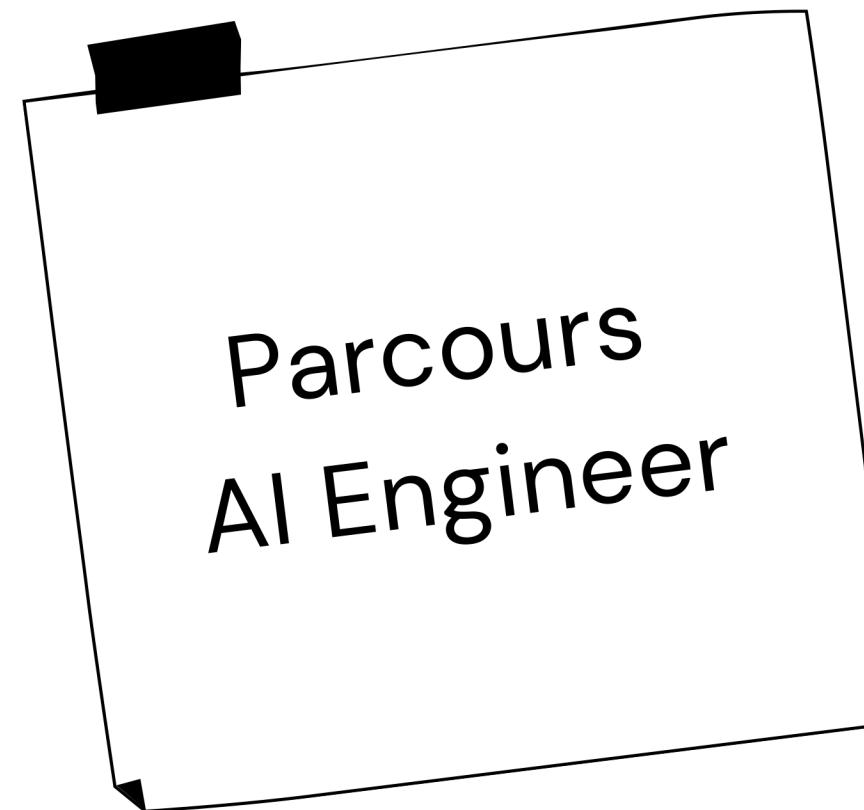
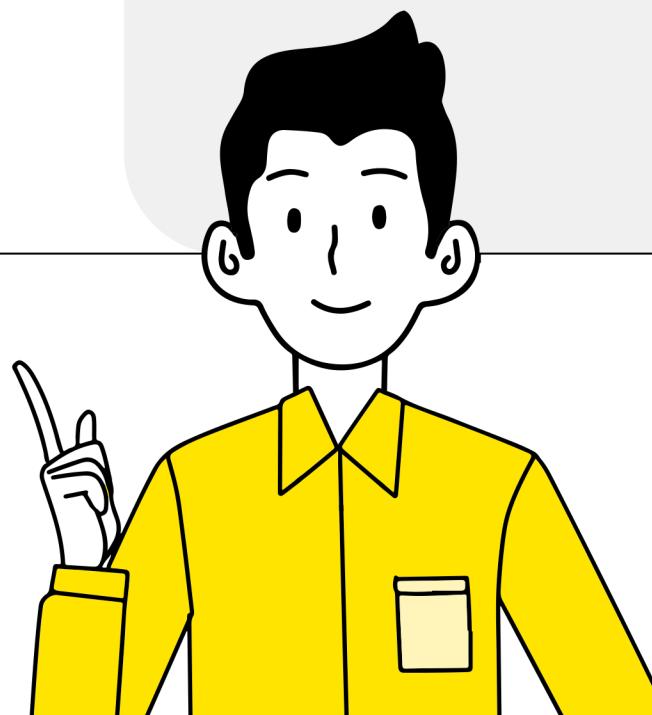


# Construction d'un modèle de scoring



**Notre mission:**

**Créer un modèle pour reconnaître  
les individus à risque**



# Les étapes d'analyse

**Partie 1**

**Exploration et  
Documentation**

**Partie 2**

**Observation**

**Partie 3**

**Test de modèles de  
Machine Learning**

**Partie 4**

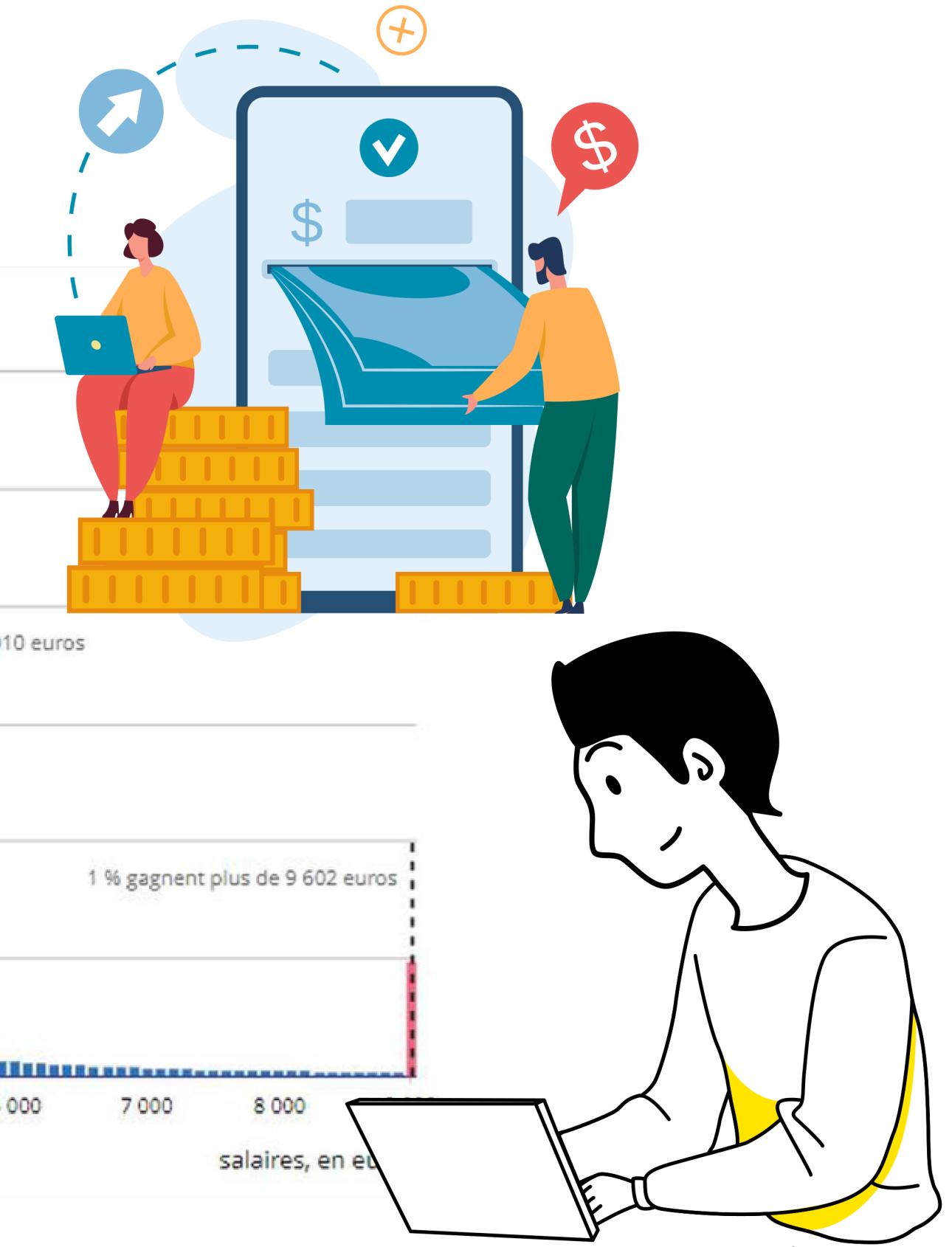
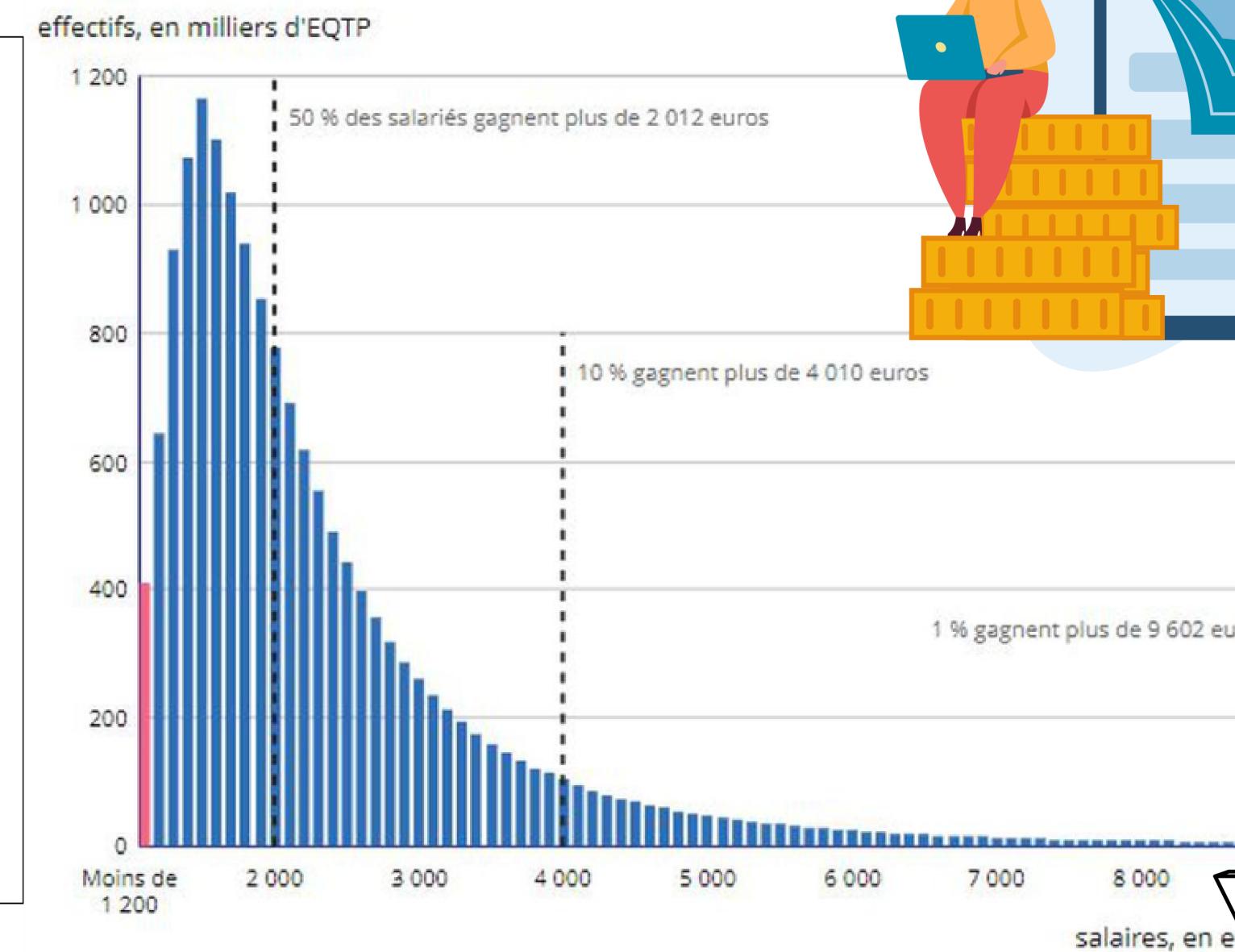
**Conclusion**

