# P4 OpenClassrooms

Construisez un modèle de scoring

# Sommaire

# Contexte

La société financière "**Pret à dépenser**" propose des crédits à la consommation pour des personnes ayant peu ou pas d'historique de prêt.

pour accorder un crédit à la consommation, l'entreprise souhaite mettre en oeuvre un **outil de "scoring crédit"** qui calcule la probabilité qu'un client le rembourse (0) ou non (1), puis classifie la demande: crédit accordé ou refusé.

0 = negatif = client stable

1 = positif = client à risque

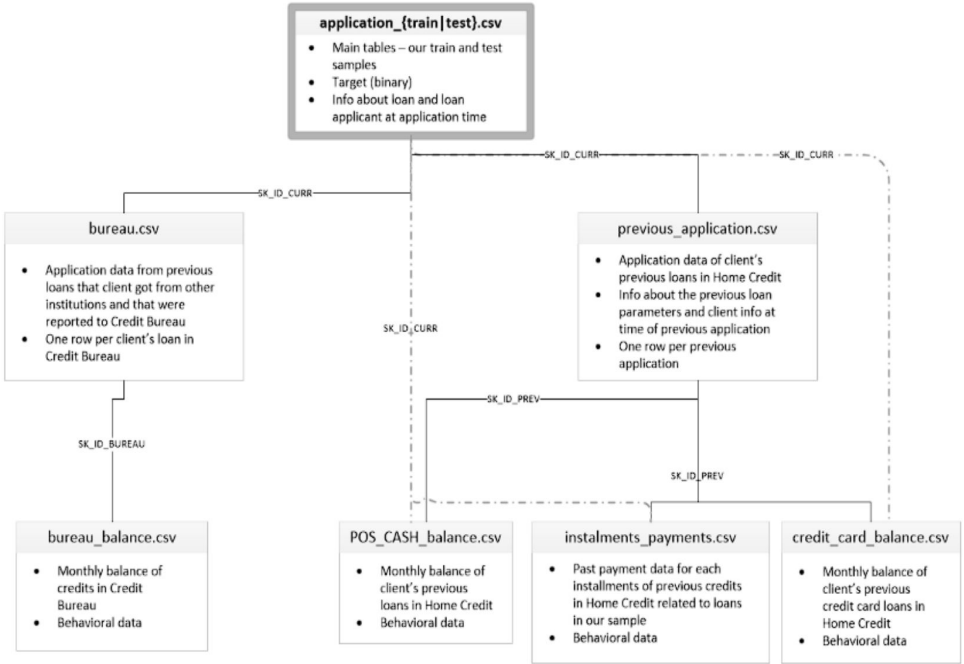
# Jeux de données

- Dans le fichier “application_train” nous avons tous les éléments utiles dont notre variable “Target” qui est la cible à prédire.

- 307511 lignes et 122 features

- Seulement celui-ci est déséquilibré dont seulement 8.78% représente les clients à risques. Il faudra rééquilibrer le jeux de données pour ne pas introduire un biais lors de l'entraînement de notre modèle.

```
TARGET
0    282686
1     24825
Name: count, dtype: int64
```



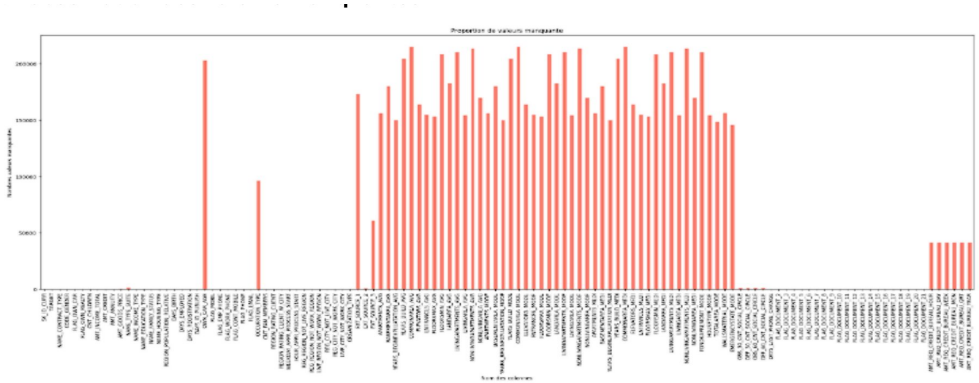
# EDA

VALEURS MANQUANTES

- Il y a 122 colonnes

- 67 ont des valeurs manquantes

- Certaines ont jusqu'à 70% de valeurs manquantes



TYPES DE COLONNES

il y a 16 variables catégorielles, 41 sont des entiers

et 65 sont des nombres flottants

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

Les variables numériques avec moins de 10 modalités

seront classé comme qualitatives

# Corrélation des variables avec TARGET

utilisation de la corrélation de Pearson

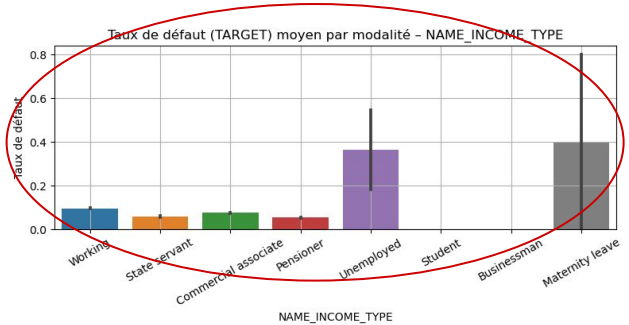
Top 10 des variables positivement et

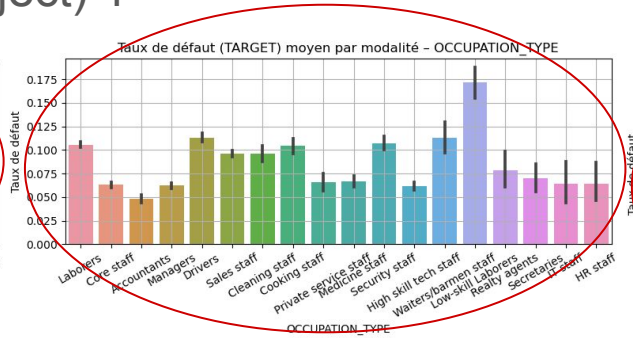
négativement corrélées avec TARGET

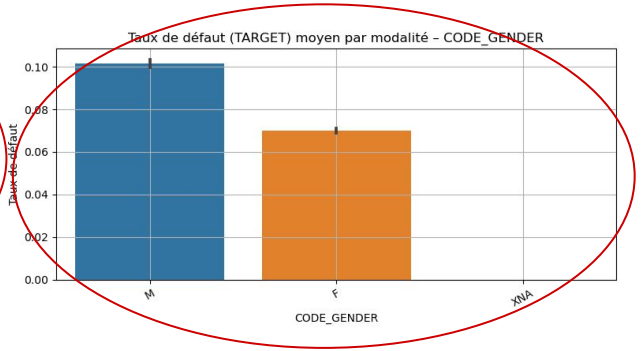
```
Variables les plus positivement corrélées avec TARGET :
TARGET                         1.000000
DAYS_BIRTH                     0.078239
REGION_RATING_CLIENT_W_CITY    0.060893
REGION_RATING_CLIENT           0.058899
DAYS_LAST_PHONE_CHANGE         0.055218
DAYS_ID_PUBLISH                0.051457
REG_CITY_NOT_WORK_CITY         0.050994
FLAG_EMP_PHONE                 0.045982
REG_CITY_NOT_LIVE_CITY         0.044395
FLAG_DOCUMENT_3                0.044346
Name: TARGET, dtype: float64

 Variables les plus négativement corrélées avec TARGET :
ELEVATORS_AVG                 -0.034199
REGION_POPULATION_RELATIVE    -0.037227
AMT_GOODS_PRICE               -0.039645
FLOORSMAX_MODE                -0.043226
FLOORSMAX_MEDI                -0.043768
FLOORSMAX_AVG                 -0.044003
DAYS_EMPLOYED                 -0.044932
EXT_SOURCE_1                  -0.155317
EXT_SOURCE_2                  -0.160472
EXT_SOURCE_3                  -0.178919
Name: TARGET, dtype: float64
```

