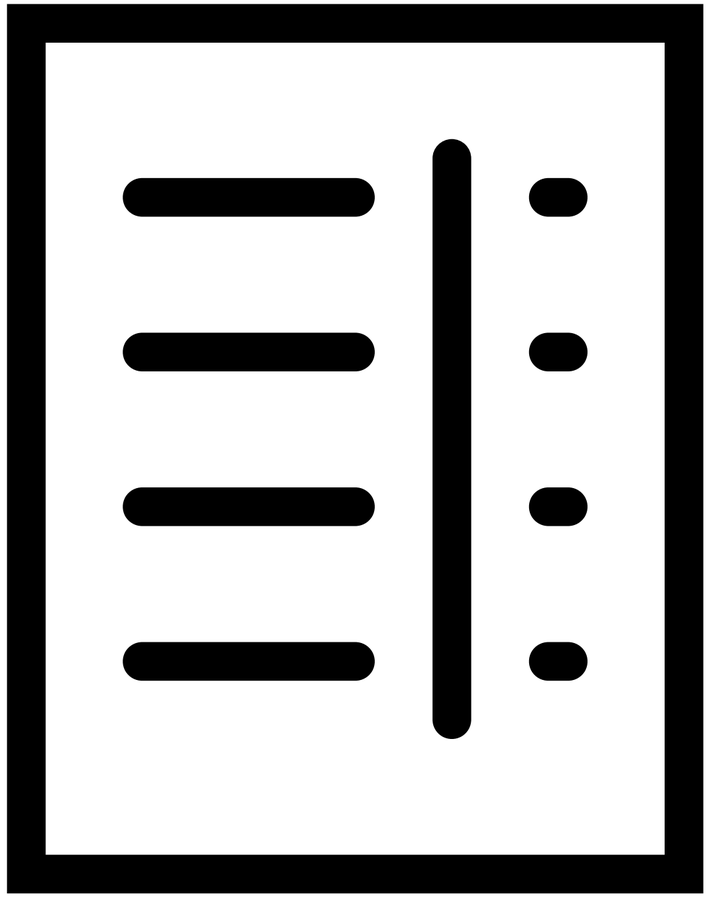
LA PRÉDICTION DES FAILLITES D'ENTREPRISES

Sommaire

1. Contexte
2. EDA
3. Preprocessing
4. Modèles
5. Conclusion



Présentation du Projet

Développeur dans une entreprise qui travaille dans le domaine de la FinTech.

Notre client est une société d'investissement qui souhaite détecter les faillites d'entreprises, ce qui leur permet de mieux comprendre et gérer les risques.

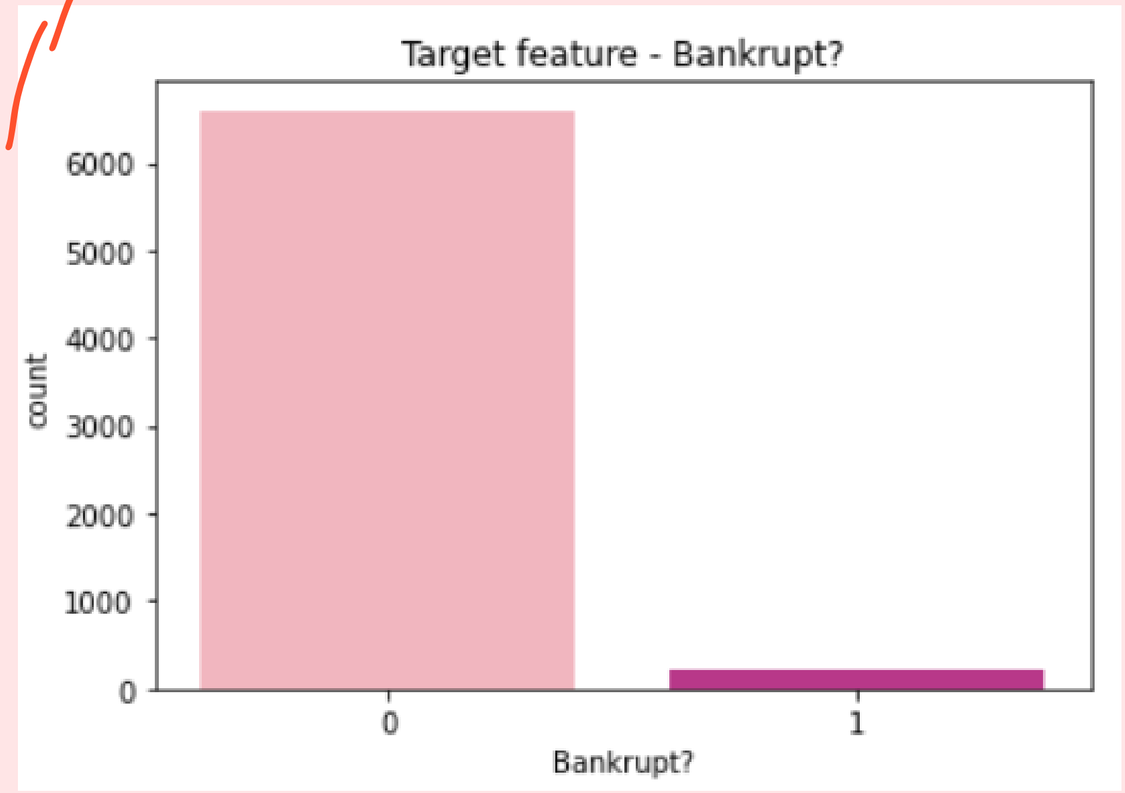
Nous avons créer un modèle de Machine Learning pour détecter la probabilité de faire faillite en utilisant plusieurs méthodes

