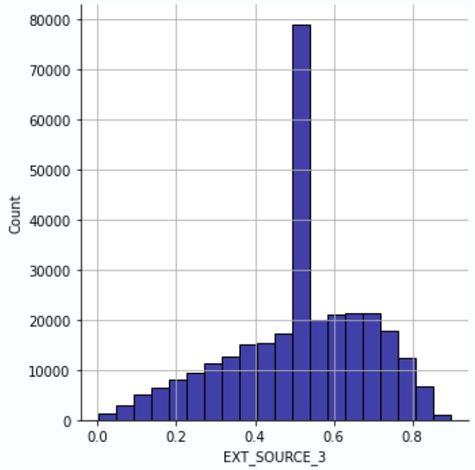
✓ 0 s terminée à 20:25

1:30 19°C Ensoleillé FRA FR



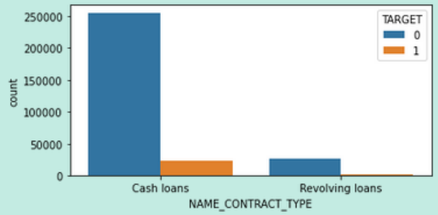


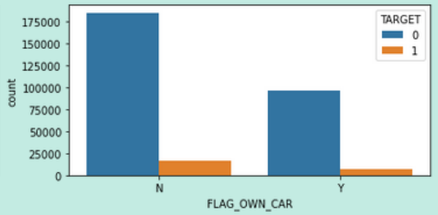


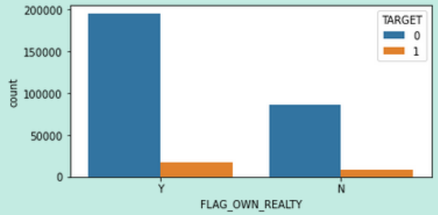


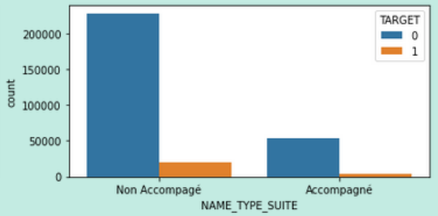
RAPPORT ENTRE NOS VARIABLES CATÉGORIELLES ET TARGET :

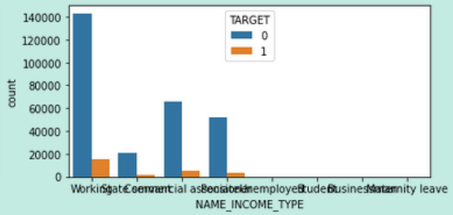
- Nous allons voir l'impact de notre variable target sur nos variables catégorielles