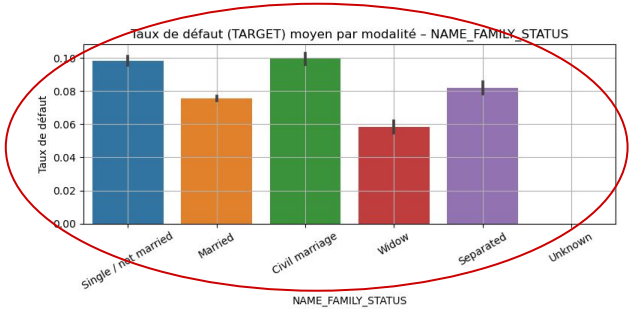
# Corrélation des variables avec TARGET

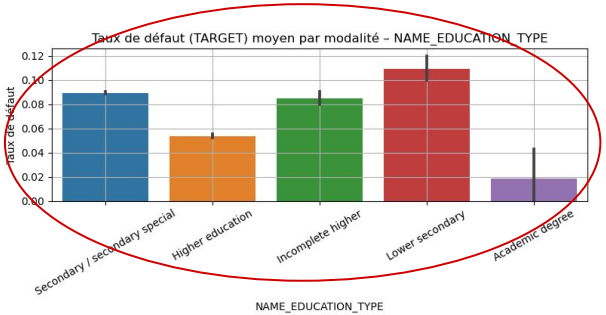
## Variables catégorielles (object) 1

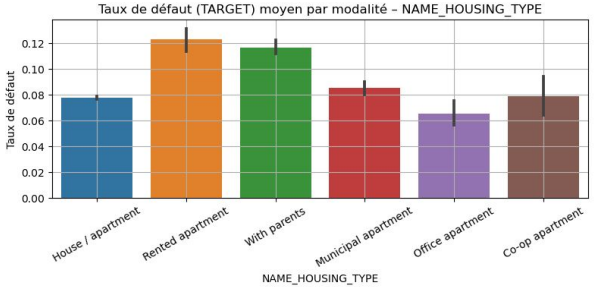


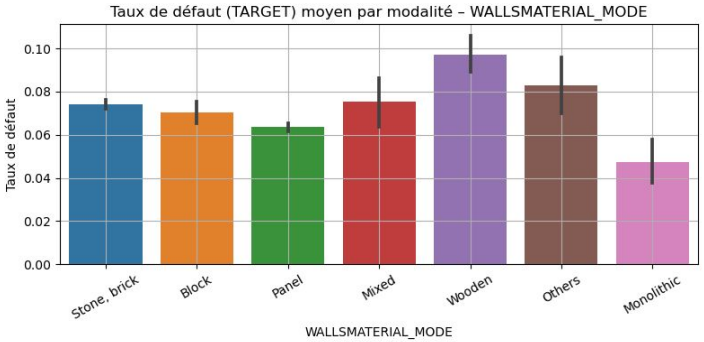



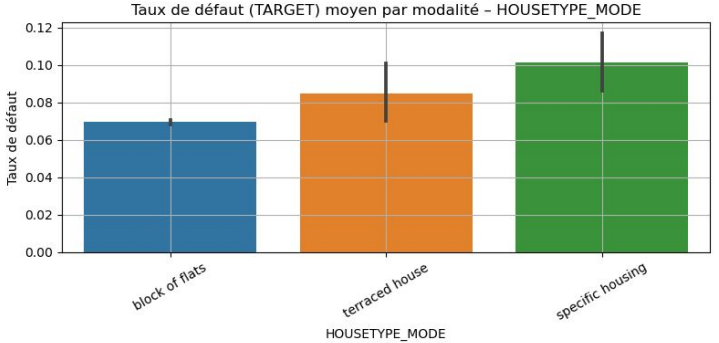



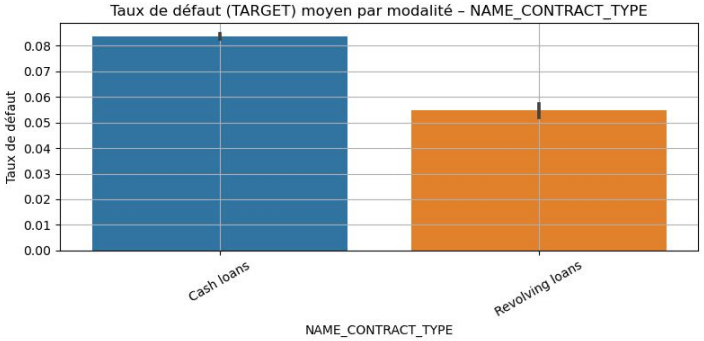



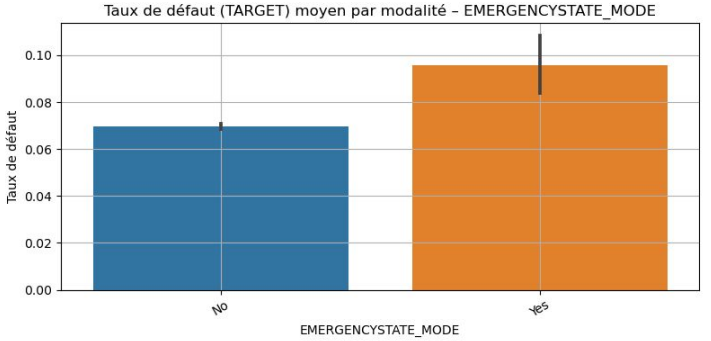
# Corrélation des variables avec TARGET

## Variables catégorielles (object) 2



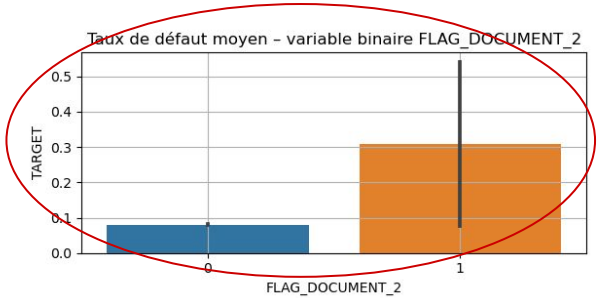



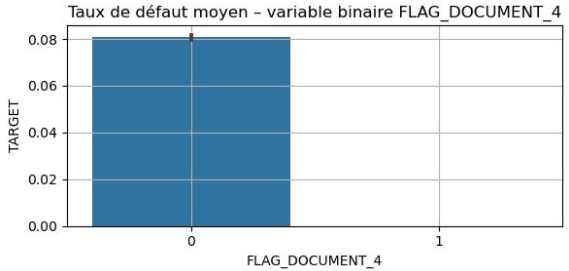



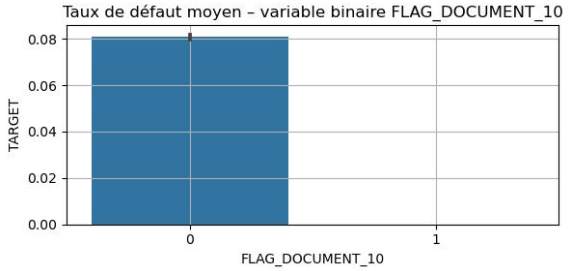



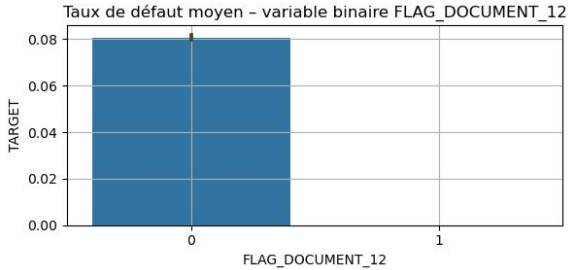
# Corrélation des variables avec TARGET

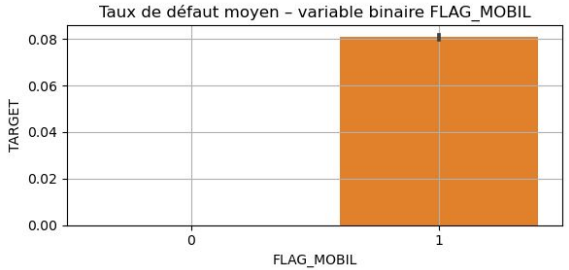
## Variables qualitatives binaires (int à 2 modalités) 1

