Python Project : bankruptcy prediction of Polish companies

Noémie Guisnel, Louise Ferreira, Thibaut Ermenault

DIA4

Summary

- Presentation of the dataset and of the problem
- Data preprocessing choices
- Data visualisation
- Machine learning

Dataset : Polish companies bankruptcy data

- 5 files .arff for 5 years :

1year.arff (7027 rows, 65 columns), 2year.arff (10173 rows, 65 columns), 3year.arff (10503 rows, 65 columns), 4year.arff (9792 rows, 65 columns), 5year.arff (5910 rows, 65 columns)

- Each dataset contains financial rates and corresponding class label that indicates if the company has been bankrupted (1 if bankrupted, 0 if not).

Here are the 64 columns present in each dataset :

- X1 net profit / total assets
- X2 total liabilities / total assets
- X3 working capital / total assets
- X4 current assets / short-term liabilities
- X5 [(cash + short-term securities + receivables - short-term liabilities) /
- (operating expenses - depreciation)] * 365
- X6 retained earnings / total assets
- X7 EBIT / total assets
- X8 book value of equity / total liabilities
- X9 sales / total assets
- X10 equity / total assets
- X11 (gross profit + extraordinary items + financial expenses) / total assets
- X12 gross profit / short-term liabilities
- X13 (gross profit + depreciation) / sales
- X14 (gross profit + interest) / total assets
- X15 (total liabilities * 365) / (gross profit + depreciation)
- X16 (gross profit + depreciation) / total liabilities
- X17 total assets / total liabilities
- X18 gross profit / total assets
- X19 gross profit / sales
- X20 (inventory * 365) / sales
- X21 sales (n) / sales (n-1)
- X22 profit on operating activities / total assets
- X23 net profit / sales
- X24 gross profit (in 3 years) / total assets
- X25 (equity - share capital) / total assets
- X26 (net profit + depreciation) / total liabilities
- X27 profit on operating activities / financial expenses
- X28 working capital / fixed assets
- X29 logarithm of total assets
- X30 (total liabilities - cash) / sales
- X31 (gross profit + interest) / sales
- X32 (current liabilities * 365) / cost of products sold
- X33 operating expenses / short-term liabilities
- X34 operating expenses / total liabilities
- X35 profit on sales / total assets
- X36 total sales / total assets
- X37 (current assets - inventories) / long-term liabilities
- X38 constant capital / total assets
- X39 profit on sales / sales
- X40 (current assets - inventory - receivables) / short-term liabilities
- X41 total liabilities / ((profit on operating activities + depreciation) * (12/365))
- X42 profit on operating activities / sales
- X43 rotation receivables + inventory turnover in days
- X44 (receivables * 365) / sales
- X45 net profit / inventory
- X46 (current assets - inventory) / short-term liabilities
- X47 (inventory * 365) / cost of products sold
- X48 EBITDA (profit on operating activities - depreciation) / total assets
- X49 EBITDA (profit on operating activities - depreciation) / sales
- X50 current assets / total liabilities
- X51 short-term liabilities / total assets
- X52 (short-term liabilities * 365) / cost of products sold)
- X53 equity / fixed assets
- X54 constant capital / fixed assets
- X55 working capital
- X56 (sales - cost of products sold) / sales
- X57 (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
- X58 total costs /total sales
- X59 long-term liabilities / equity
- X60 sales / inventory
- X61 sales / receivables
- X62 (short-term liabilities *365) / sales
- X63 sales / short-term liabilities
- X64 sales / fixed assets
- X65 class

Question : is it possible to predict if a company has been bankrupted based on its financial rates ?

We need to understand the important financial terms of the dataset to do the machine learning afterwards :