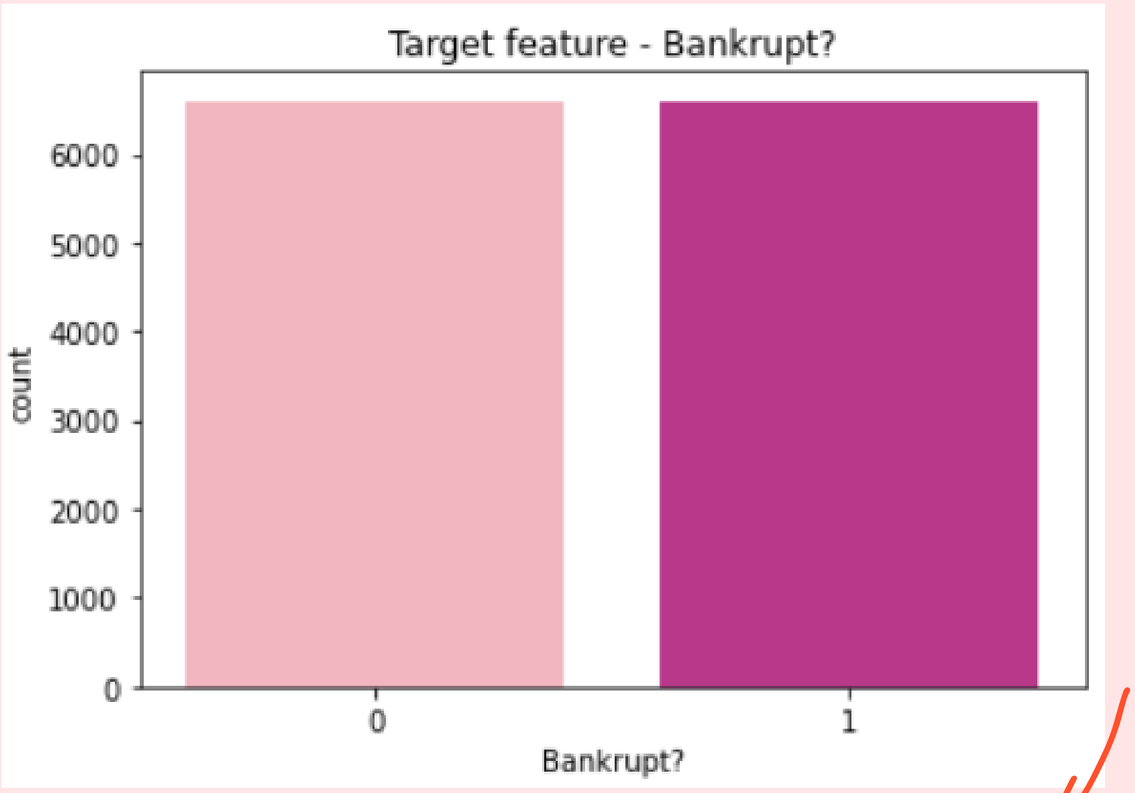
EDA

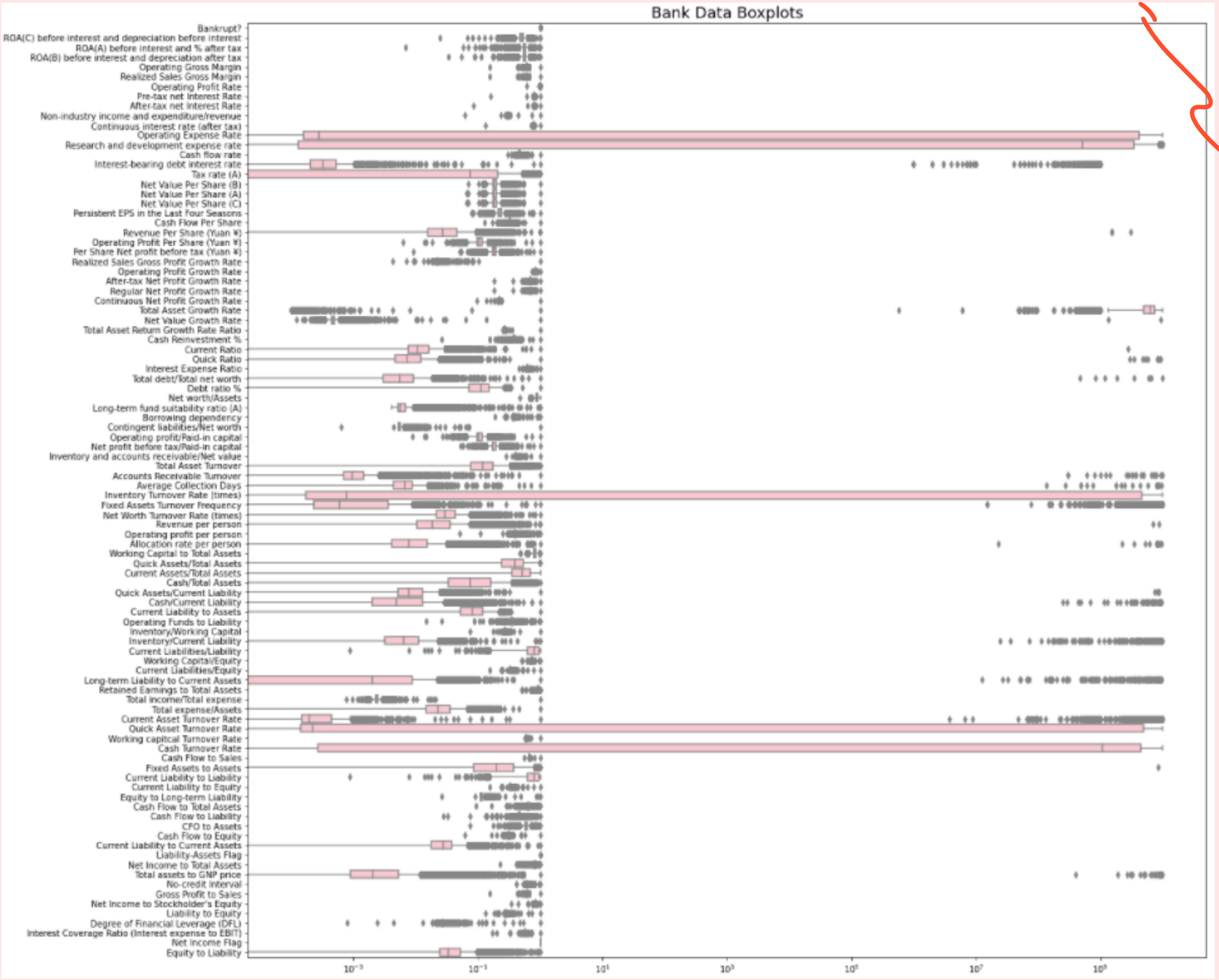
Présentation du Dataset

Lignes	*Colonnes*	*Doublons*	*Valeurs manquantes*
6819	**96**	**0**	**0**

Sur-échantillonnage de la target à l'aide de SMOTE







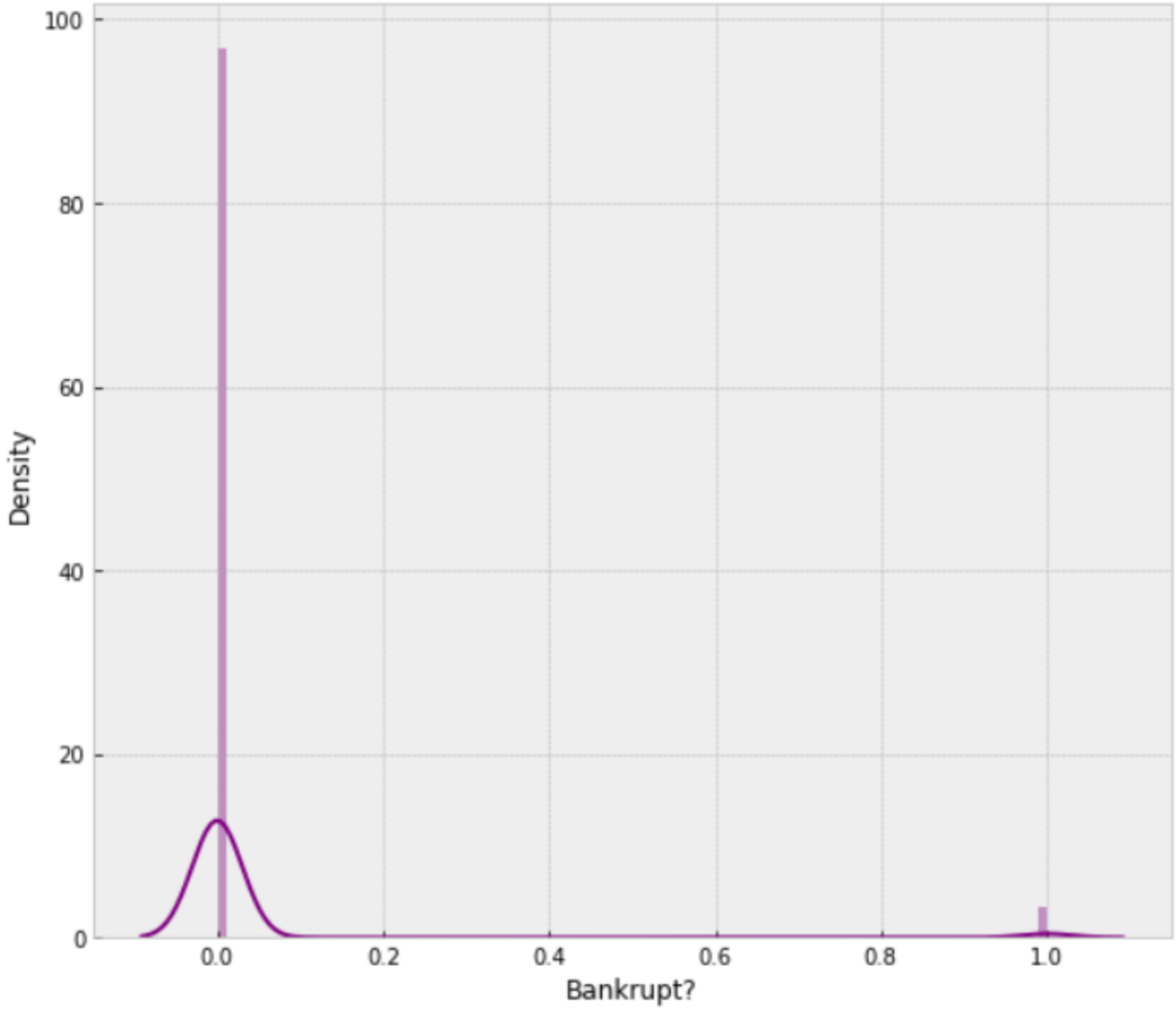
Outliers

Forte présence d'outlier parmis la plupart des features

Corrélations

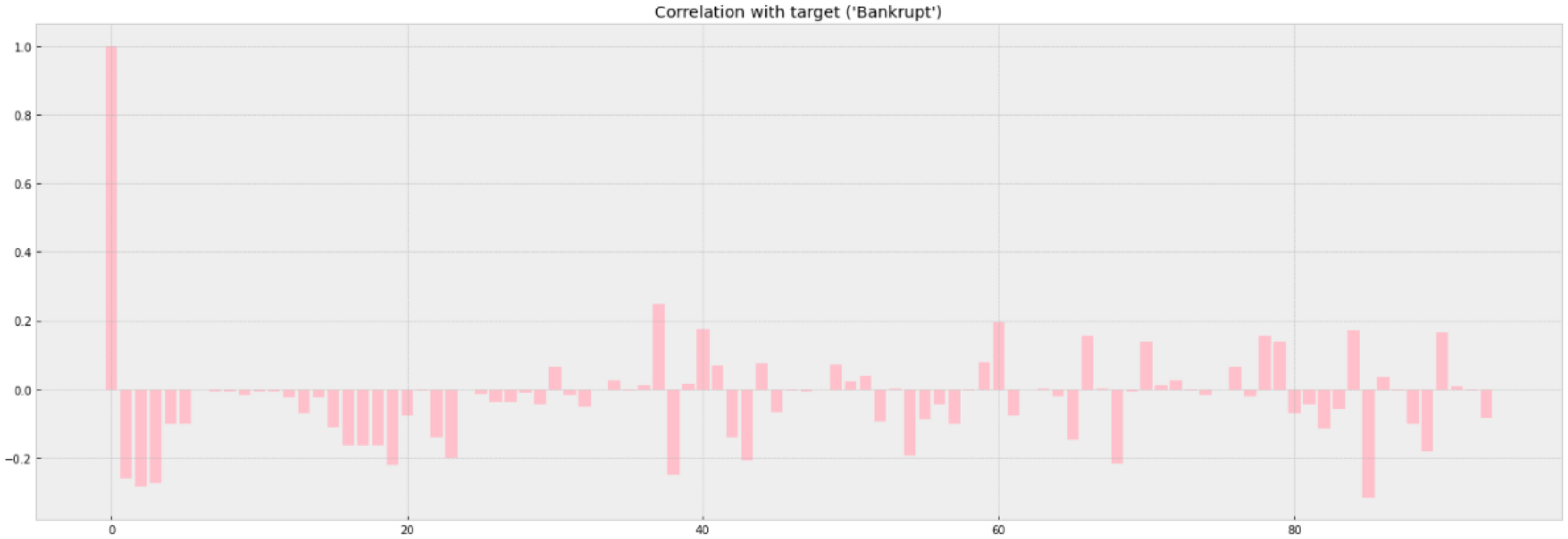
Analyse de la target

- Notre cible est énormément déséquilibrée
- Seulement 3% de 1 (faillites)
- Contre 97% de 0 (pas de faillites)



Corrélation avec la target

Plusieurs features sont corrélées positivement et négativement avec notre cible.



PREPROCESSING

Features Selection

```
Il y a 22  variables corrélés à 0.15% avec la cible:
Bankrupt?                                                    1.000000