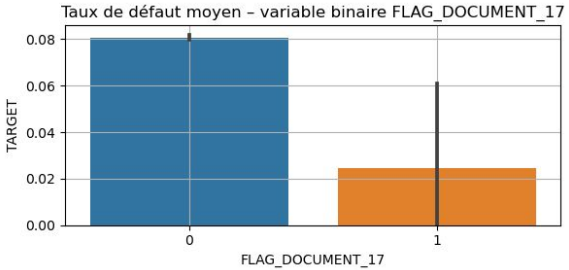


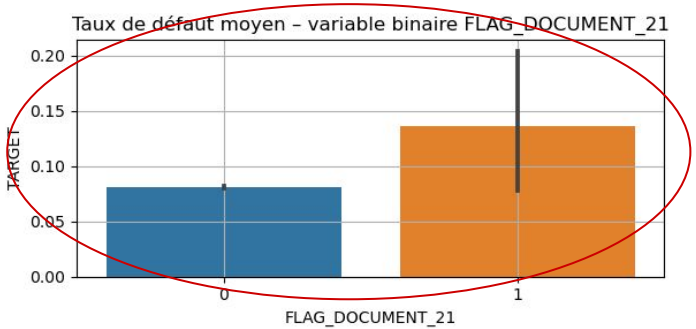



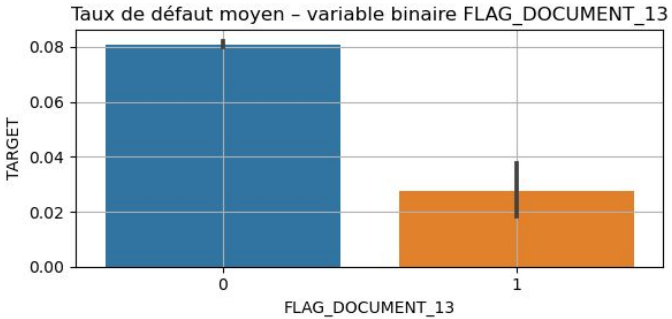



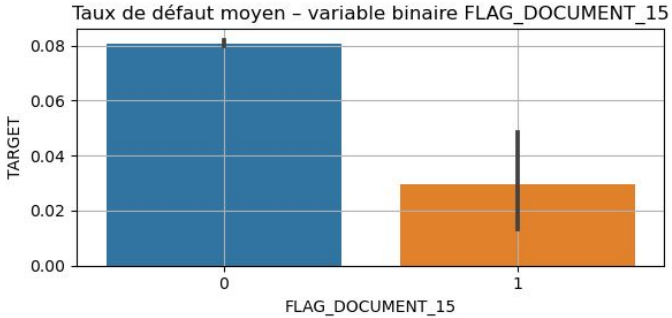



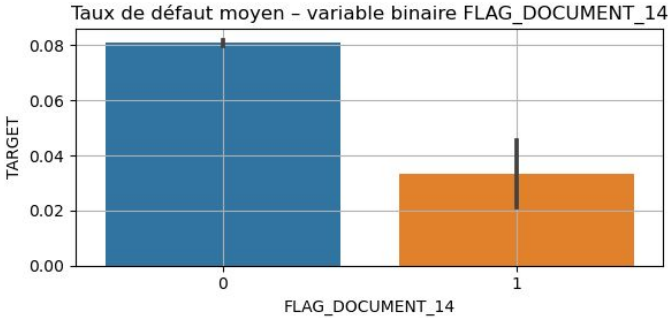
# Corrélation des variables avec TARGET

## Variables qualitatives binaires (int à 2 modalités) 2



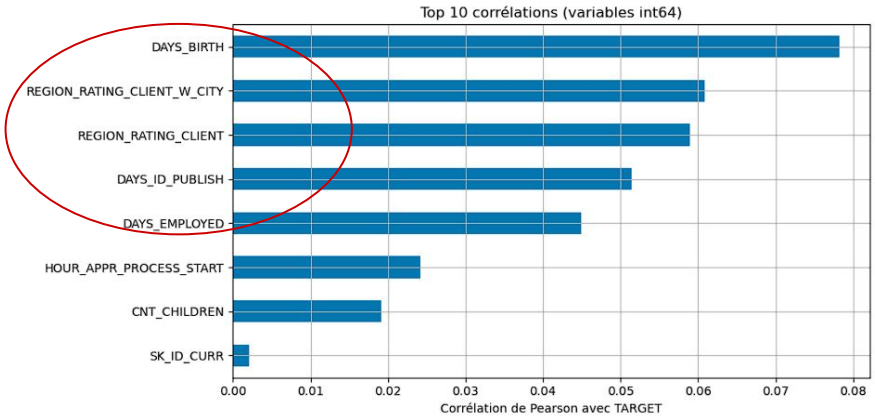



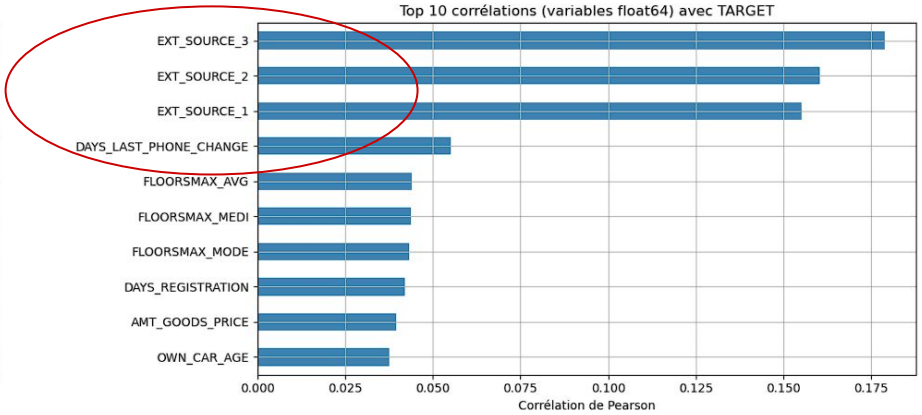



# Corrélation des variables avec TARGET

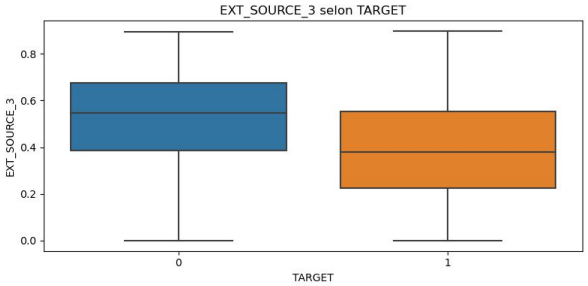
Variables quantitatives entières (2 modalités) - Variables quantitatives décimales

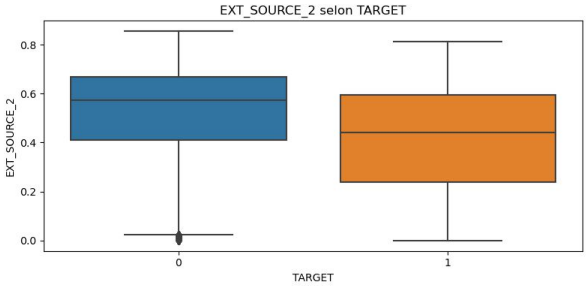


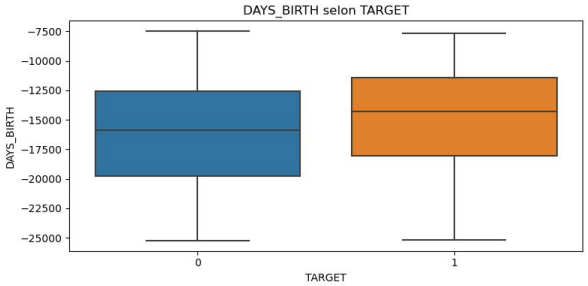



# Corrélation des variables avec TARGET

## Visualisation de quelques variables par classe TARGET



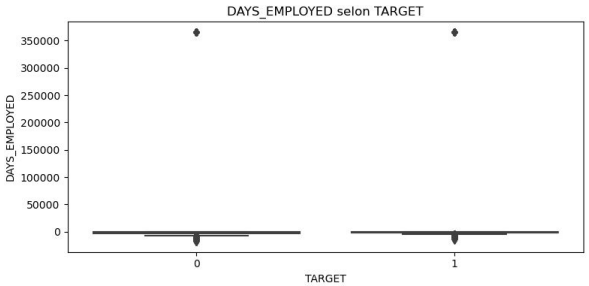



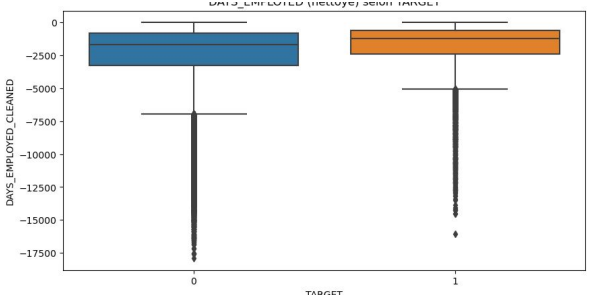




avant nettoyage

après nettoyage