# **Partie 1: Exploration et documentation.**

# Partie 1: Exploration et documentation.

## Source externe

- Kaggle
- Internet



# Partie 1: Exploration et documentation.

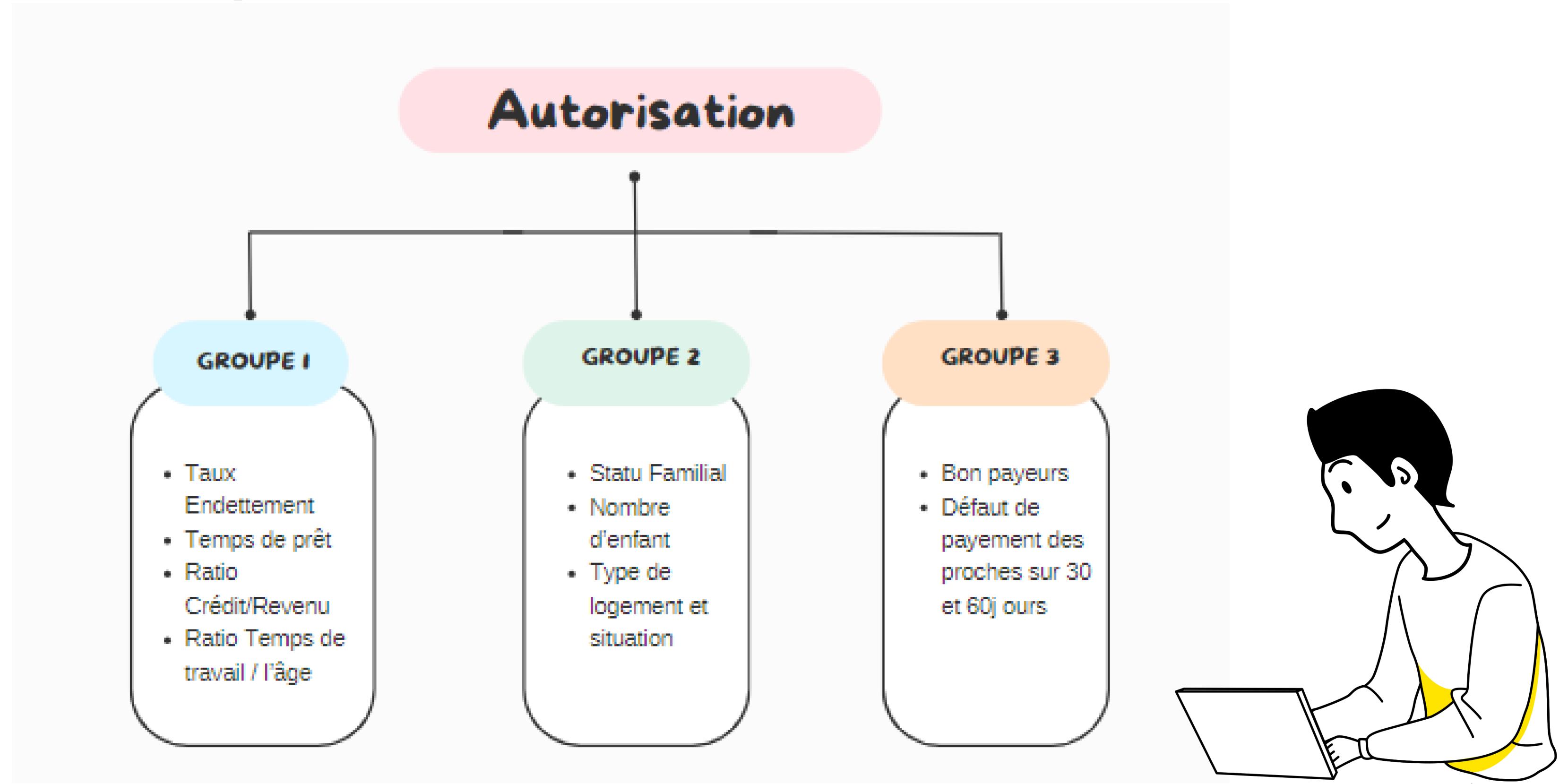
## Jeu de données

- Nombreux “.csv”
- Exploration de “train”
- La colonne ‘TARGET’

```
with open('HomeCredit_columns_description.csv', 'r', encoding='Windows-1252') as fichier:  
    contenu = fichier.read()  
  
print(contenu)  
  
,Table,Row,Description,Special  
1,application_{train|test}.csv,SK_ID_CURR, ID of loan in our sample,  
2,application_{train|test}.csv,TARGET,"Target variable (1 - client with payment difficulties  
ne of the first Y installments of the loan in our sample, 0 - all other cases)",  
5,application_{train|test}.csv,NAME_CONTRACT_TYPE,Identification if loan is cash or revolving  
6,application_{train|test}.csv,CODE_GENDER,Gender of the client,  
7,application_{train|test}.csv,FLAG_OWN_CAR,Flag if the client owns a car,  
8,application_{train|test}.csv,FLAG_OWN_REALTY,Flag if client owns a house or flat,  
9,application_{train|test}.csv,CNT_CHILDREN,Number of children the client has,  
  
'TARGET',  
'NAME_CONTRACT_TYPE',  
'CODE_GENDER',  
'FLAG_OWN_CAR',  
'FLAG_OWN_REALTY',  
'CNT_CHILDREN',  
'AMT_INCOME_TOTAL',  
'AMT_CREDIT',  
'AMT_ANNUITY',  
'AMT_GOODS_PRICE',  
'NAME_TYPE_SUITE',  
'NAME_INCOME_TYPE',  
'NAME_EDUCATION_TYPE',  
'NAME_FAMILY_STATUS',  
'NAME_HOUSING_TYPE',
```



# Partie 1: Exploration et documentation.



# Partie 1: Exploration et documentation.

## Liste de manipulations

- **Ajustement** de quelque colonne avec **médian**
- **Beaucoup seront considéré atypique:**  
Nous n'avons pas assez de détail sur la politique de la société financière.
- **Calcul et création de colonne :**
  - Age et Ancienneté**, remise en année.
  - Temps prêt et Taux endettement**
    - Score Métier**
- **Numérisation** des valeurs avec :  
**Label Encoding et One Hot Encoding**



# Partie 1: Exploration et documentation.

**Calcul du  
Score Métier :**

## Observation de

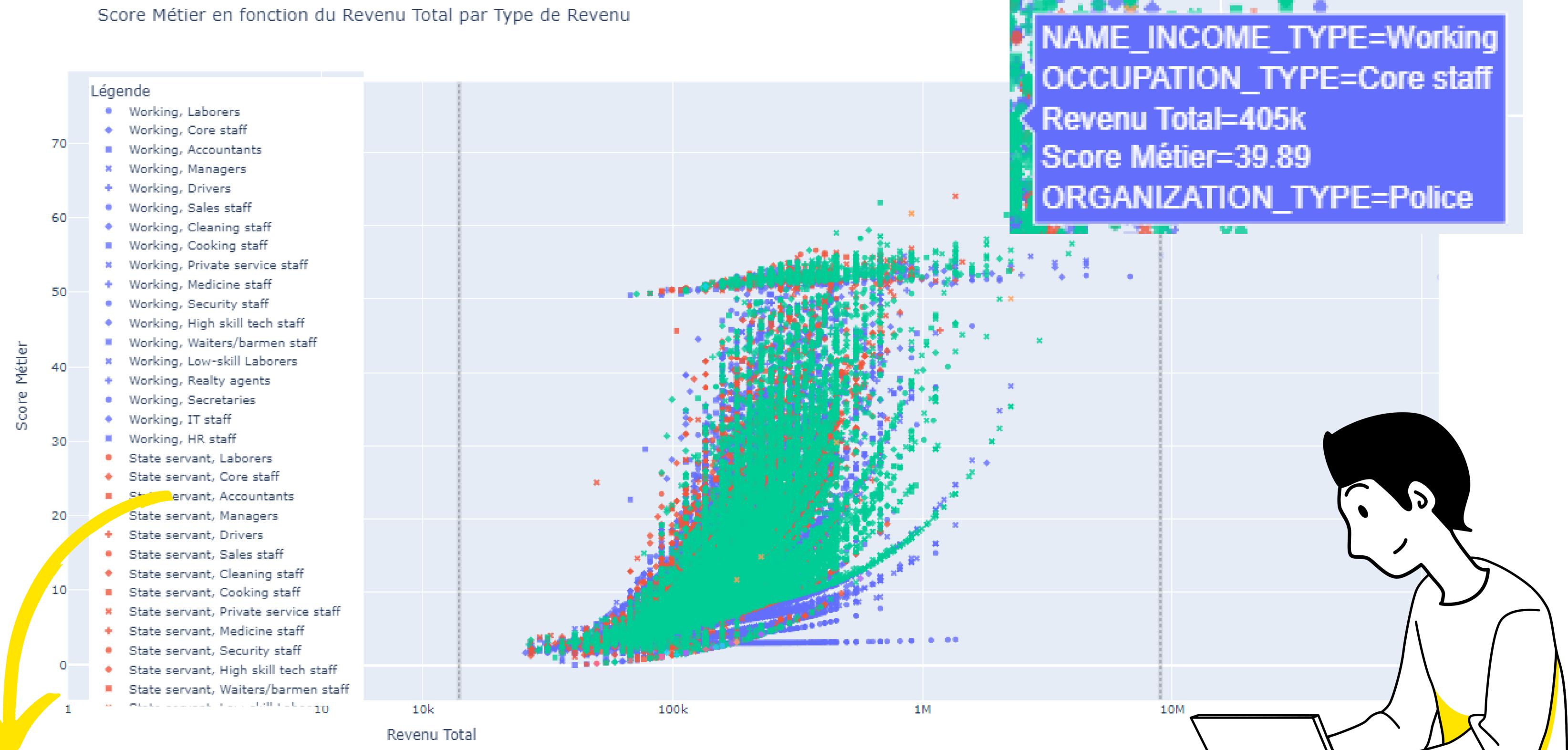
- 'NAME\_INCOME\_TYPE',
- 'OCCUPATION\_TYPE',
- 'ORGANIZATION\_TYPE'