Debt ratio %                                                 0.250161
Current Liability to Assets                                  0.194494
Borrowing dependency                                         0.176543
Current Liability to Current Assets                          0.171306
Liability to Equity                                          0.166812
Current Liabilities/Equity                                   0.153828
Current Liability to Equity                                  0.153828
Net Value Per Share (C)                                     -0.164784
Net Value Per Share (B)                                     -0.165399
Net Value Per Share (A)                                     -0.165465
Net Income to Stockholder's Equity                          -0.180987
Working Capital to Total Assets                             -0.193083
Per Share Net profit before tax (Yuan ¥)                    -0.201395
Net profit before tax/Paid-in capital                       -0.207857
Retained Earnings to Total Assets                           -0.217779
Persistent EPS in the Last Four Seasons                     -0.219560
Net worth/Assets                                            -0.250161
ROA(C) before interest and depreciation before interest     -0.260807
ROA(B) before interest and depreciation after tax           -0.273051
ROA(A) before interest and % after tax                      -0.282941
Net Income to Total Assets                                  -0.315457
Name: Bankrupt?, dtype: float64
```

Préprocessing et Techniques

- **Split des données avec StratifiedKFold**
- **Scaling avec Robust Scaler**
- **Balancement sur le X-train et Y-train avec la méthode de SMOTE**

MODÈLES

Itération des différents modèles

```
Fitting 5 folds for each of 18 candidates, totalling 90 fits
0.9030497611157557
{'model__learning_rate': 1, 'model__n_estimators': 70}
              precision    recall  f1-score   support

           0       0.99      0.93      0.96      1319
           1       0.26      0.73      0.39        44

    accuracy                           0.93      1363
   macro avg       0.63      0.83      0.67      1363
weighted avg       0.97      0.93      0.94      1363

[[1229   90]
 [  12   32]]
```

Gradient Boosting

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
0.9222927851250491
{'model__n_neighbors': 2, 'model__weights': 'uniform'}
              precision    recall  f1-score   support