Création d'une nouvelle variable booléenne

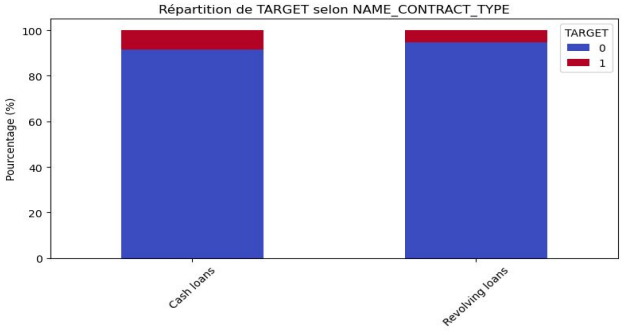


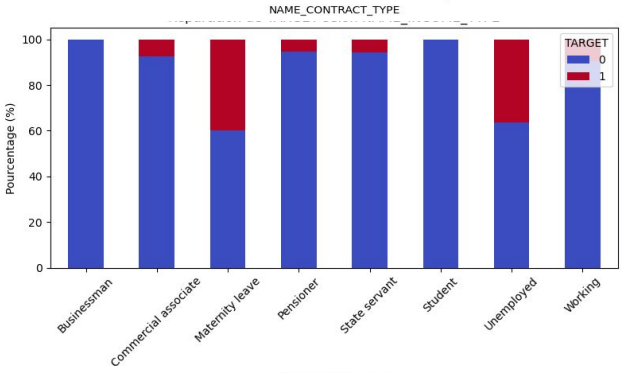



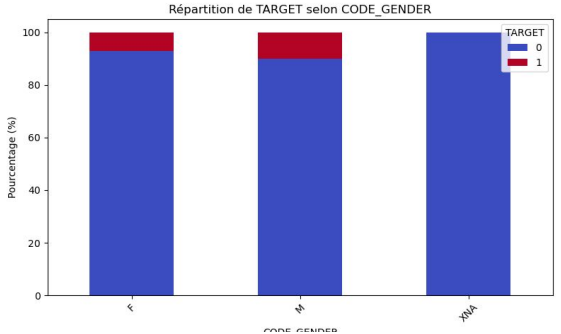

| | DAYS_EMPLOYED | DAYS_EMPLOYED_CLEANED | HAS_JOB |
|---|---|---|---|
| 195202 | -7376 | -7376.0 | 1 |
| 168379 | -1292 | -1292.0 | 1 |
| 47419 | -709 | -709.0 | 1 |
| 116263 | -1525 | -1525.0 | 1 |
| 64475 | -2264 | -2264.0 | 1 |

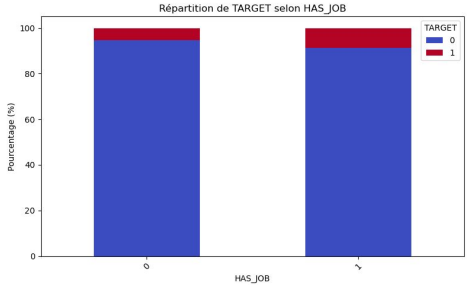
# Corrélation des variables avec TARGET

## Comparer la répartition de TARGET dans chaque catégorie



# Choix des variables après nettoyage et encodage

Data set de départ

```
Nombre de lignes : 307511
Nombre de colonnes : 122
```

data set avec variables retenues

```
Nombre de lignes : 98859
Nombre de colonnes : 41
```

Top5 variables pour modélisation

```
features = ['EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3', 'DAYS_BIRTH', 'AMT_INCOME_TOTAL']
```