## Par rapport à

- AMT\_INCOME\_TOTAL



# Partie 1: Exploration et documentation.



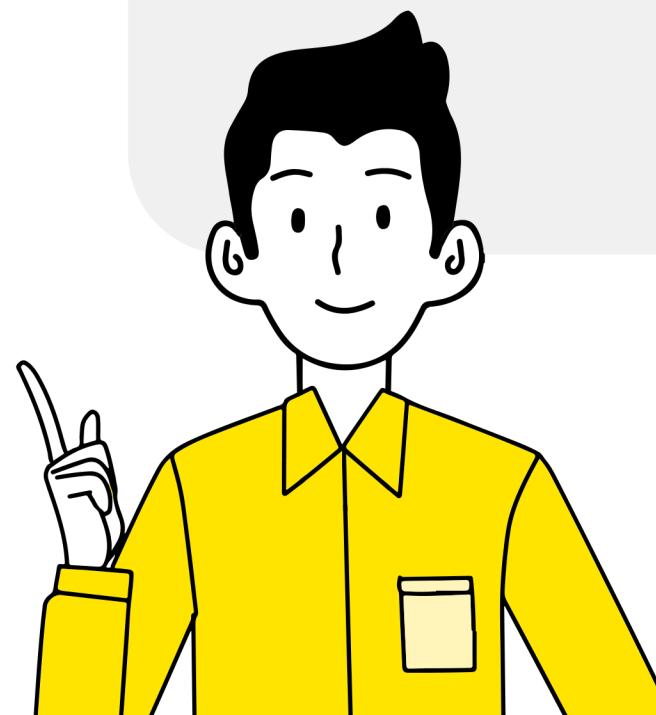
Légende de 69 combinaisons



## **Partie 2: Observation**

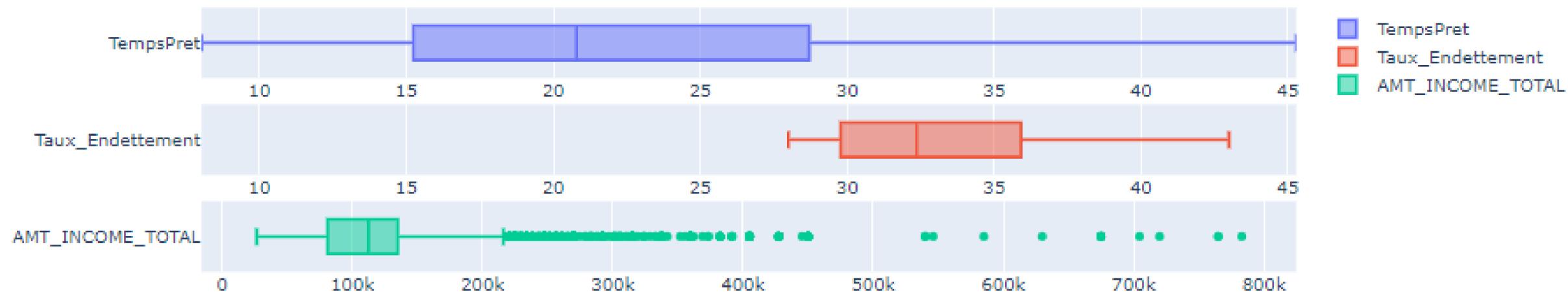
**Question.**

**Arriver à anticiper les Faux Positifs  
et les Faux Négatifs**



# Partie 2: Observation

Boxplots horizontaux pour TempsPret et Taux\_Endettement avec échelle x identique



```
[42]: # Calcul de la somme de 'AMT_ANNUITY' pour les individus avec un 'Taux_Endettement' entre 28 et 33
somme_amt_annuity_28_33 = individus_filtrés[(individus_filtrés['Taux_Endettement'] > 28) & (individus_filtrés['Taux_Endettement'] <= 33)]['AMT_ANNUITY']

# Calcul de la somme de 'AMT_ANNUITY' pour les individus avec un 'Taux_Endettement' entre 33 et 43
somme_amt_annuity_33_43 = individus_filtrés[(individus_filtrés['Taux_Endettement'] > 33) & (individus_filtrés['Taux_Endettement'] <= 43)]['AMT_ANNUITY']

somme_amt_annuity_28_33, somme_amt_annuity_33_43

[42]: (719201106.0, 624640720.5)

719.201.106.0, 624.640.720.5
```

Il y a pour **1,34 Milliards** de prêts concerné



# Partie 2: Observation

## Le cas 'TARGET':

```
Most Positive Correlations:  
    OCCUPATION_TYPE_Laborers          0.037696  
    DAYS_ID_PUBLISH                   0.040007  
    NAME_INCOME_TYPE_Working          0.041922  
    FLAG_DOCUMENT_3                   0.042261  
    REG_CITY_NOT_LIVE_CITY            0.043037  
    REG_CITY_NOT_WORK_CITY            0.043163  
    CODE_GENDER_M                     0.047624  
    NAME_EDUCATION_TYPE_Secondary / secondary special 0.060637  
    DAYS_LAST_PHONE_CHANGE             0.061375  
    REGION_RATING_CLIENT              0.063884  
    DAYS_BIRTH                         0.065899  
    REGION_RATING_CLIENT_W_CITY       0.066265  
    DAYS_EMPLOYED                      0.068157  
    TARGET                            1.000000  
    FLAG_MOBIL                         NaN  
  
Name: TARGET, dtype: float64  
  
Most Negative Correlations:  
    EXT_SOURCE_3                      -0.180317  
    EXT_SOURCE_2                      -0.171309  
    EXT_SOURCE_1                      -0.154996  
    Nbr_AnnéeTravail                  -0.068157  
    NAME_EDUCATION_TYPE_Higher education -0.067365  
    Age                               -0.065919  
    RatioVieTravail                  -0.063894  
    AMT_GOODS_PRICE                    -0.050614  
    CODE_GENDER_F                      -0.047617  
    FLOORSMAX_AVG                     -0.047458  
    FLOORSMAX_MEDI                    -0.047208  
    FLOORSMAX_MODE                    -0.046685  
    OWN_CAR_AGE                       -0.045855  
    EMERGENCYSTATE_MODE_No             -0.042933  
    HOUSETYPE_MODE_block of flats     -0.041457  
  
Name: TARGET, dtype: float64
```

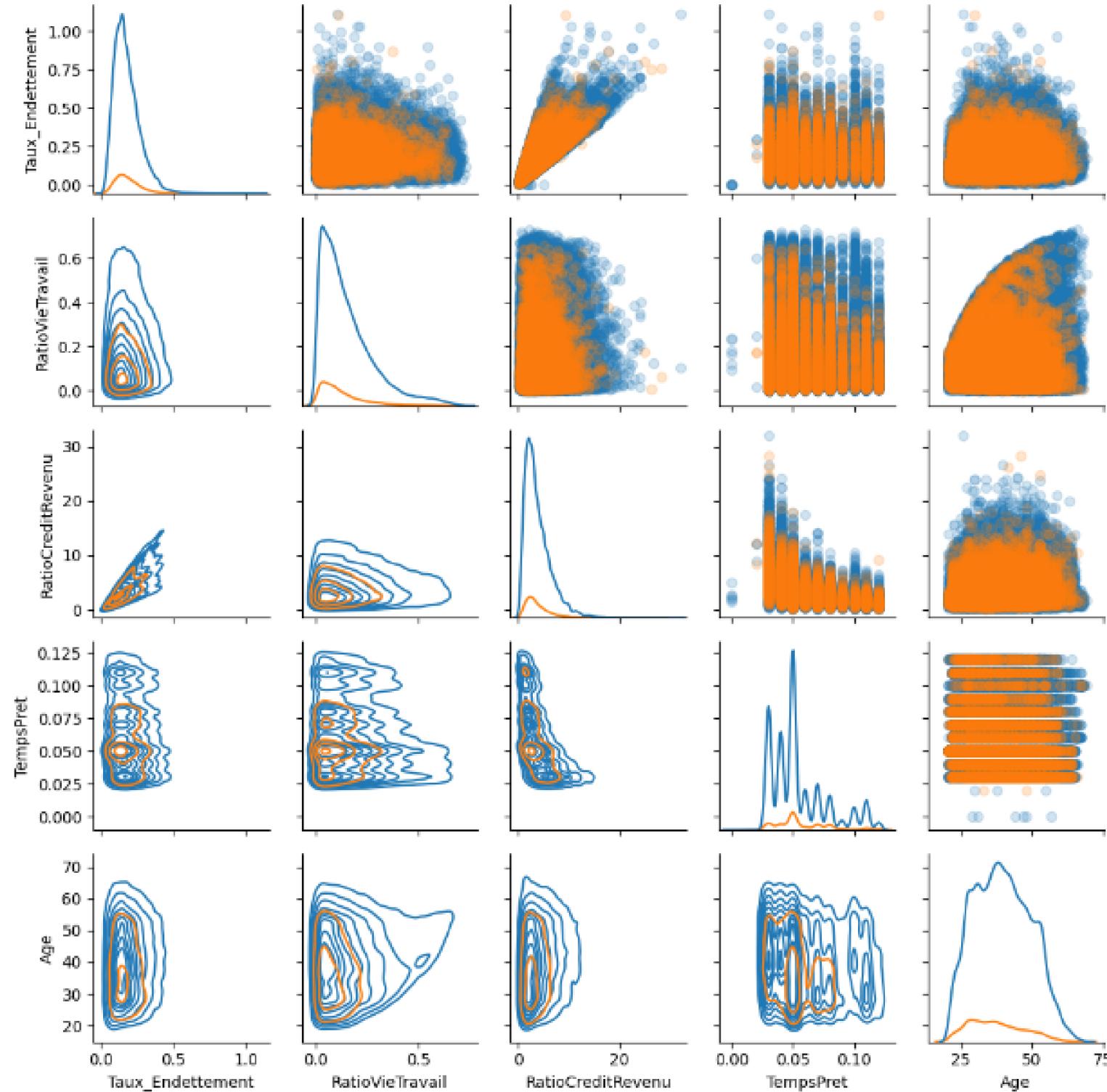


# Partie 2: Observation

## Corrélation des groupes :

### GROUPE I

- Taux Endettement
- Temps de prêt
- Ratio Crédit/Revenu
- Ratio Temps de travail / l'âge