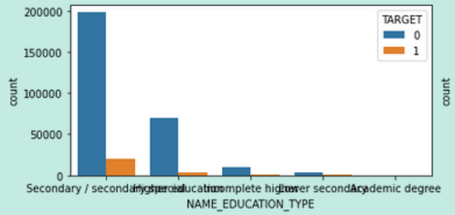


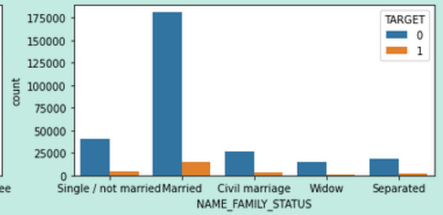


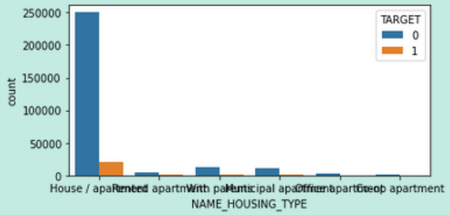


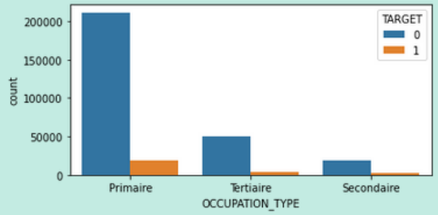


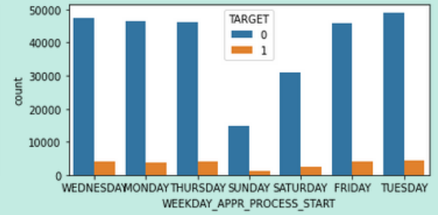


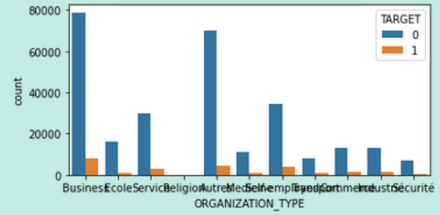






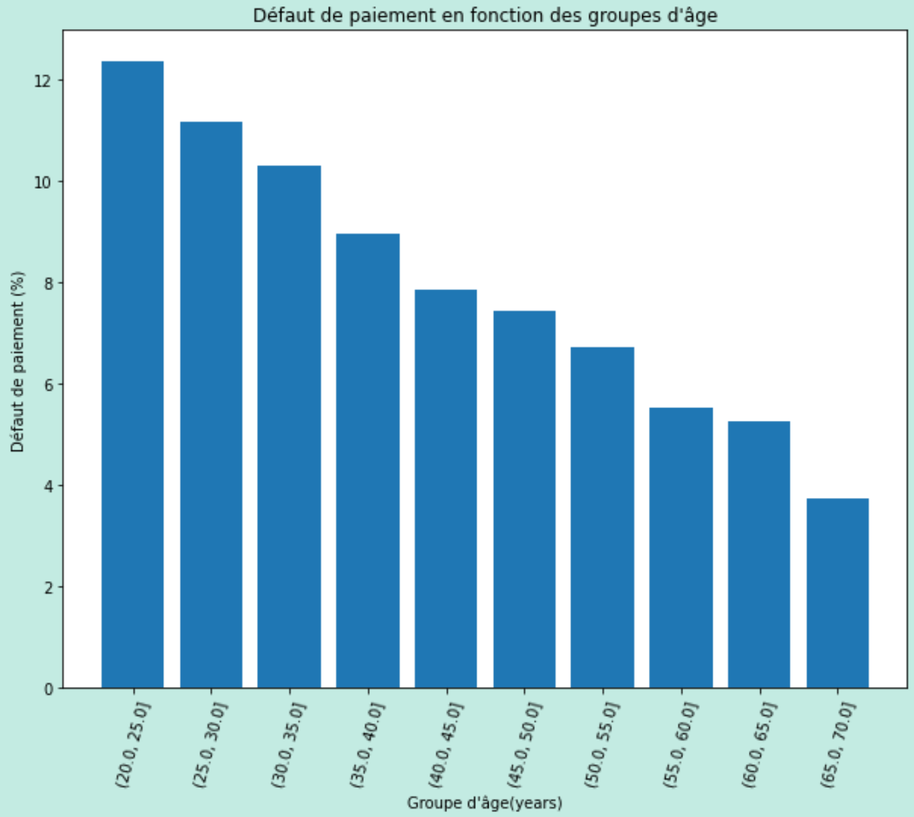






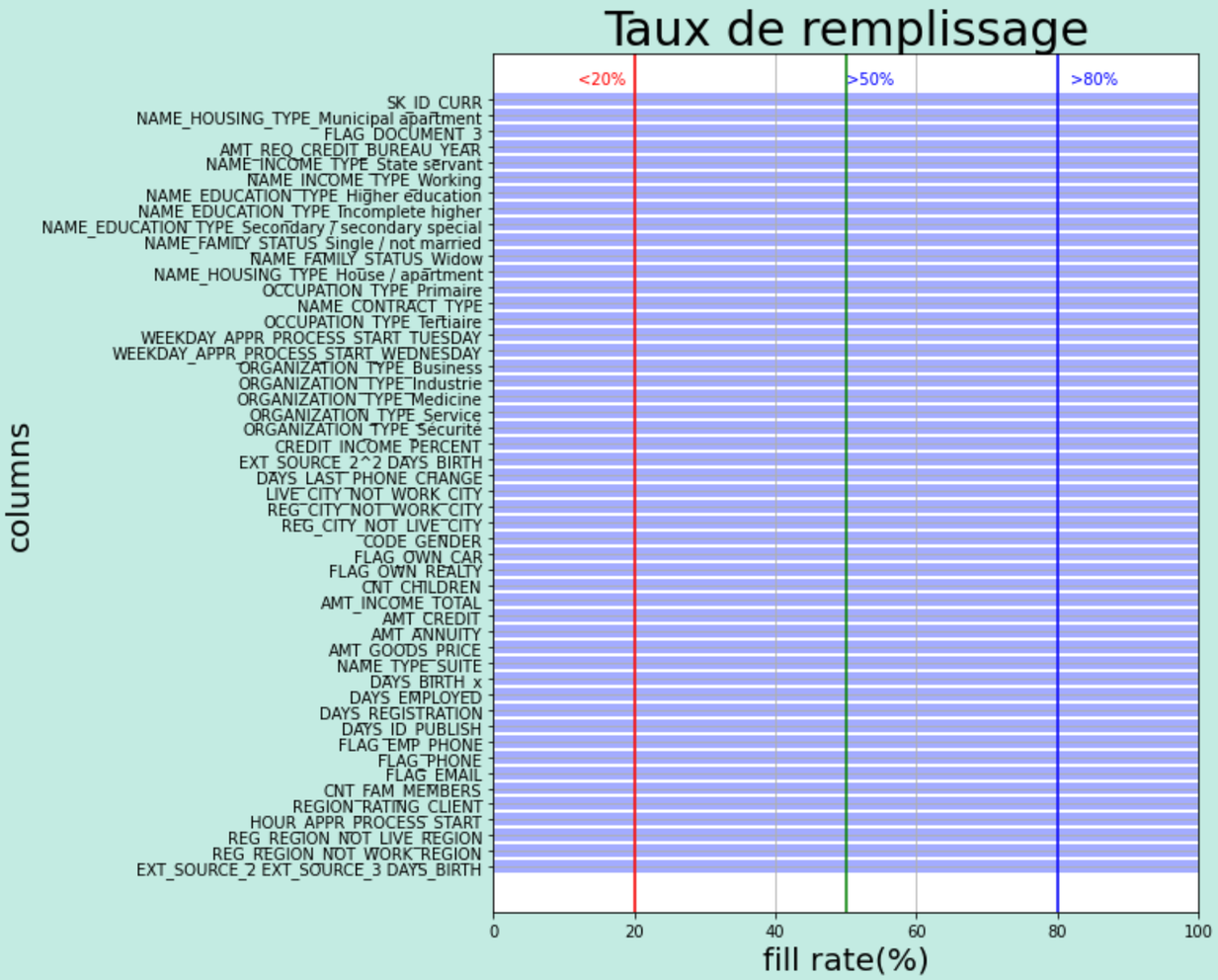
Feature Enginering

- Aprés l'encodage des variables catégorielles on passe de 65 variables à 101 variables pour le TRAIN
- Nous allons étudier la variable qui donne l'âge des clients pour voir s'il y'a une influence sur notre target
- Création d'une fonction polynomiale avec les variables suivantes poly_features = **TRAIN[['EXT_SOURCE_2', 'EXT_SOURCE_3', 'DAYS_BIRTH', 'TARGET']]**
- Création des features suivantes:**CREDIT_INCOME_PERCENT** : le pourcentage du montant du crédit par rapport au revenu du client.
- **ANNUITY_INCOME_PERCENT** : le pourcentage de l'annuité du prêt par rapport au revenu du client.
- **CREDIT_TERM** : la durée du paiement en mois (l'annuité étant le montant mensuel dû).
- **DAYS_EMPLOYED_PERCENT** : le pourcentage des jours d'emploi par rapport à l'âge du client.
- On se se retrouve avec 303584 et 126 features



Feature Enginering

- Nous allons sélectionner nos variables grâce à Scikit Learn en utilisant les méthodes suivantes:
- **Variance Threshold** qui élimine les variables dont la variance est inférieur à la variance la plus faible (0,suppression de deux variables)
- **Select Kbest** qui permet de sélectionner des variables dont le score de dépendance est plus élevé par rapport à la target en utilisant un test statistique (corrélation) on garde 100 features
- **Select From Model** il entraine un estimateur et sélectionne les variables les plus importantes pour cet estimateur. On utilise la descente de gradient stochastique (SGDRegressor). On sélectionne 50 features.