Top10 variables pour modélisation

```
features_plus = [
    'EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3', 'DAYS_BIRTH',
    'DAYS_LAST_PHONE_CHANGE', 'DAYS_REGISTRATION', 'HAS_JOB',
    'OWN_CAR_AGE', 'CODE_GENDER_M', 'NAME_EDUCATION_TYPE_Lower secondary'
]
```

# métrique

RECALL = pour détecter les clients à risque

$$\text{Recall} = \frac{TP}{TP + FN}$$

PRECISION = pour détecter les client stable

$$\text{Precision} = \frac{TP}{TP + FP}$$

ACCURACY = mesure la proportions de prédictions correctes du modéle

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

F1-SCORE = Pour évaluer les modèle d'apprentissage automatique

Equilibre entre recall et precision

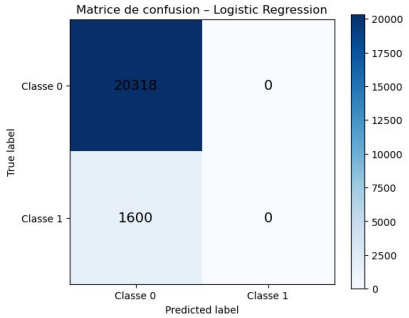
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

AUC = Principalement utilisée pour évaluer les performances des modèles de classification binaire

$$\text{AUC ROC} = \int_0^1 \text{TPR}(x)\,dx$$

# Modélisation

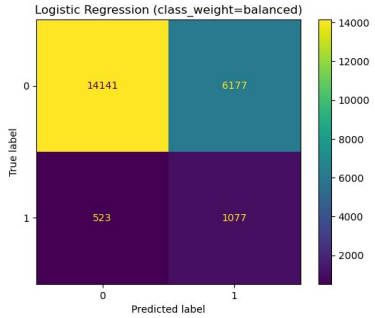
## Logistic regression sans / avec équilibrage




Rapport de classification :

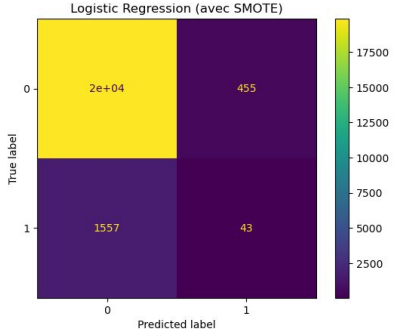
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 1.00 | 0.96 | 20318 |
| 1 | 0.00 | 0.00 | 0.00 | 1600 |
| accuracy | | | 0.93 | 21918 |
| macro avg | 0.46 | 0.50 | 0.48 | 21918 |
| weighted avg | 0.86 | 0.93 | 0.89 | 21918 |

AUC ROC : 0.5725




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.70 | 0.81 | 20318 |
| 1 | 0.15 | 0.67 | 0.24 | 1600 |
| accuracy | | | 0.69 | 21918 |
| macro avg | 0.56 | 0.68 | 0.53 | 21918 |
| weighted avg | 0.90 | 0.69 | 0.77 | 21918 |

AUC ROC : 0.7489

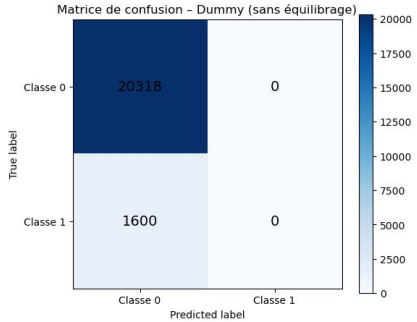



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.98 | 0.95 | 20318 |
| 1 | 0.09 | 0.03 | 0.04 | 1600 |
| accuracy | | | 0.91 | 21918 |
| macro avg | 0.51 | 0.50 | 0.50 | 21918 |
| weighted avg | 0.87 | 0.91 | 0.89 | 21918 |

AUC ROC : 0.5427

# Modélisation

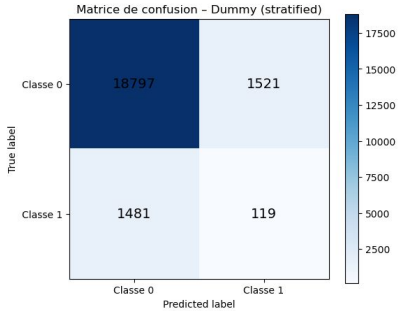
## Dummy sans / avec équilibrage




Rapport de classification :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 1.00 | 0.96 | 20318 |
| 1 | 0.00 | 0.00 | 0.00 | 1600 |
| accuracy | | | 0.93 | 21918 |
| macro avg | 0.46 | 0.50 | 0.48 | 21918 |
| weighted avg | 0.86 | 0.93 | 0.89 | 21918 |

AUC ROC : 0.5000

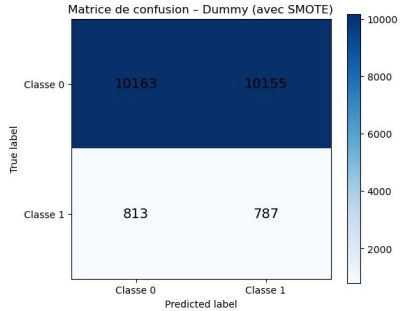



Dummy Stratified (approximation balanced) :

[[18797 1521]
[ 1481 119]]

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.93 | 0.93 | 20318 |
| 1 | 0.07 | 0.07 | 0.07 | 1600 |
| accuracy | | | 0.86 | 21918 |
| macro avg | 0.50 | 0.50 | 0.50 | 21918 |
| weighted avg | 0.86 | 0.86 | 0.86 | 21918 |

AUC ROC : 0.4998




Dummy Classifier avec SMOTE :

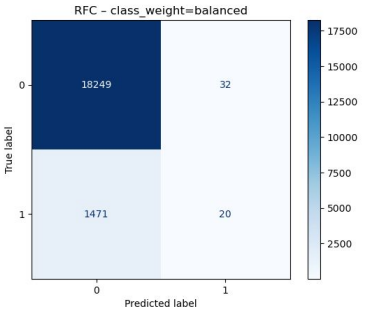
[[10163 10155]
[ 813 787]]

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.50 | 0.65 | 20318 |
| 1 | 0.07 | 0.49 | 0.13 | 1600 |
| accuracy | | | 0.50 | 21918 |
| macro avg | 0.50 | 0.50 | 0.39 | 21918 |
| weighted avg | 0.86 | 0.50 | 0.61 | 21918 |

AUC ROC : 0.4960

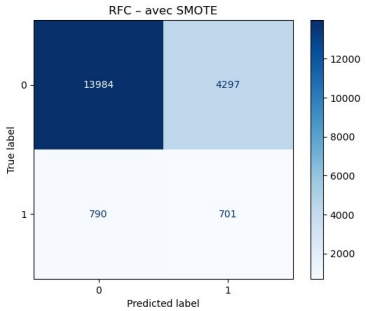
# Modélisation

## autres modèles



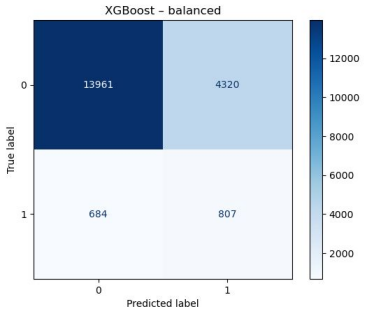

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 1.00 | 0.96 | 18281 |
| 1 | 0.38 | 0.01 | 0.03 | 1491 |
| accuracy | | | 0.92 | 19772 |
| macro avg | 0.66 | 0.51 | 0.49 | 19772 |
| weighted avg | 0.88 | 0.92 | 0.89 | 19772 |