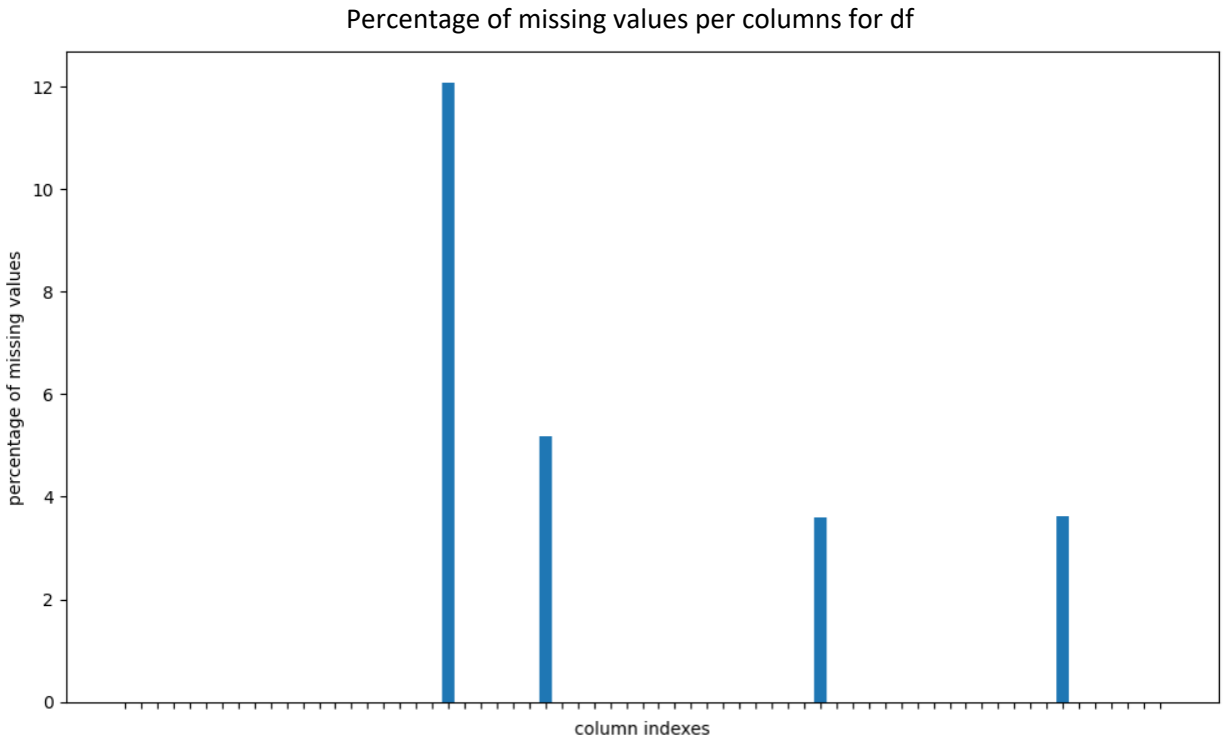
- Assets (company's entire heritage. Examples : buildings, equipment, receivables, patents filed, etc.) are opposed to indebted (means of the company to finance its assets. It consists of equity (fixed liabilities) and liabilities (current liabilities))
- EBIT : earnings before interest and taxes
- EBITDA : Earnings Before Interest, Taxes, Depreciation, and Amortization

Data cleaning and imputation

After an initial cleaning on all datasets, we gathered the 5 datasets in 1 dataset (df_tot), and added a column ‘Year’.

df_tot has still missing values and extreme values (cleaning and imputation required).

We removed the column named ‘(current assets - inventories) / long-term liabilities’, which had about 40 % of missing values.



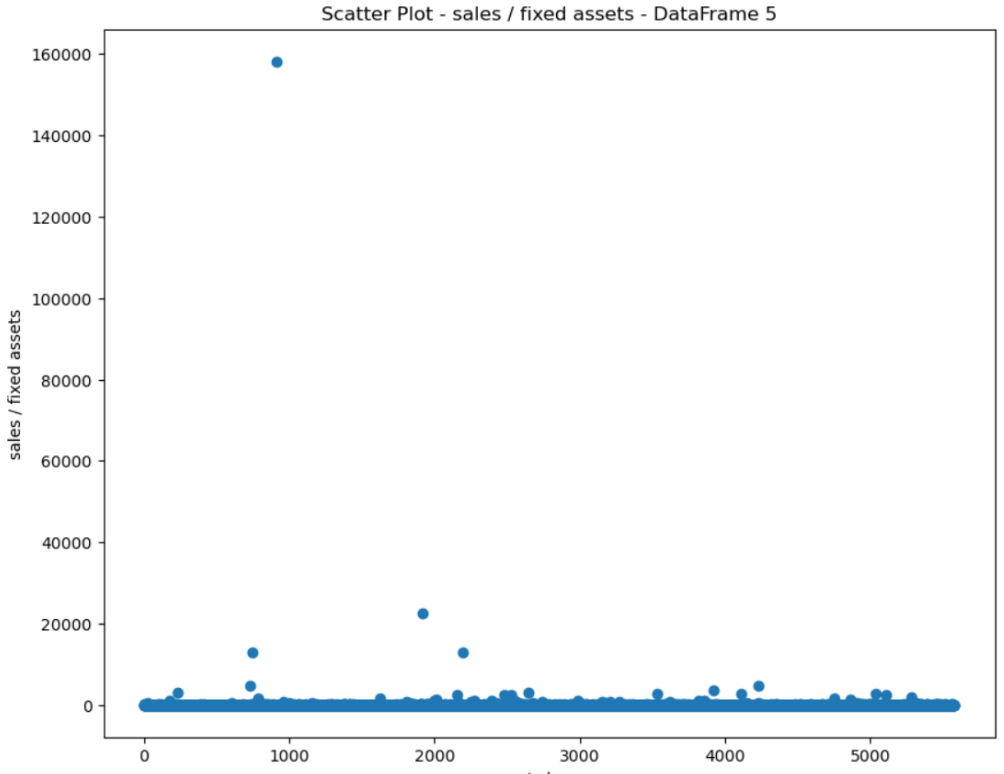
Data cleaning and imputation

Mean method for the column ‘sales (n) / sales (n-1)’ (= variation of the sales from a period to another) : 12 % missing values