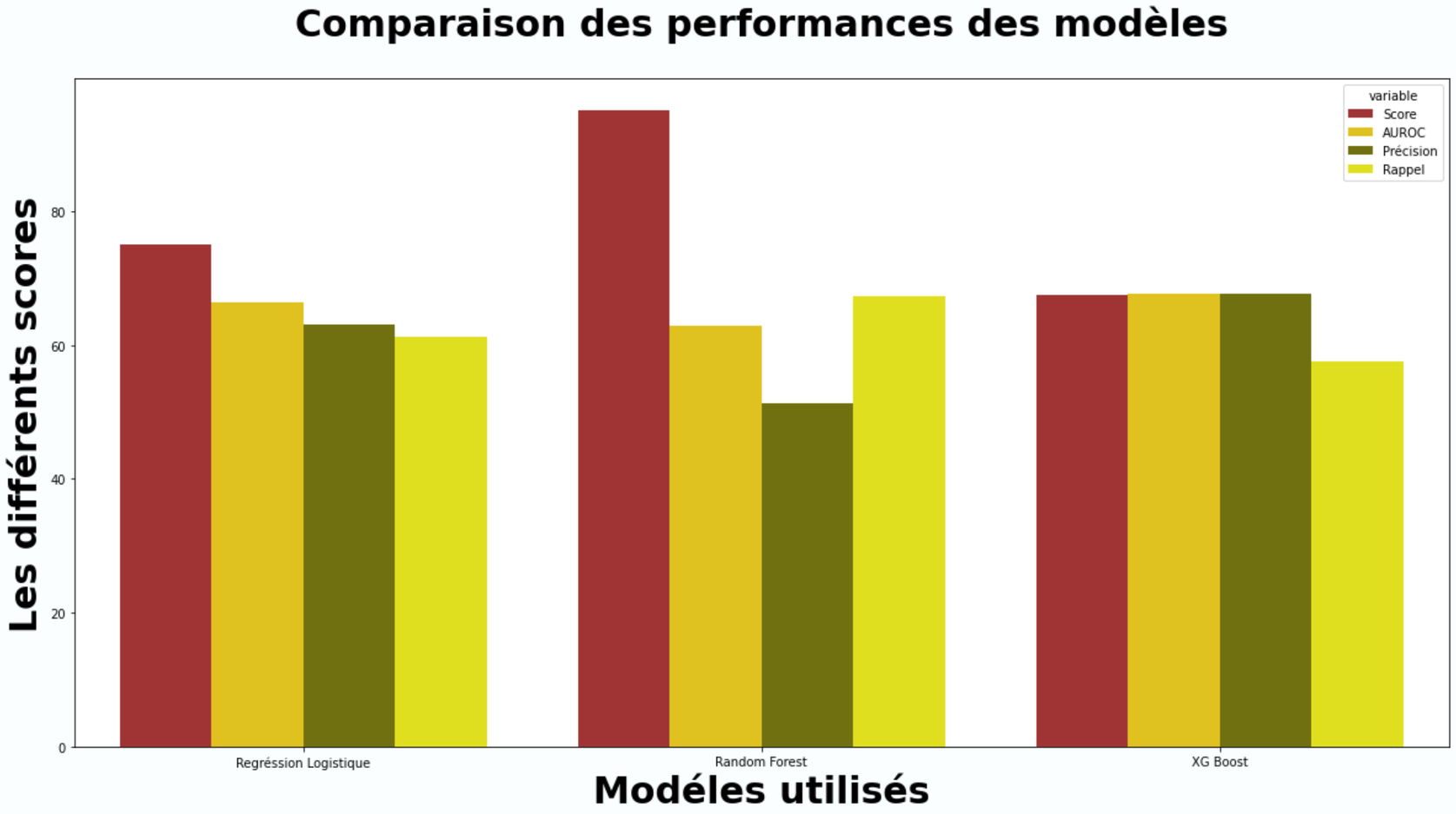


Modélisation et création du score

- Nous allons d'abord diviser notre table en train et test
- Nous voyons que notre target est vraiment déséquilibre pour cela nous allons utiliser des méthodes pour rééquilibrer les classes.
- Il existe deux méthodes principales le sur-échantillonnage : Oversampling et le sous-échantillonnage :Undersampling.
- Les méthodes d'Oversampling fonctionnent en augmentant le nombre d'observations de la classe minoritaire pour atteindre des classes équilibrés.
- Les méthodes d'Undersampling fonctionnent en diminuant le nombre d'observations de la classe majoritaire.
- Nous allons utiliser une méthode Oversampling, smote pour rééquilibrer nos classes
- Nous allons créer un score en se basant sur cette hypothése : **on cherche à minimiser les faux négatifs (erreur de type 2) c'est à dire la classe minoritaire est mal prédite le client ne peut pas rembourser on le prédit quil rembourse.On veut éviter un trop grand nombre de faux positifs le client peut rembourser et on le prédit ne peut pas rembourser .**
- On réalise une combinaison linéaire entre FN et FP en mettant de poids sur le FN et en prenant un réel compris entre 0 et 1.