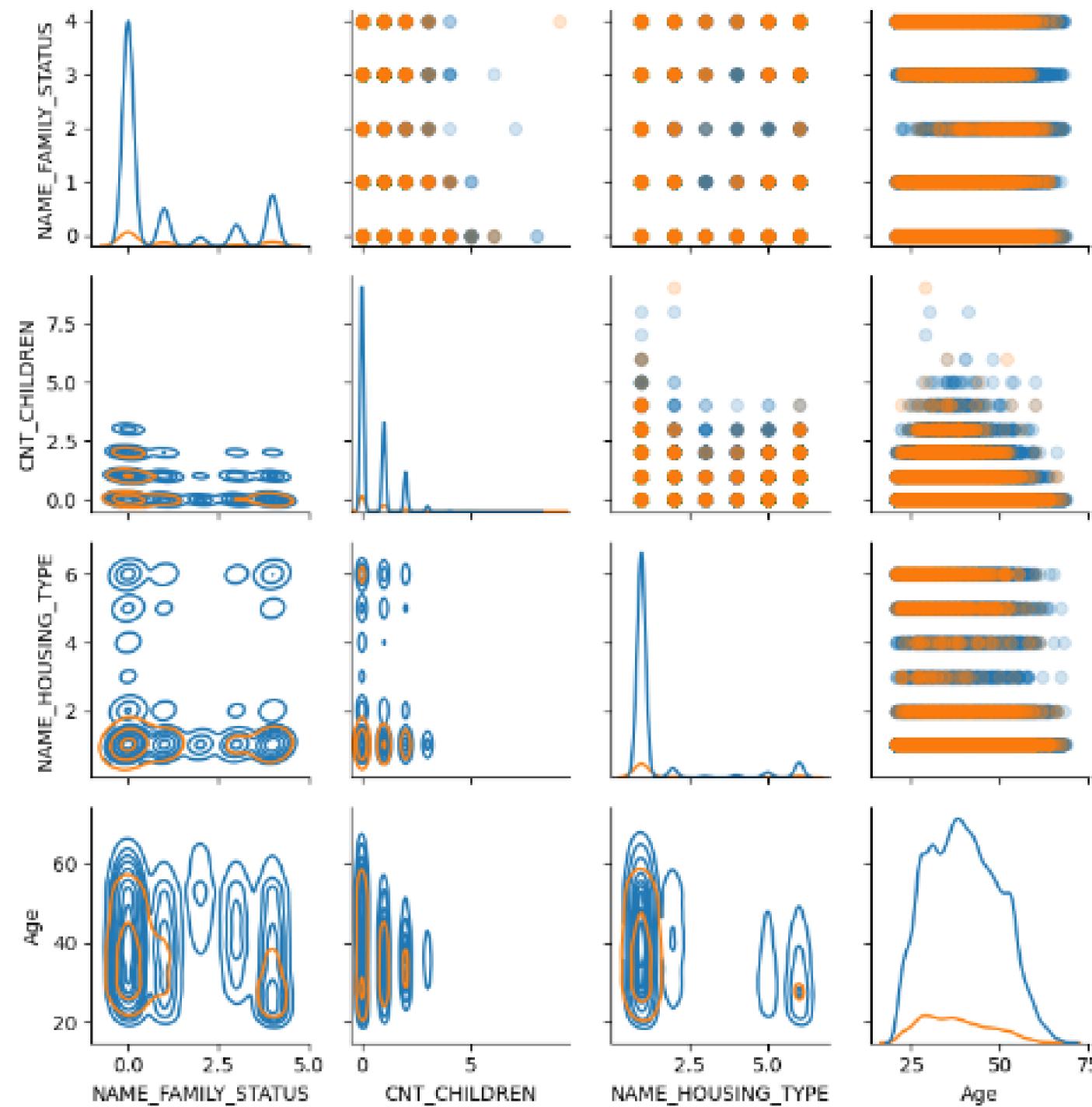
# Partie 2: Observation

## Corrélation des groupes :

### GROUPE 2

- Statu Familial
- Nombre d'enfant
- Type de logement et situation

Ext Source and Age Features Pairs Plot

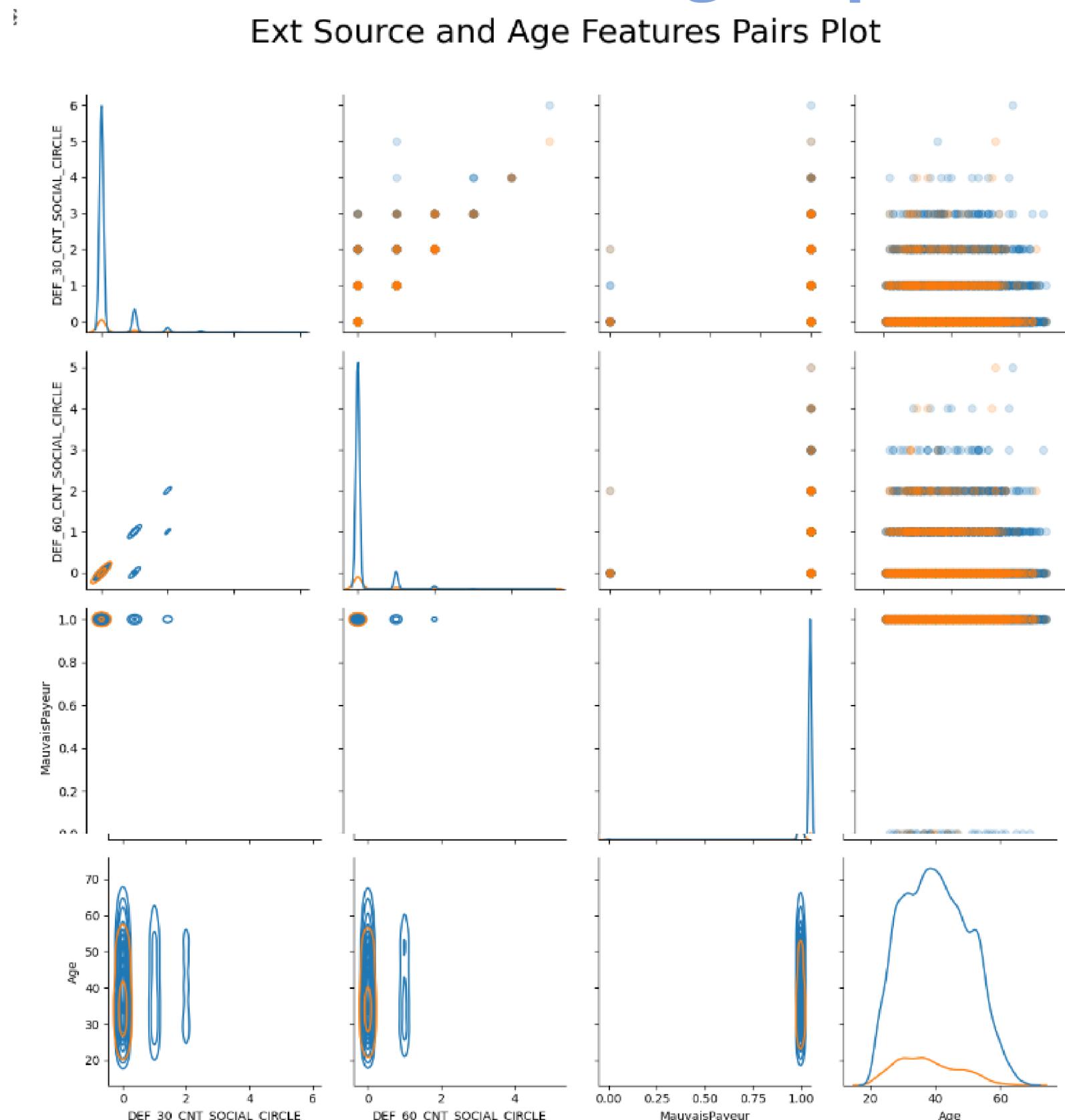


# Partie 2: Observation

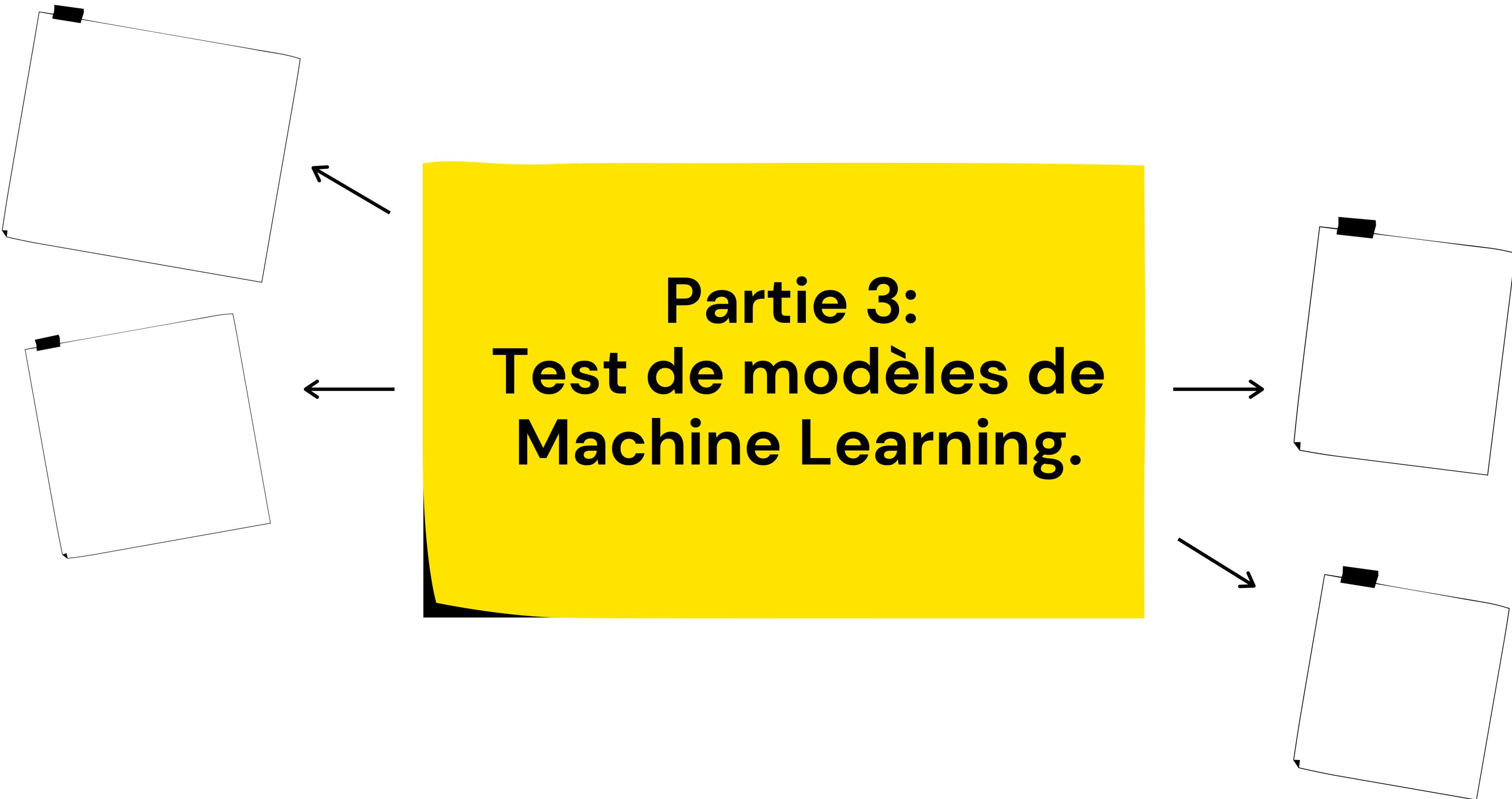
## Corrélation des groupes :