AUC ROC : 0.7031



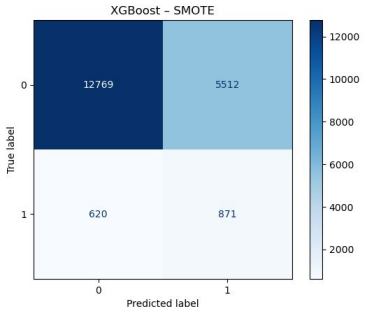

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.76 | 0.85 | 18281 |
| 1 | 0.14 | 0.47 | 0.22 | 1491 |
| accuracy | | | 0.74 | 19772 |
| macro avg | 0.54 | 0.62 | 0.53 | 19772 |
| weighted avg | 0.89 | 0.74 | 0.80 | 19772 |

AUC ROC : 0.6919




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.76 | 0.85 | 18281 |
| 1 | 0.16 | 0.54 | 0.24 | 1491 |
| accuracy | | | 0.75 | 19772 |
| macro avg | 0.56 | 0.65 | 0.55 | 19772 |
| weighted avg | 0.89 | 0.75 | 0.80 | 19772 |

AUC ROC: 0.7160580498838259

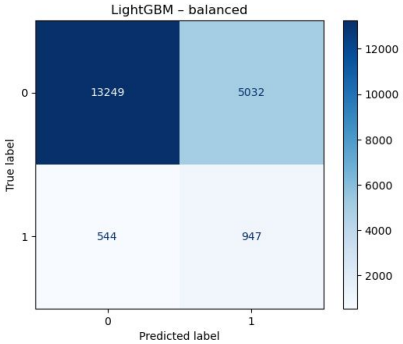



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.70 | 0.81 | 18281 |
| 1 | 0.14 | 0.58 | 0.22 | 1491 |
| accuracy | | | 0.69 | 19772 |
| macro avg | 0.55 | 0.64 | 0.51 | 19772 |
| weighted avg | 0.89 | 0.69 | 0.76 | 19772 |

AUC ROC: 0.6979976975431349

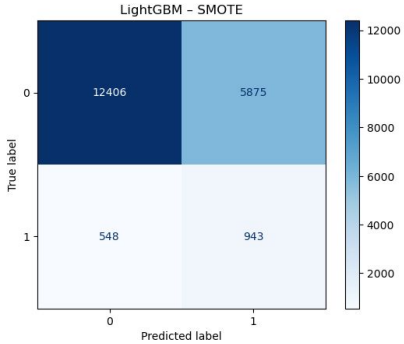
# Modélisation

## Modèles plus avancés



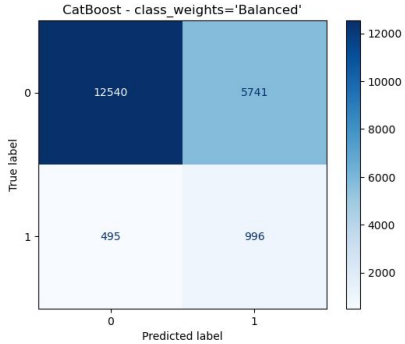

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.72 | 0.83 | 18281 |
| 1 | 0.16 | 0.64 | 0.25 | 1491 |
| accuracy | | | 0.72 | 19772 |
| macro avg | 0.56 | 0.68 | 0.54 | 19772 |
| weighted avg | 0.90 | 0.72 | 0.78 | 19772 |

AUC ROC: 0.7473520076753943



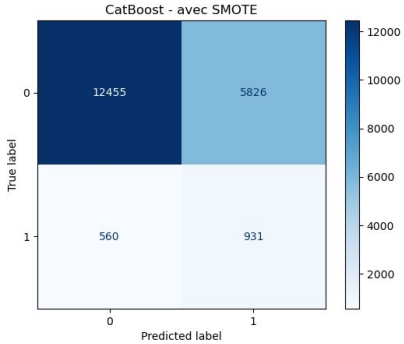

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.68 | 0.79 | 18281 |
| 1 | 0.14 | 0.63 | 0.23 | 1491 |
| accuracy | | | 0.68 | 19772 |
| macro avg | 0.55 | 0.66 | 0.51 | 19772 |
| weighted avg | 0.90 | 0.68 | 0.75 | 19772 |

AUC ROC: 0.7161414780827994




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.69 | 0.80 | 18281 |
| 1 | 0.15 | 0.67 | 0.24 | 1491 |
| accuracy | | | 0.68 | 19772 |
| macro avg | 0.55 | 0.68 | 0.52 | 19772 |
| weighted avg | 0.90 | 0.68 | 0.76 | 19772 |

AUC ROC : 0.7466




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.68 | 0.80 | 18281 |
| 1 | 0.14 | 0.62 | 0.23 | 1491 |
| accuracy | | | 0.68 | 19772 |
| macro avg | 0.55 | 0.65 | 0.51 | 19772 |
| weighted avg | 0.90 | 0.68 | 0.75 | 19772 |

AUC ROC : 0.7146

# Modèles avec top 5 features

| Modèle | M. équilibrage | AUC ROC | Accuracy | Recall | Précision | F1-score |
|---|---|---|---|---|---|---|
| Logistic Regression | Balanced | 0,7489 | 0,69 | 0,68 | 0,56 | 0,53 |
| Logistic Regression | Smote | 0,5427 | 0,91 | 0,50 | 0,51 | 0,50 |
| Dummy | Stratified | 0,4998 | 0,86 | 0,50 | 0,50 | 0,50 |
| Dummy | Smote | 0,4960 | 0,50 | 0,50 | 0,50 | 0,39 |
| RFC | Balanced | 0,7031 | 0,92 | 0,51 | 0,66 | 0,49 |
| RFC | Smote | 0,6919 | 0,74 | 0,62 | 0,54 | 0,53 |
| XGBoost | Balanced | 0,7160 | 0,75 | 0,65 | 0,56 | 0,55 |
| XGBoost | Smote | 0,6979 | 0,69 | 0,64 | 0,55 | 0,51 |
| LightGBM | Balanced | 0,7473 | 0,72 | 0,68 | 0,56 | 0,54 |
| LightGBM | Smote | 0,7161 | 0,68 | 0,66 | 0,55 | 0,51 |
| CatBoost | Balanced | 0,7466 | 0,68 | 0,68 | 0,55 | 0 |
| CatBoost | Smote | 0,7146 | 0,68 | 0,65 | 0,55 | 0,51 |