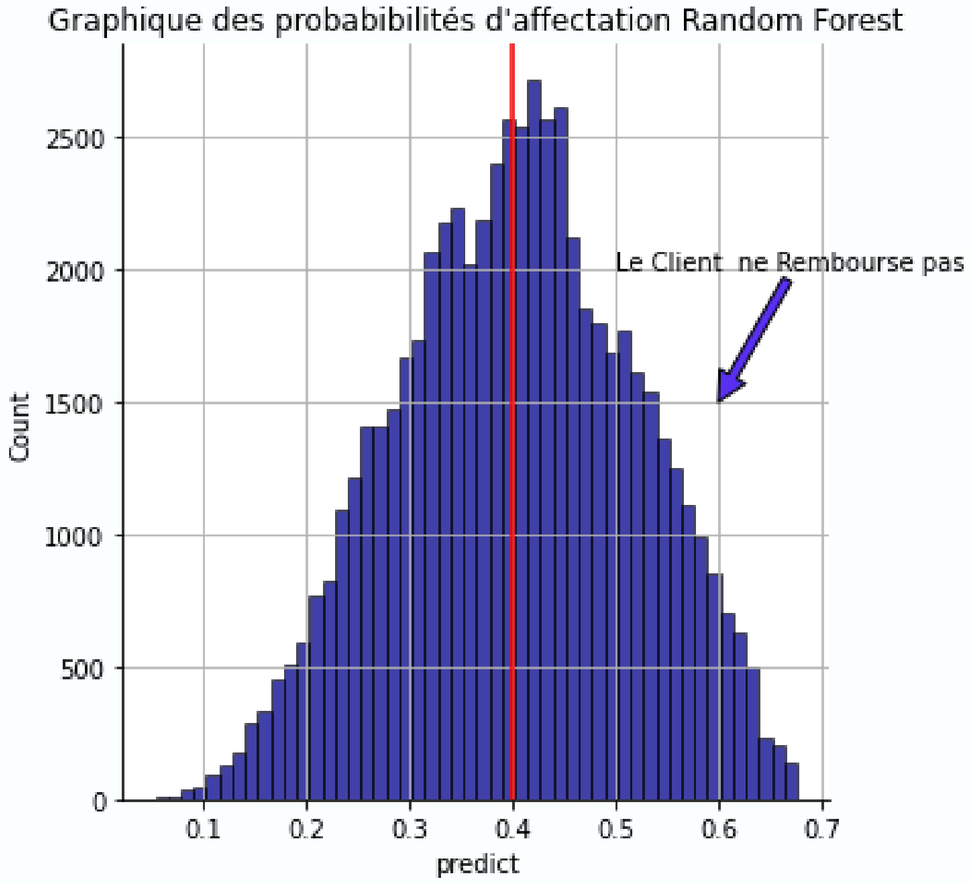
Modélisation

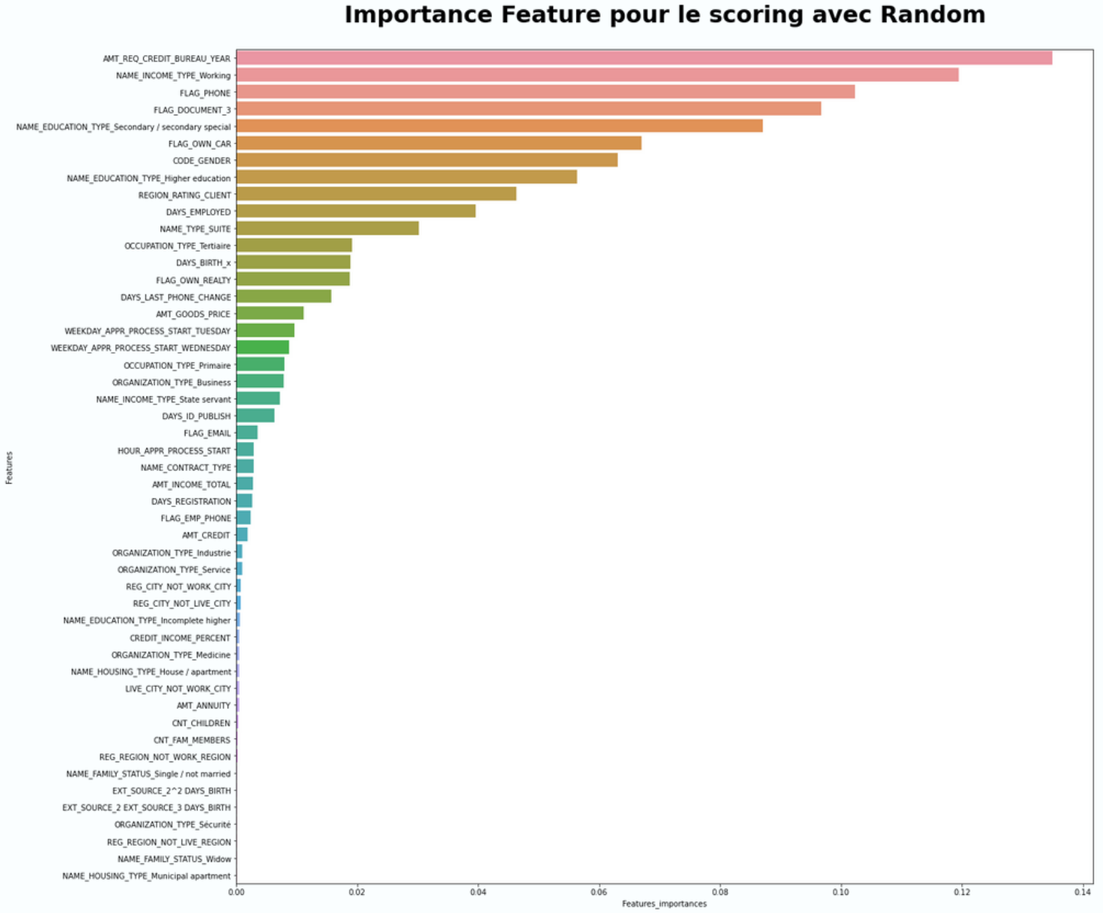
- Utilisons de trois algorithmes la **régression logistique, Random Forest et le Gradient Boosting**
- En utilisant notre scorer nous avons choisis l'algorithme Random Forest.