### GROUPE 3

- Bon payeurs
- Défaut de paiement des proches sur 30 et 60j ours



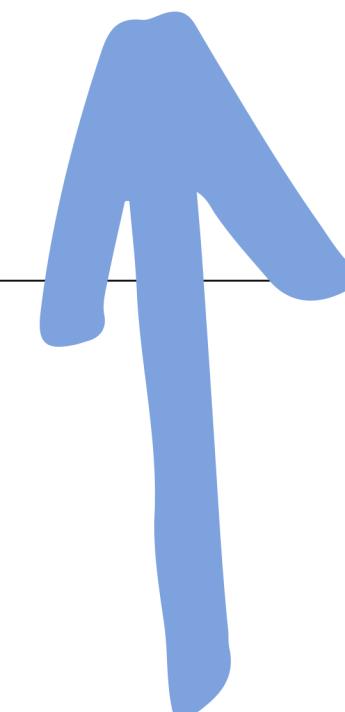
## **Partie 3: Test de modèles de Machine Learning.**



# Test de 3 modèles

modèle 1

Random Forest



modèle 2

Booster Gradiant

modèle 3

Classification et  
HyperParamètre

# Partie 3: Test de modèles de Machine Learning

modèle 1

## Le Random Forest

On construit un grand nombre d'arbres de décision lors de l'entraînement, puis on fait la moyenne des résultats (régression).

L'utilisation de plusieurs arbres permet de réduire l'overfitting et d'améliorer la précision du modèle.

Nos Paramètres :

n\_estimators = 100, random\_state = 50, verbose = 1, n\_jobs = -1

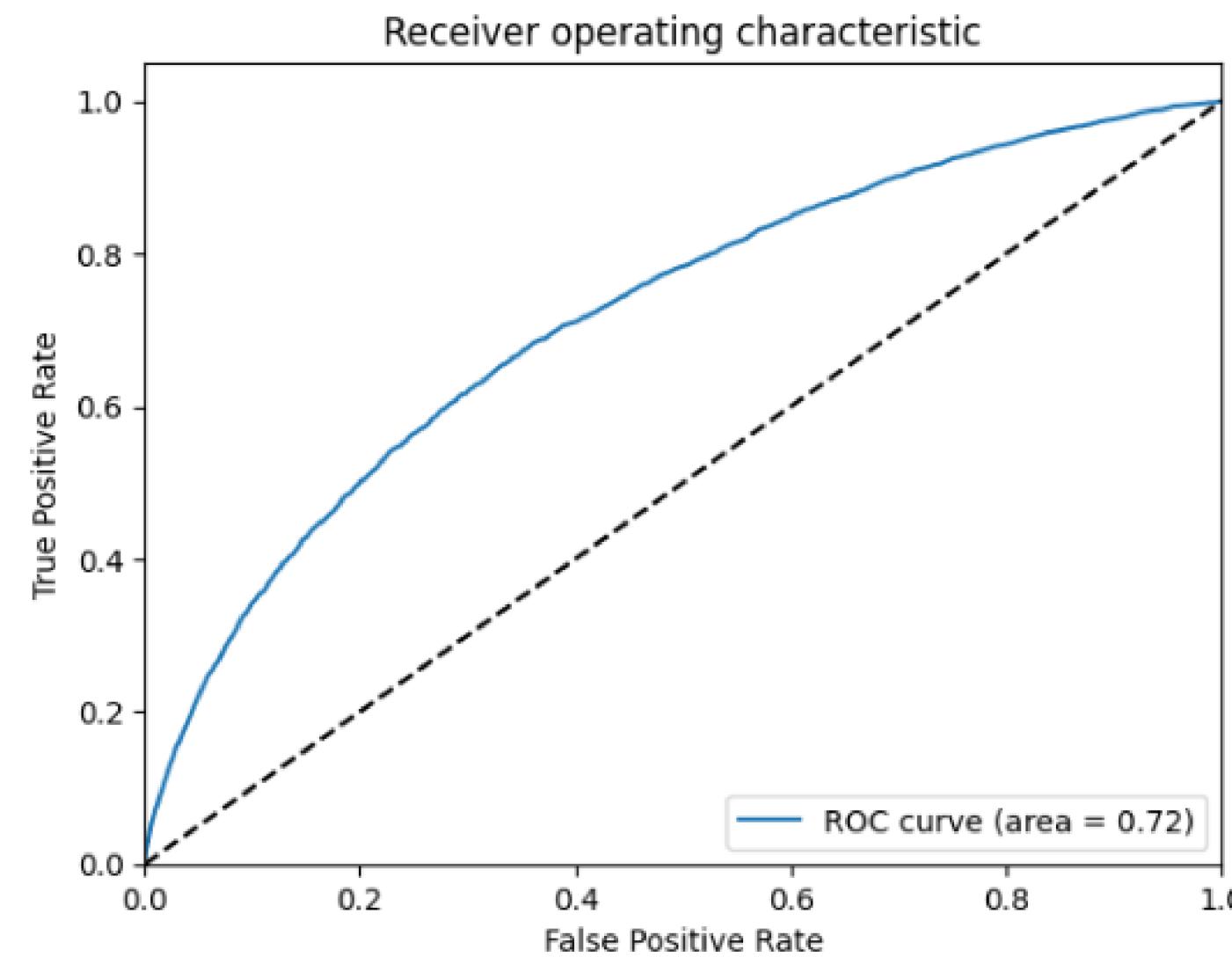


# Partie 3: Test de modèles de Machine Learning

modèle 1

Le Random Forest

Nos Résultats :



Mais :