# Modèles avec top 10 features

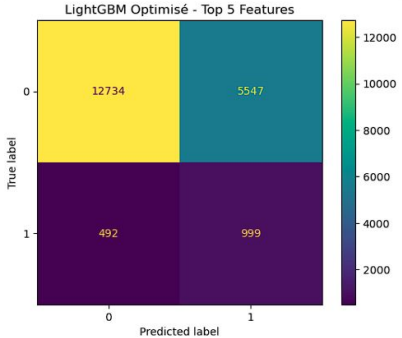
| Modèle | M. équilibrage | AUC ROC | Accuracy | Recall | Précision | F1-score |
|---|---|---|---|---|---|---|
| Logistic Regression | Balanced | 0,7494 | 0,69 | 0,68 | 0,56 | 0,52 |
| Logistic Regression | Smote | 0,7295 | 0,71 | 0,67 | 0,56 | 0,53 |
| Dummy | Stratified | 0,4973 | 0,86 | 0,50 | 0,50 | 0,50 |
| Dummy | Smote | 0,4990 | 0,50 | 0,50 | 0,50 | 0,39 |
| RFC | Balanced | 0,7158 | 0,92 | 0,50 | 0,73 | 0,49 |
| RFC | Smote | 0,6830 | 0,81 | 0,58 | 0,54 | 0,55 |
| XGBoost | Balanced | 0,7106 | 0,76 | 0,65 | 0,56 | 0,55 |
| XGBoost | Smote | 0,6923 | 0,82 | 0,61 | 0,56 | 0,56 |
| LightGBM | Balanced | 0,7436 | 0,72 | 0,68 | 0,56 | 0,54 |
| LightGBM | Smote | 0,7028 | 0,82 | 0,61 | 0,55 | 0,56 |
| CatBoost | Balanced | 0,7429 | 0,69 | 0,68 | 0,56 | 0,53 |
| CatBoost | Smote | 0,6988 | 0,83 | 0,60 | 0,55 | 0,56 |

# Optimisation

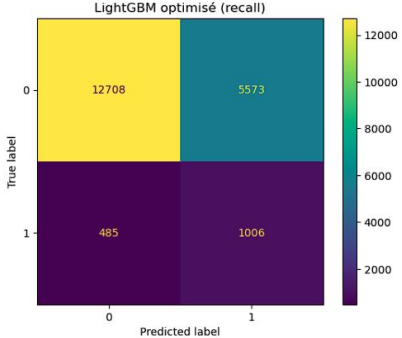
```
🔍 Corrélations avec la cible (TARGET)
AGE_YEARS                -0.061470
IS_YOUNG                  0.048316
PHONE_CHANGED_RECENTLY    0.023710
ANNUITY_INCOME_RATIO      0.018104
CREDIT_INCOME_RATIO      -0.010299
HAS_JOB                        NaN
Name: TARGET, dtype: float64
```

```
features_enriched = [
    'EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
    'DAYS_BIRTH', 'AMT_INCOME_TOTAL',
    'AGE_YEARS', 'IS_YOUNG', 'PHONE_CHANGED_RECENTLY',
    'ANNUITY_INCOME_RATIO', 'CREDIT_INCOME_RATIO'
]
```

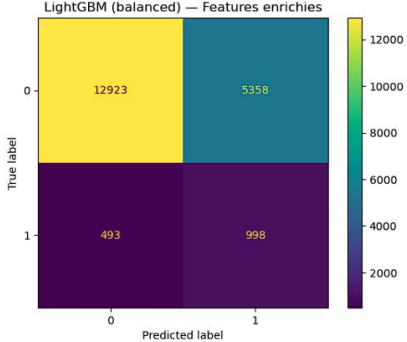
```
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.70      0.81     18281
           1       0.15      0.67      0.25      1491

    accuracy                           0.69     19772
   macro avg       0.56      0.68      0.53     19772
weighted avg       0.90      0.69      0.77     19772

AUC ROC : 0.7488
```



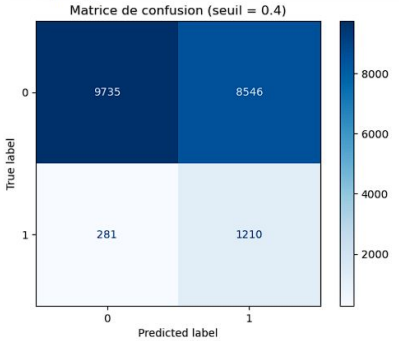

```
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.70      0.81     18281
           1       0.15      0.67      0.25      1491

    accuracy                           0.69     19772
   macro avg       0.56      0.68      0.53     19772
weighted avg       0.90      0.69      0.77     19772

AUC ROC : 0.7509
```



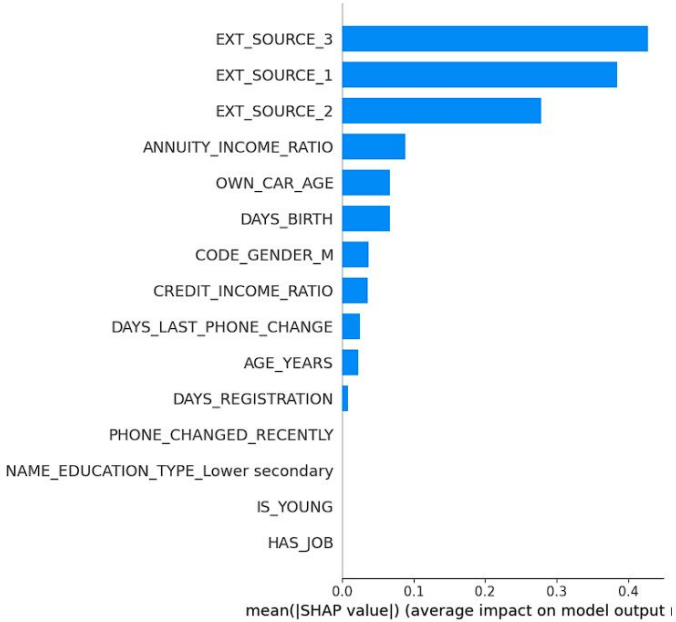

```
📊 Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.71      0.82     18281
           1       0.16      0.67      0.25      1491

    accuracy                           0.70     19772
   macro avg       0.56      0.69      0.53     19772
weighted avg       0.90      0.70      0.77     19772

AUC ROC : 0.7546
```

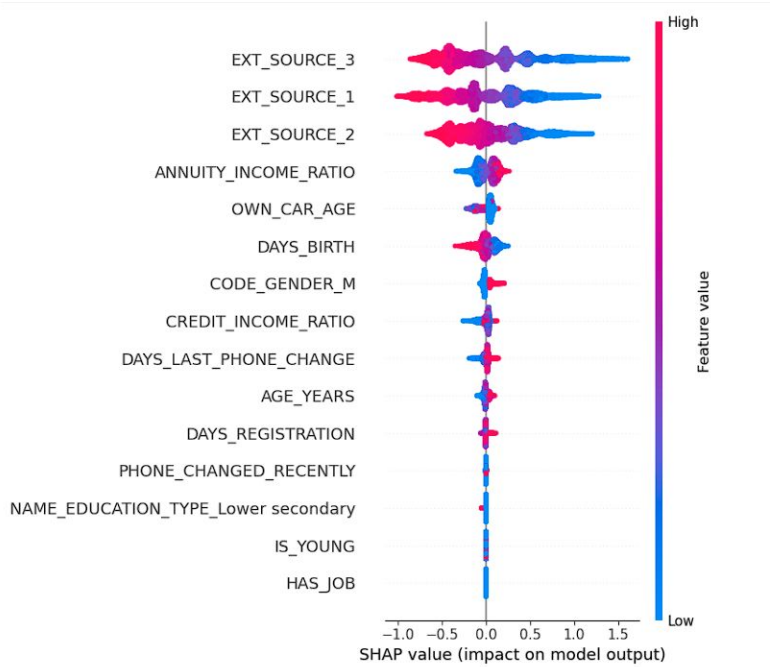



```
📌 Résultats avec seuil personnalisé = 0.4

              precision    recall  f1-score   support

           0       0.972     0.533     0.688     18281
           1       0.124     0.812     0.215      1491

    accuracy                           0.554     19772
   macro avg       0.548     0.672     0.452     19772
weighted avg       0.908     0.554     0.652     19772

AUC ROC (probas) : 0.7545
```