           0       0.98      0.95      0.96      1319
           1       0.18      0.32      0.23        44

    accuracy                           0.93      1363
   macro avg       0.58      0.63      0.60      1363
weighted avg       0.95      0.93      0.94      1363

[[1254   65]
 [  30   14]]
```

KNN

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
0.9039671907682907
{'model__max_depth': None, 'model__splitter': 'random'}
              precision    recall  f1-score   support

           0       0.98      0.92      0.95      1319
           1       0.16      0.45      0.24        44

    accuracy                           0.91      1363
   macro avg       0.57      0.69      0.60      1363
weighted avg       0.95      0.91      0.93      1363

[[1217  102]
 [  24   20]]
```

Decision Tree

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
0.9079968305449515
{'model__class_weight': 'balanced_subsample', 'model__max_depth': 10, 'model__n_estimators': 200}
              precision    recall  f1-score   support

           0       0.99      0.94      0.97      1319
           1       0.31      0.84      0.45        44

    accuracy                           0.93      1363
   macro avg       0.65      0.89      0.71      1363
weighted avg       0.97      0.93      0.95      1363

[[1237   82]
 [   7   37]]
```

Random Forest

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
0.9202761186262561
{'model__base_estimator': RandomForestClassifier(class_weight='balanced_subsample', max_depth=10,
                       n_estimators=250), 'model__n_estimators': 150}
              precision    recall  f1-score   support

           0       0.99      0.95      0.97      1319
           1       0.34      0.75      0.47        44

    accuracy                           0.95      1363
   macro avg       0.67      0.85      0.72      1363
weighted avg       0.97      0.95      0.96      1363

[[1256   63]
 [  11   33]]
```

AdaBoost + RandomForest

```
Fitting 5 folds for each of 3 candidates, totalling 15 fits
0.7397523191748674
{'model__learning_rate': 'adaptive', 'model__learning_rate_init': 0.1, 'model__max_iter': 400}
              precision    recall  f1-score   support

           0       1.00      0.74      0.85      1319
           1       0.10      0.91      0.19        44

    accuracy                           0.75      1363
   macro avg       0.55      0.82      0.52      1363
weighted avg       0.97      0.75      0.83      1363

[[976 343]
 [  4  40]]
```

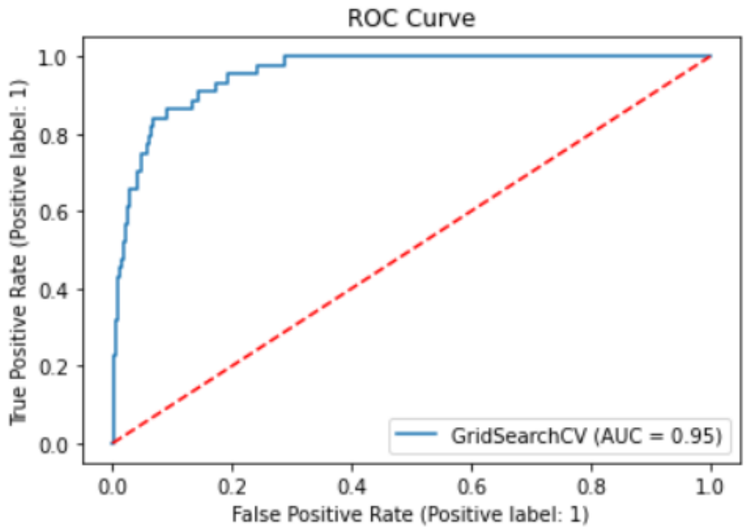
Multi Layer Perceptron

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
0.9266904040047944
{'model__final_estimator': LogisticRegression(class_weight='balanced')}
              precision    recall  f1-score   support

           0       0.99      0.95      0.97      1319
           1       0.34      0.82      0.48        44

    accuracy                           0.94      1363
   macro avg       0.67      0.88      0.72      1363
weighted avg       0.97      0.94      0.95      1363