Accuracy: 0.9124668435013262

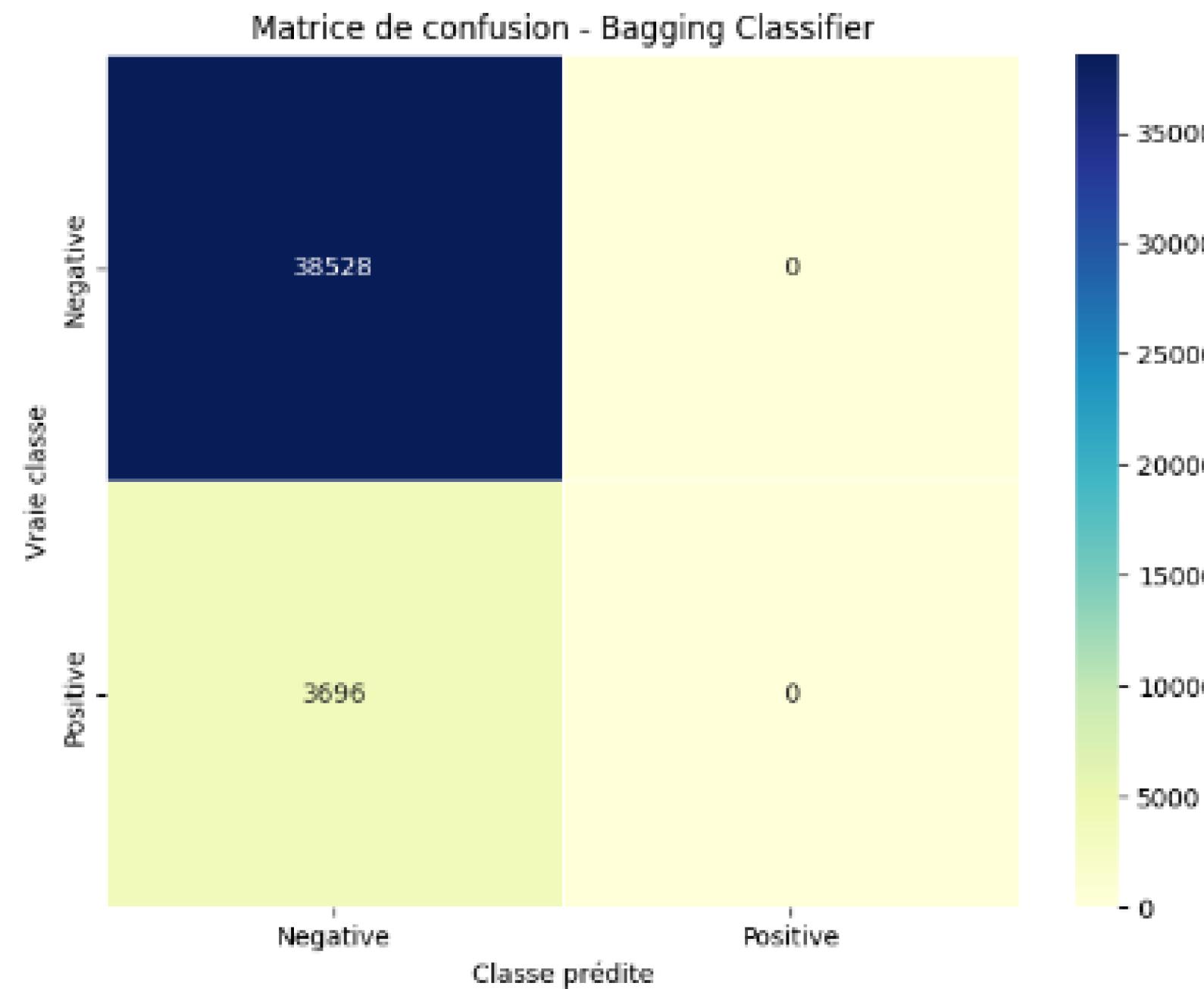


# Partie 3: Test de modèles de Machine Learning

modèle 1

## Le Random Forest

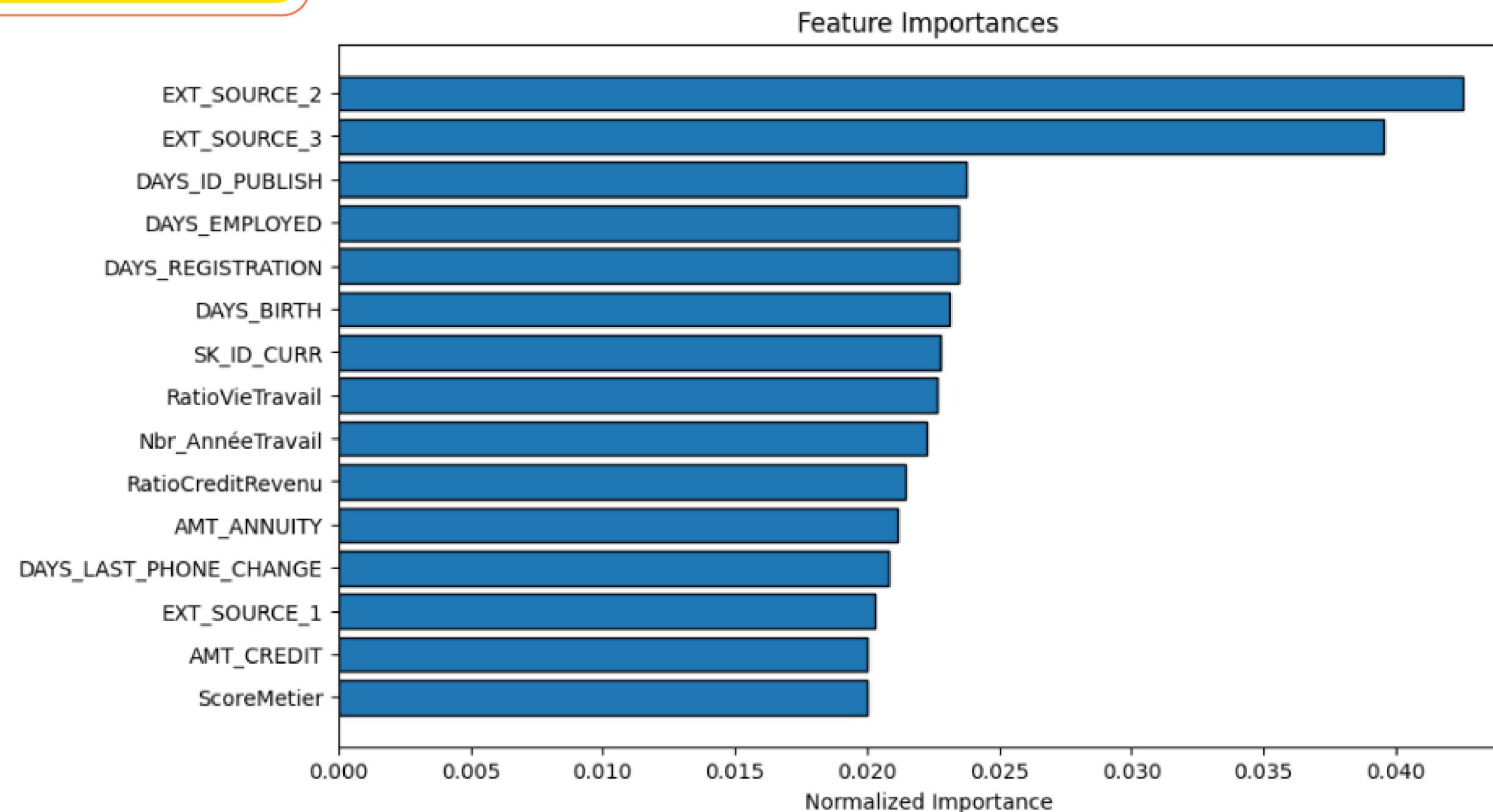
Nos Résultats :



# Partie 3: Test de modèles de Machine Learning

modèle 1

## Le Random Forest



# Test de 3 modèles

modèle 1

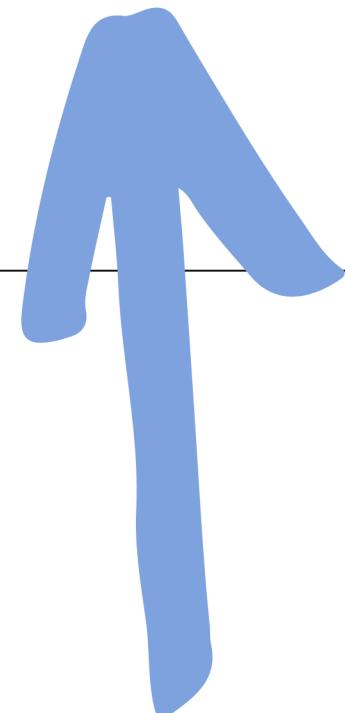
Random Forest

modèle 2

Booster Gradiant

modèle 3

Classification et  
HyperParamètre



# Partie 3: Test de modèles de Machine Learning

## modèle 2

## Booster Gradiant

Le Gradient Boosting est un algorithme d'**apprentissage automatique** qui combine **plusieurs modèles faibles pour former un modèle plus fort**, en se concentrant sur la réduction des erreurs de prédiction de manière itérative.

### Nos Paramètres :

```
# Create the model
model = lgb.LGBMClassifier(n_estimators=10000, objective='binary',
                            class_weight='balanced', learning_rate=0.05,
                            reg_alpha=0.1, reg_lambda=0.1,
                            subsample=0.8, n_jobs=-1, random_state=50)
```

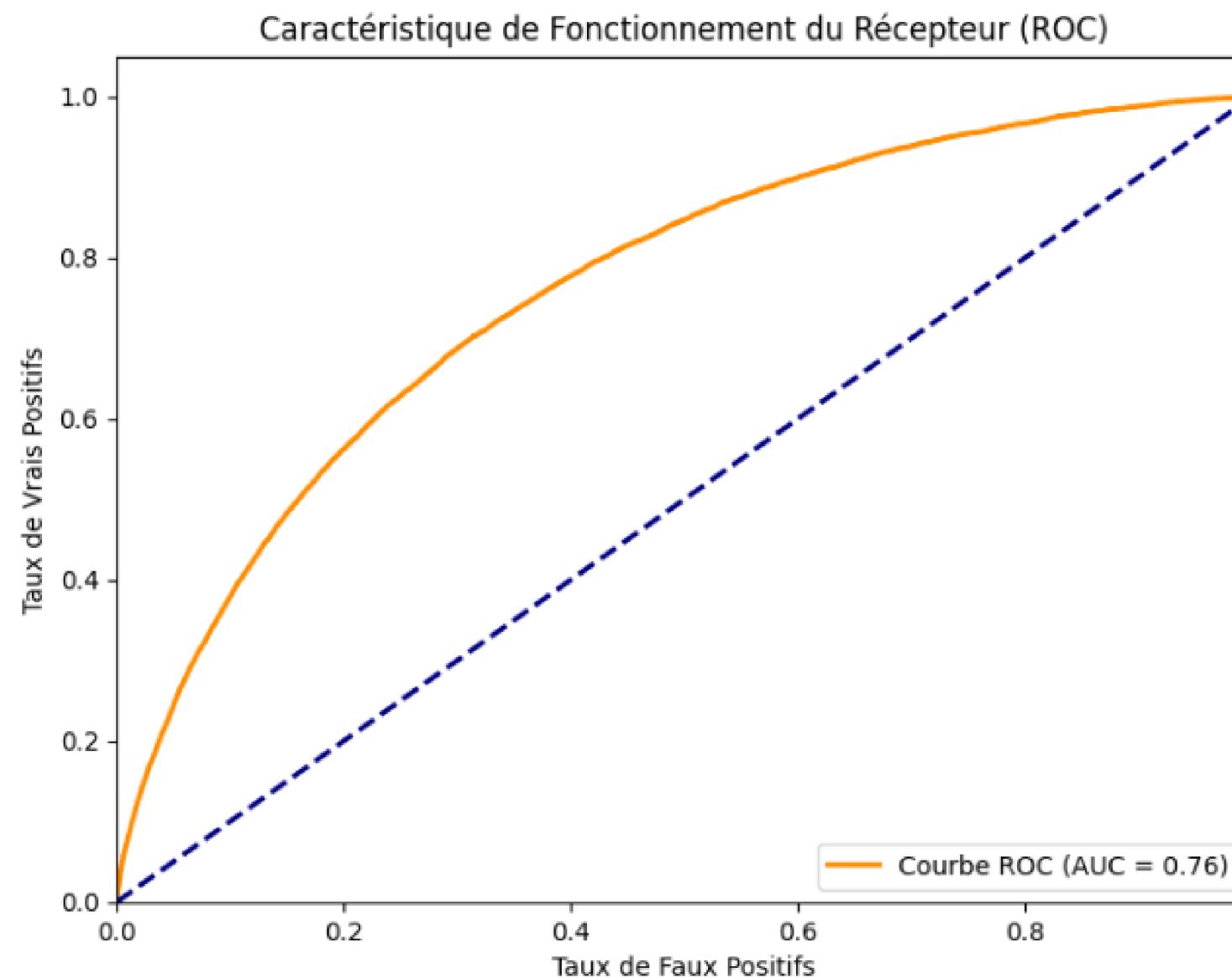


# Partie 3: Test de modèles de Machine Learning

modèle 2

Booster Gradiant

Nos Résultats :



Avec :

Baseline metrics

	fold	train	valid
0	0	0.803611	0.758140
1	1	0.808530	0.760888
2	2	0.812129	0.759065
3	3	0.824901	0.765348
4	4	0.811053	0.760058
5	overall	0.812045	0.760660