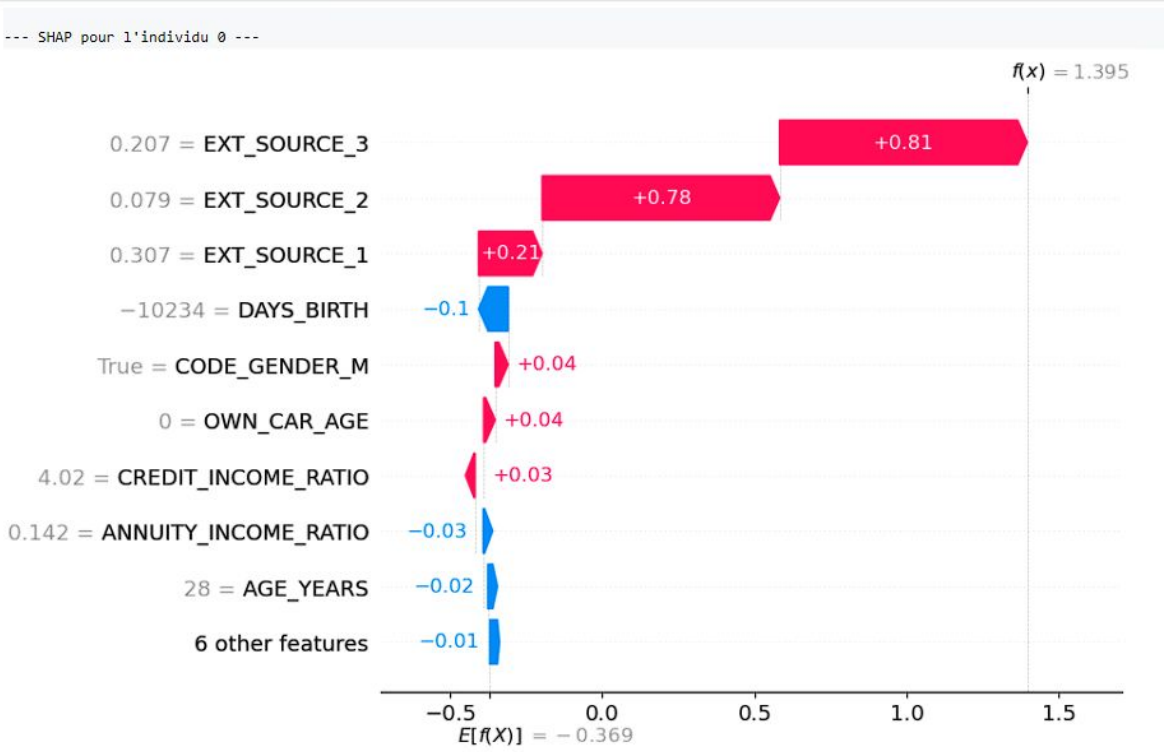
# Explication globale



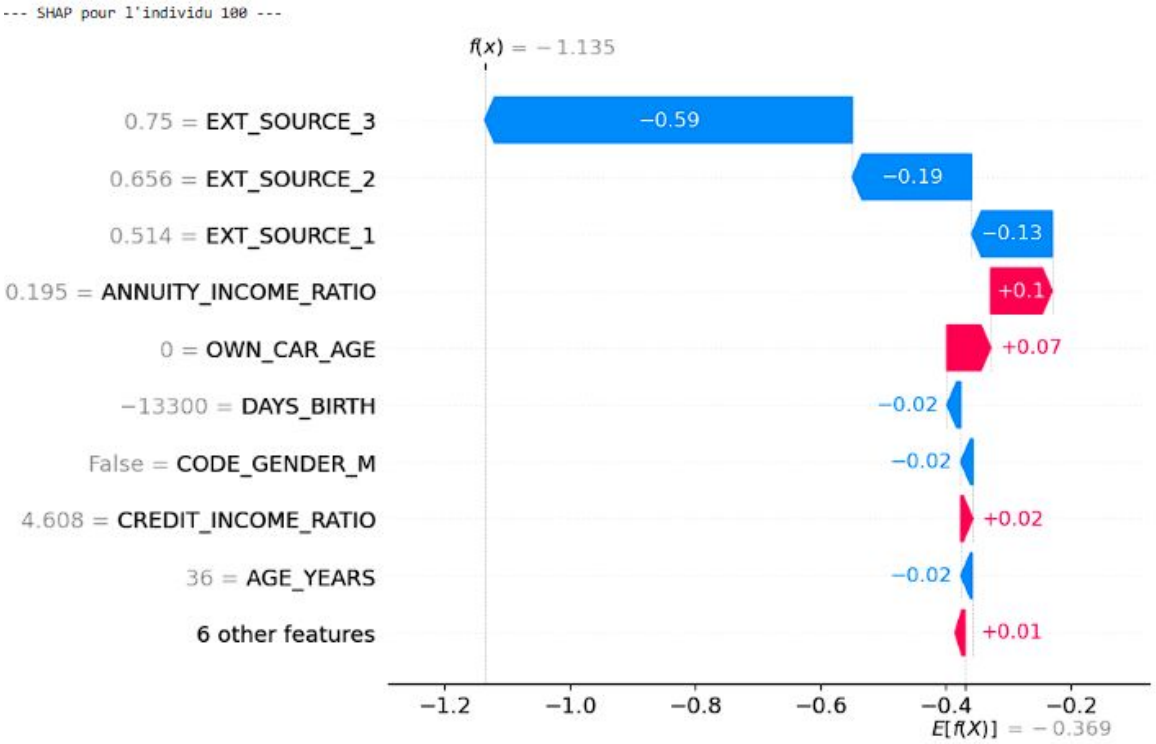
# Explication locale

## Cas positive

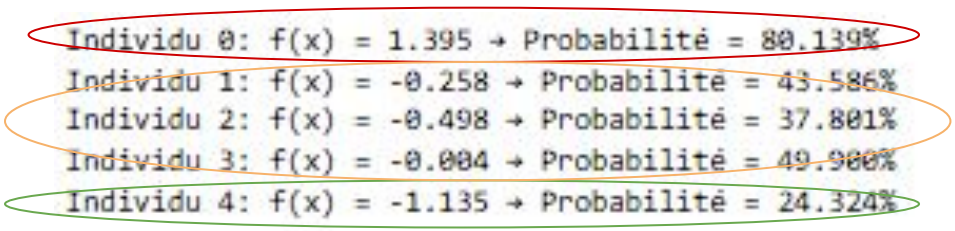


# Explication locale

cas négative



# Explication locale




Client à risque

Incertain

Client stable

# Conclusion

. Le modèle est loin d'être parfait mais avec les données en notre possession il est difficile de mieux faire.

. Le modèle permet de détecter environ 67 à 81 % des clients à risque mais rate encore 281 a 492 clients à risque ce qui peux être critique en contexte opérationnel (exemple: crédit, assurance).

. Pour améliorer notre modèle il faudra collecter ou enrichir les données avec de nouvelles features.

. Ce que l'on comprend déjà sont les variables qui influencent le plus les décisions et on peut justifier individuellement pourquoi un client est considéré à risque ou non.