Modélisation

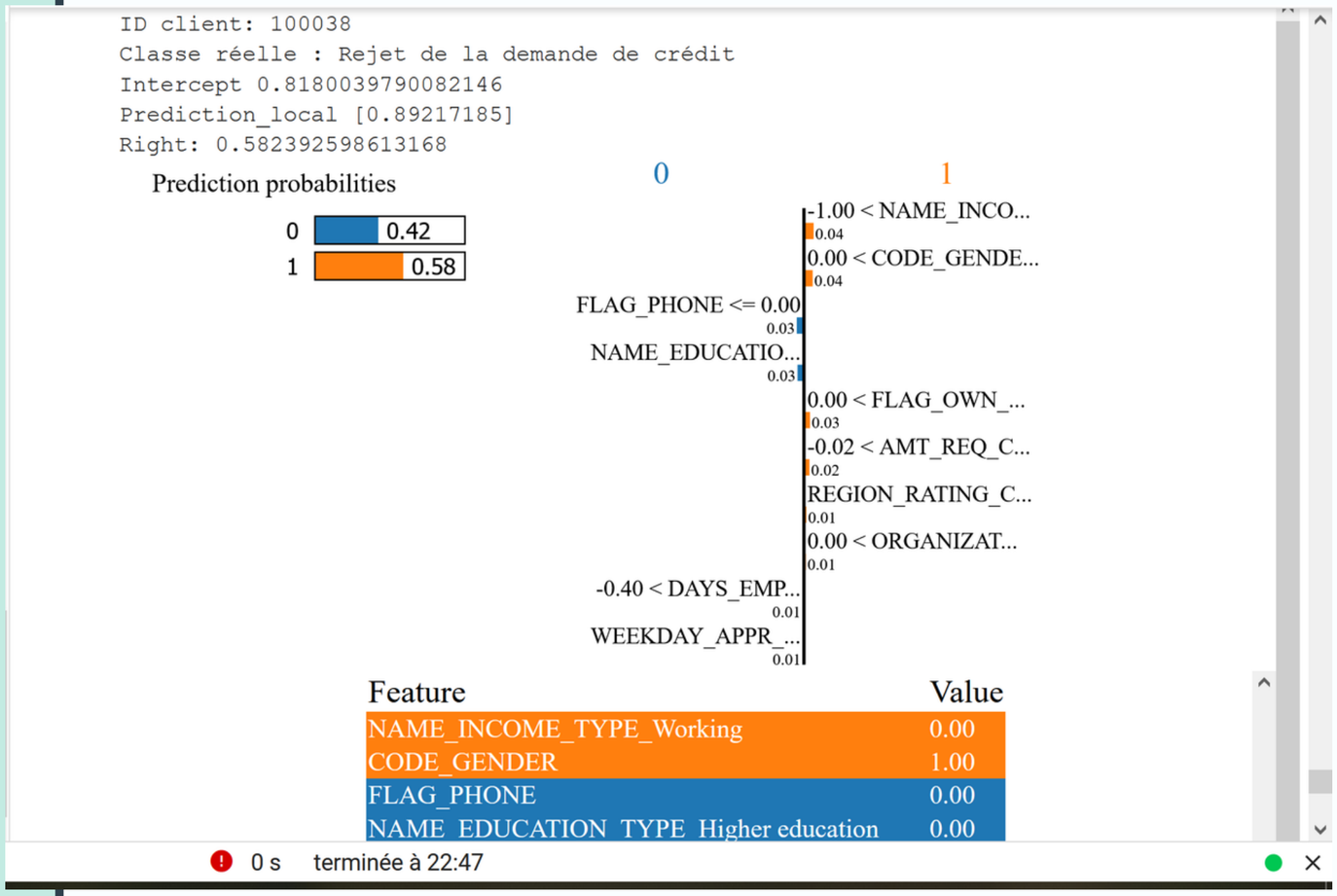
- En choisissant l'algorithme **Random Forest** on doit se fixer un seuil pour le choix de nos cibles.
- Si on dépasse ce seuil on est sur la target numéro 1, en dessous on est sur la target numéro 0.
- On choisit un seuil égal à 0.4





Exemple d'interpretation:

- Nous allons utiliser une méthode d'interprétabilité en vue de comprendre le résultat de notre modéle
- Pour cela nous allons utiliser la méthode LIME, ci joint un exemple de résultat



CONCLUSION

- La modélisation effectuée dans le cadre de ce projet a été effectuée sur la base de la définition d’une mérique d’évaluation : création d'une combinaison linéaire entre le FN (faux négatifs) et FP (faux positifs) en mettant plus de poids sur le FN.
- Ce critère minimise la perte financière, ce choix n'est pas confirmé par les gens du métier
- Ainsi que sur le choix des variables pour la réalisation du modéle, qui est basé sur des critéres statistiques. On peut choisir d'autres features ou créer d'autres variables en discutant avec les personnes du secteur bancaire.
- On peut améliorer notre modéle en utilisant les features des autres tableaux.

FINAL WORDS

MERCI POUR VOTRE ÉCOUTE

SOULEYMANE CAMARA ETUDIANT
DATA SCIENTIST