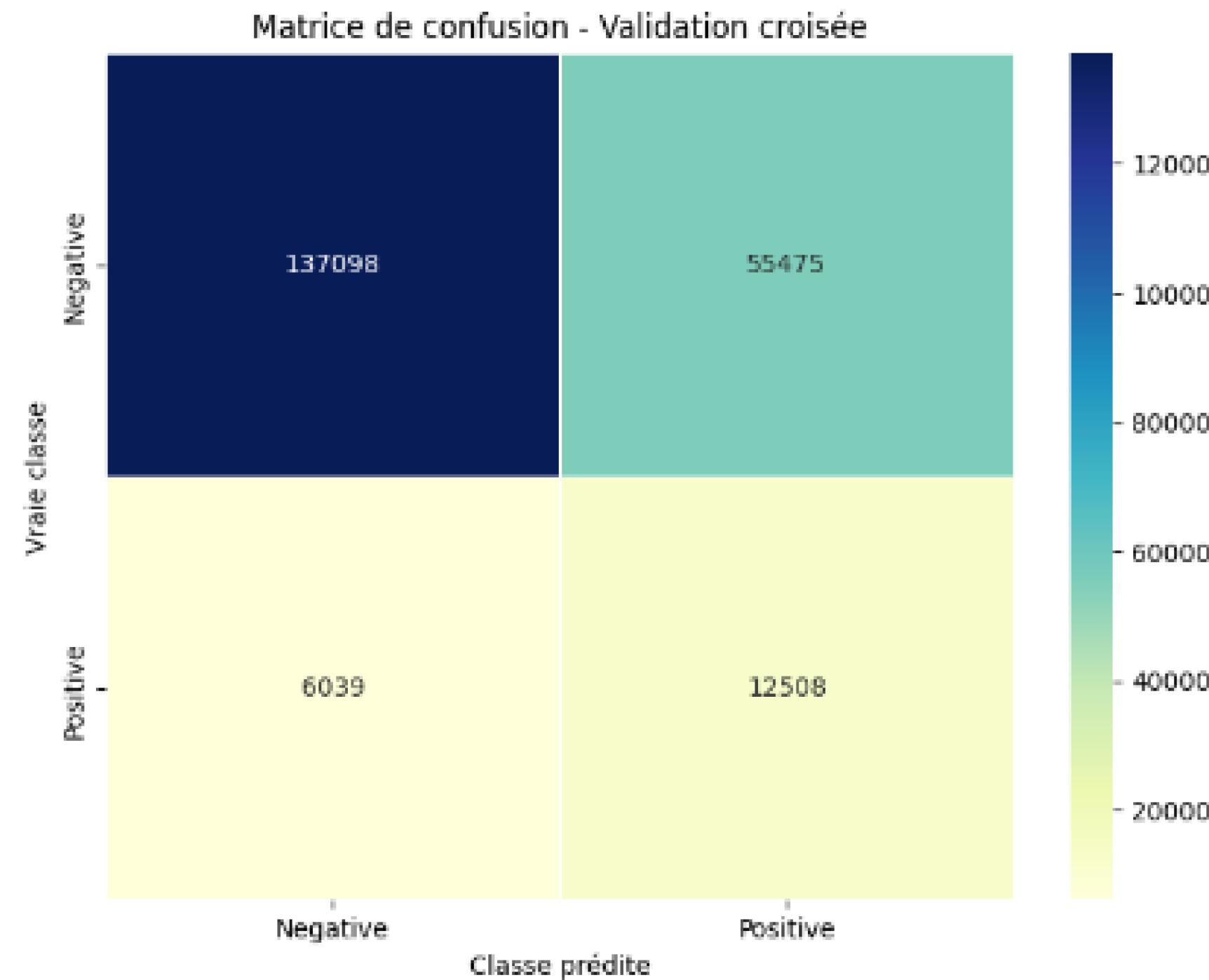


# Partie 3: Test de modèles de Machine Learning

modèle 2

Booster Gradiant

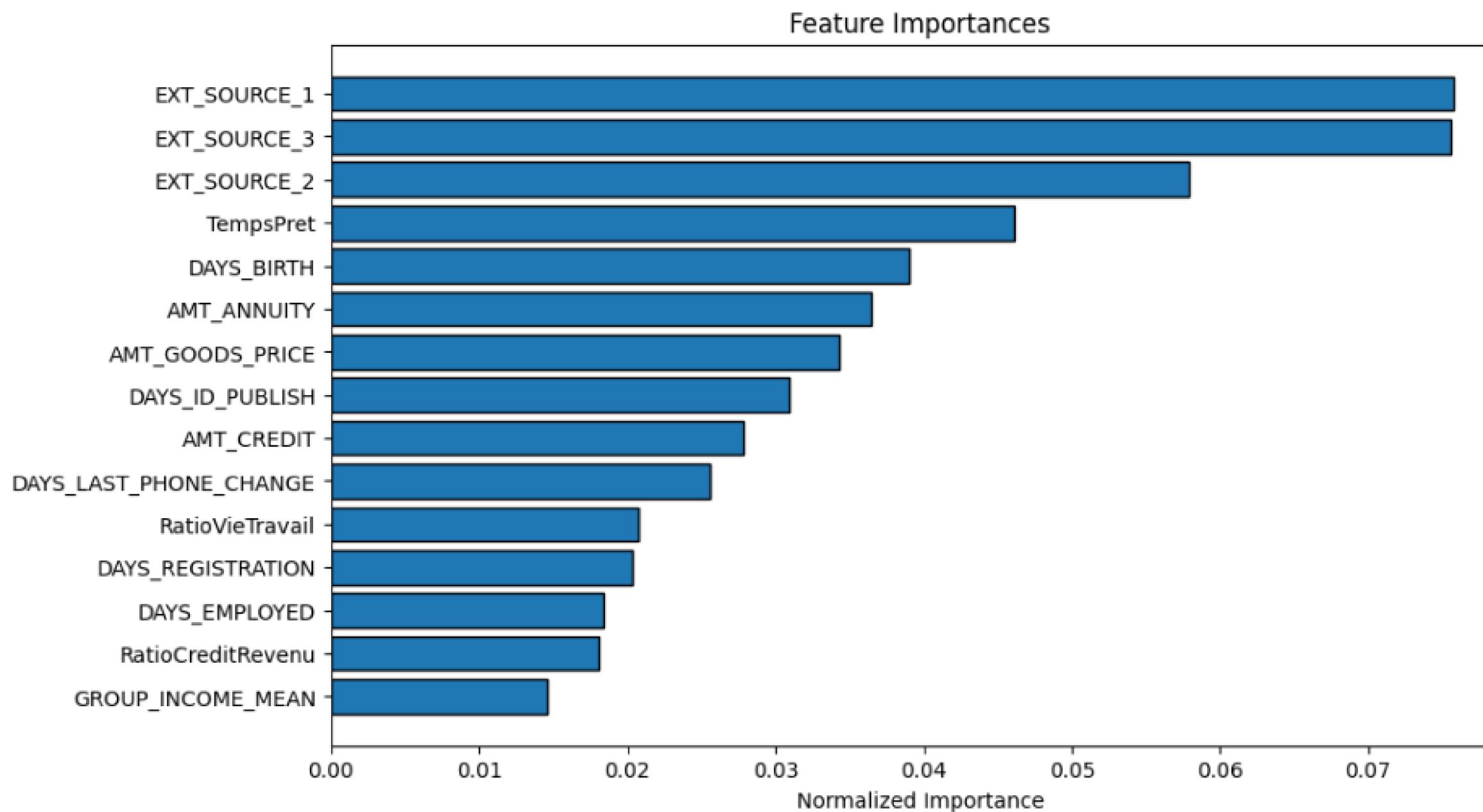
Nos Résultats :



# Partie 3: Test de modèles de Machine Learning

modèle 2

Booster Gradiant



# Test de 3 modèles

modèle 1

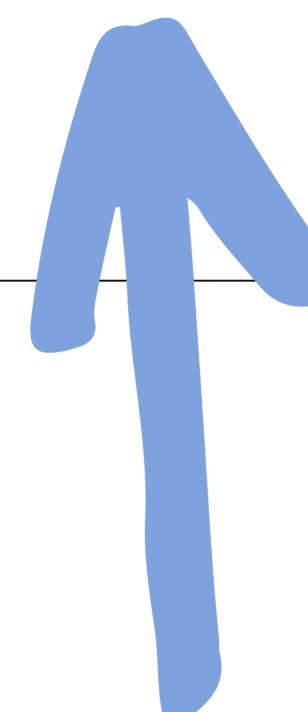
Random Forest

modèle 2

Booster Gradient

modèle 3

Classification et  
HyperParamètre



# Partie 3: Test de modèles de Machine Learning

modèle 3

## Classification et HyperParamètre

Un modèle de **classification** est un algorithme qui prédit des catégories discrètes. Les **hyperparamètres** sont des paramètres **ajustables** avant l'**entraînement** pour optimiser les performances du modèle.

On utilisera :

GradientBoostingClassifier avec SMOTE et GridSearchCV



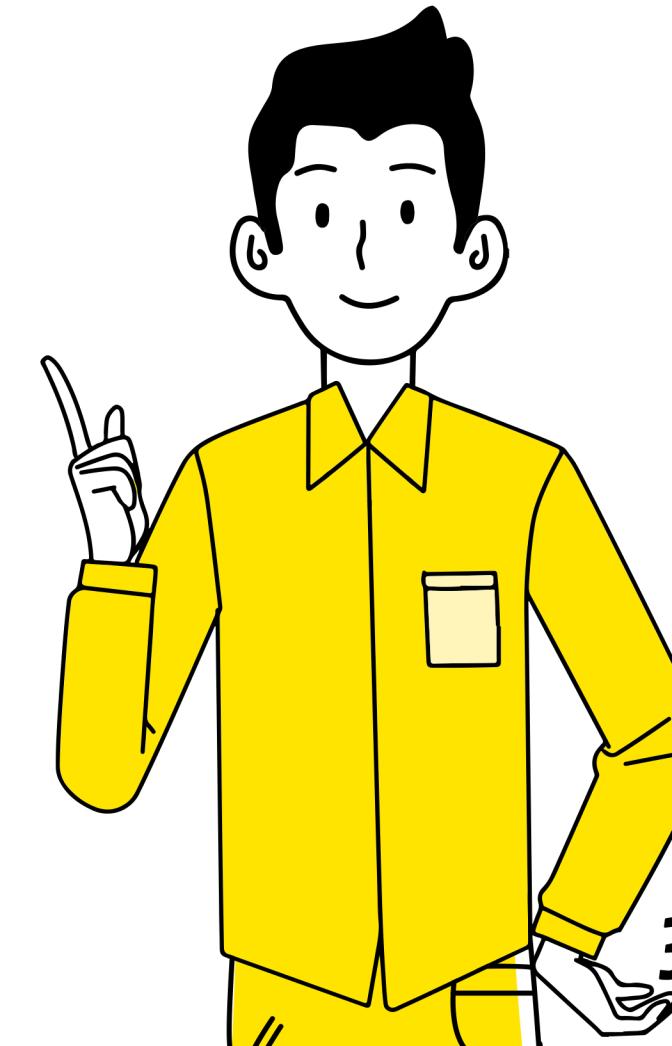
# Partie 3: Test de modèles de Machine Learning

modèle 3

Classification et HyperParamètre



Temps de calcul  
extrêmement long.  
Obligation d'adaptation.