[[1249   70]
 [   8   36]]
```

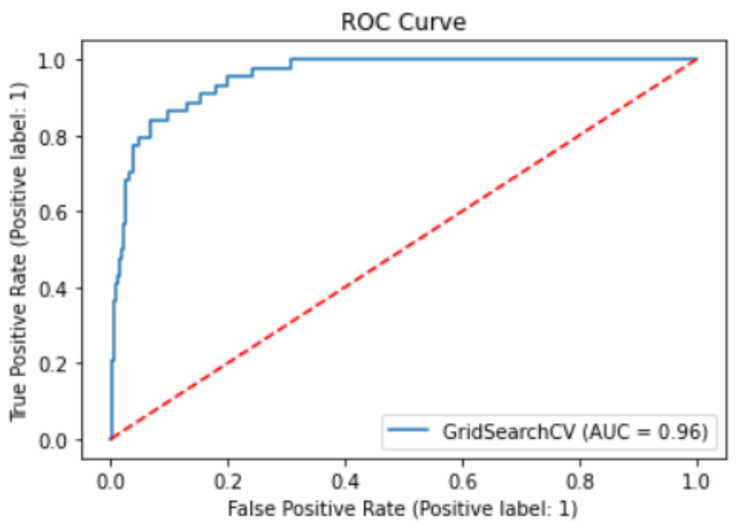
Stacking

Evaluation des meilleurs modèles

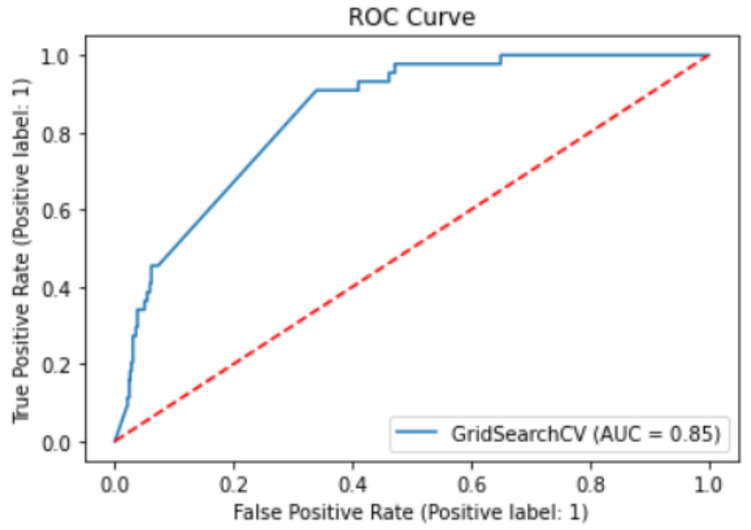
RandomForest



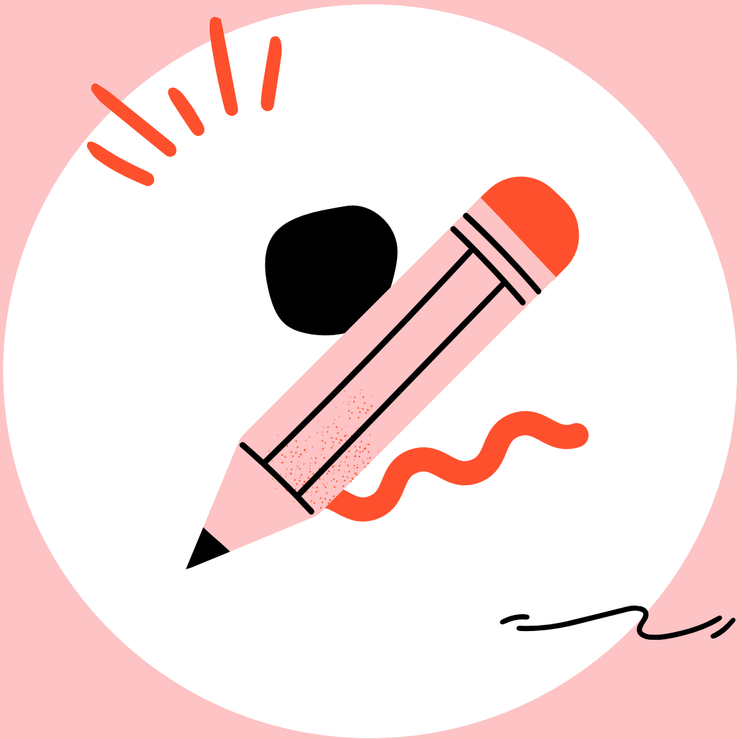
Stacking



Multi Layer Perceptron



Conclusion



- Le déséquilibre de la cible entraîne une mauvaise prédiction du model.
- Grâce à certaines techniques de balancement il est possible de palier à ce problème.
- La plupart des modèles nous permette d'obtenir une bonne reconnaissance de la classe en infériorité numérique.
- Mais cela ce fait au détriment de la reconnaissance de la classe dominante.