# Partie 3: Test de modèles de Machine Learning

## modèle 3

### Classification et HyperParamètre

#### Nos Paramètres :

```
param_grid = {  
    'gradientboostingclassifier__n_estimators': [100, 120], # Ex  
    'gradientboostingclassifier__learning_rate': [ 0.05, 0.07],  
    'gradientboostingclassifier__max_depth': [3, 4], # Le 4 semble  
  
    'gradientboostingclassifier__min_samples_split': [6, 8, 10],  
    'gradientboostingclassifier__min_samples_leaf': [1, 2], # Telle  
    'gradientboostingclassifier__subsample': [0.9, 0.95], # Test  
}  
  
tance GridSearchCV  
earchCV(estimator=pipeline, param_grid=param_grid, cv=2, scoring='accu
```

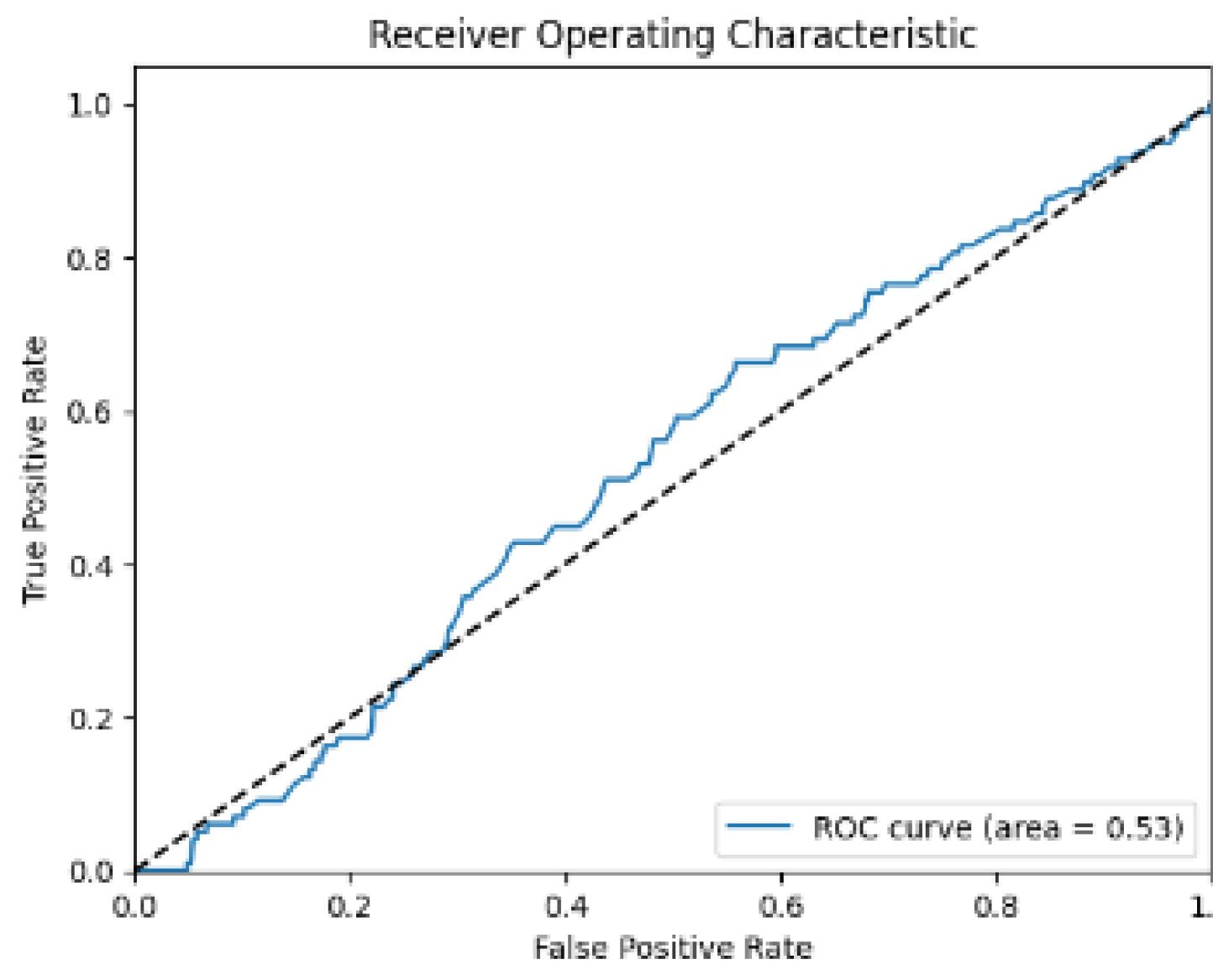


# Partie 3: Test de modèles de Machine Learning

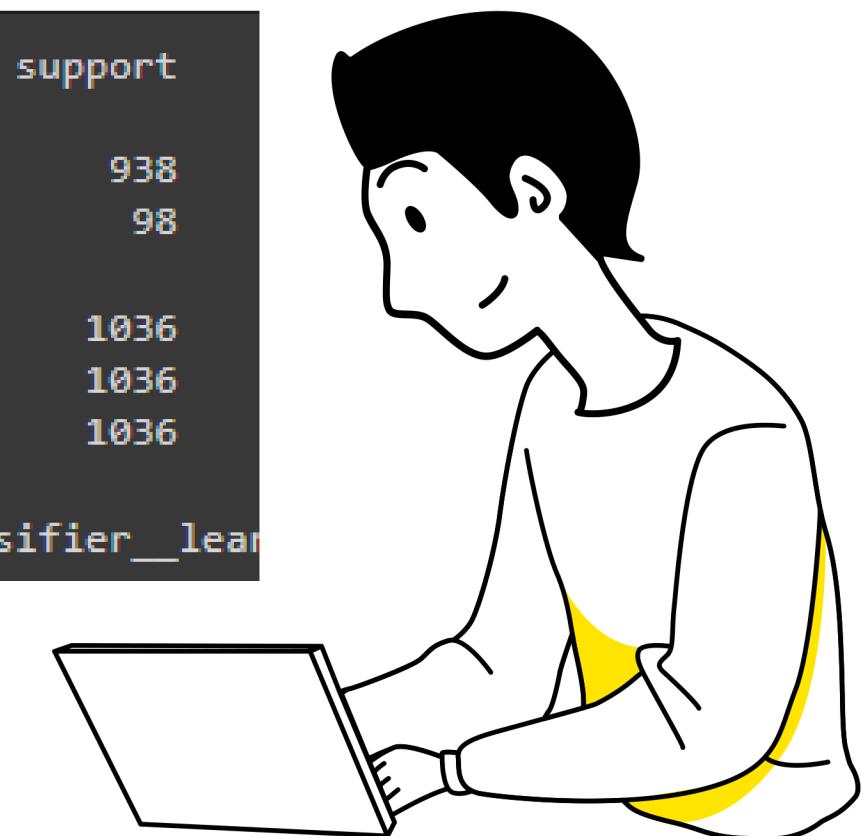
modèle 3

## Classification et HyperParamètre

### Nos Résultats :



	precision	recall	f1-score	support
0	0.91	1.00	0.95	938
1	0.50	0.01	0.02	98
accuracy				
0.70				
macro avg				
0.50				
weighted avg				
0.87				
Meilleurs paramètres: {'gradientboostingclassifier_lear				

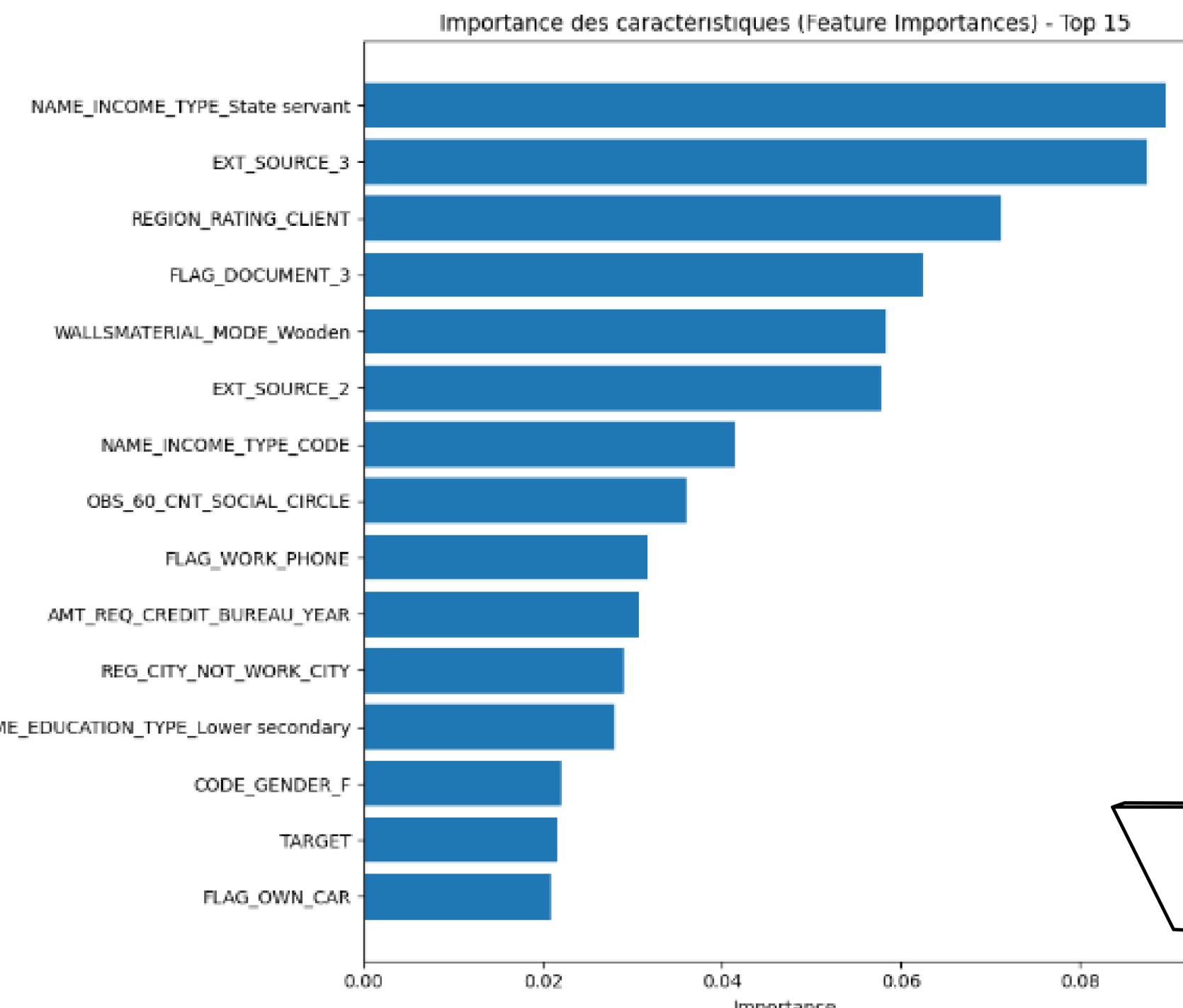


# Partie 3: Test de modèles de Machine Learning

modèle 3

## Classification et HyperParamètre

Nos Résultats :



## **Partie 4: Conclusion**

## Partie 4: Conclusion

Pouvons-nous utiliser un modèle de scoring pour anticiper les faux positifs et les faux négatifs ?

Oui mais

- Trouver le bon modèle. (ici le 2)
- Un modèle n'est jamais parfait.
- Facteur humain.
- Réajuster avec notre politique.



# Merci!