Median method for ‘profit on operating activities / financial expenses’ : 5%

KNN method for ‘net profit / inventory’ and ‘sales/inventory’ : 3%

We used the method of the quantiles with coefficients to eliminate the outliers.
Q1 : 25%, Q3 : 75%
Low_bound = Q1 – coefficient * (Q3 – Q1)
High_bound = Q3 + coefficient * (Q3 – Q1)
We remove everything that is not in the interval [Low_bound, High_bound]

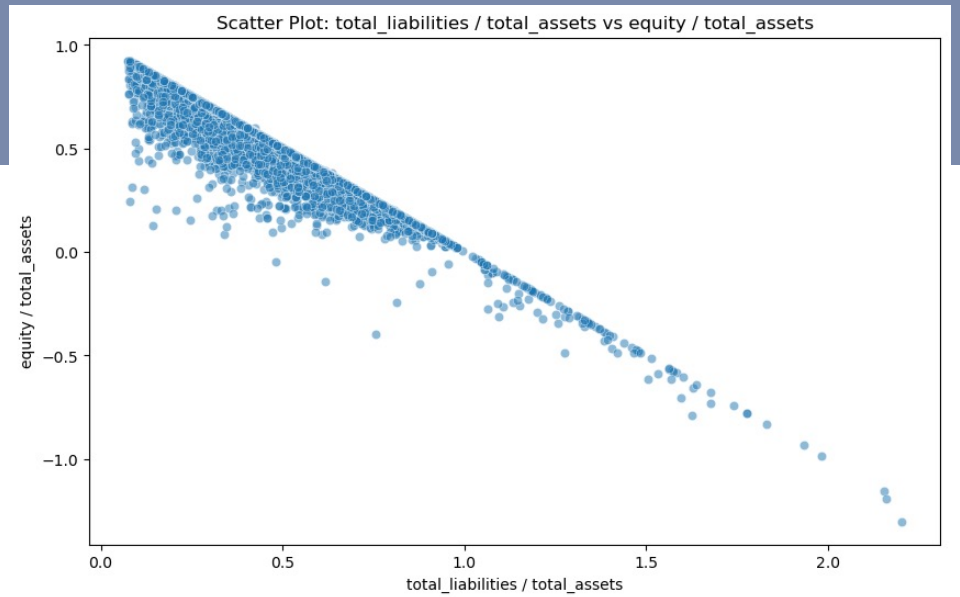


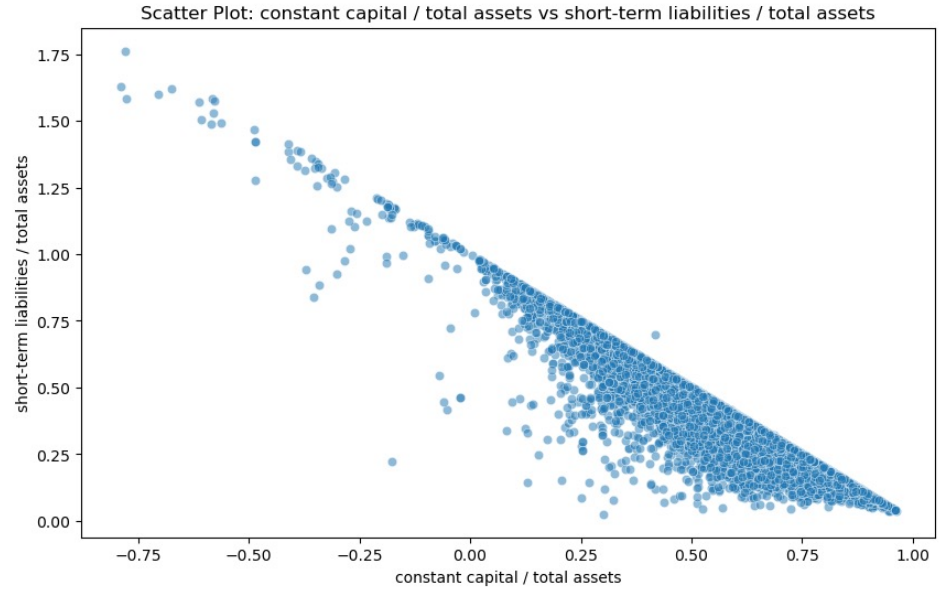
Data visualisation

Thanks to the correlation matrix, we knew which variables are the most positively or negatively correlated :

5 most correlated positively variables	5 most correlated negatively variables
(short-term liabilities * 365) / cost of products sold) and (current liabilities * 365) / cost of products sold : 0.99	equity / total_assets and total_liabilities / total_assets: -0.97
total assets / total liabilities and book_value_of_equity / total_liabilities : 0.99	short-term liabilities / total assets and constant capital / total assets: -0.94
EBIT / total_assets and net_profit / total_assets : 0.99	total costs /total sales and (sales - cost of products sold) / sales: -0.94
(gross profit + interest) / total assets and net_profit / total_assets : 0.98	constant capital / total assets and total_liabilities / total_assets: -0.82
gross profit / total assets and net_profit / total_assets : 0.98	short-term liabilities / total assets and equity / total_assets: -0.81

Examples of correlation





Indeed there is a high correlation between net profit and EBIT, because both concern earnings and profits. The only difference is that the EBIT has still the interests and taxes.

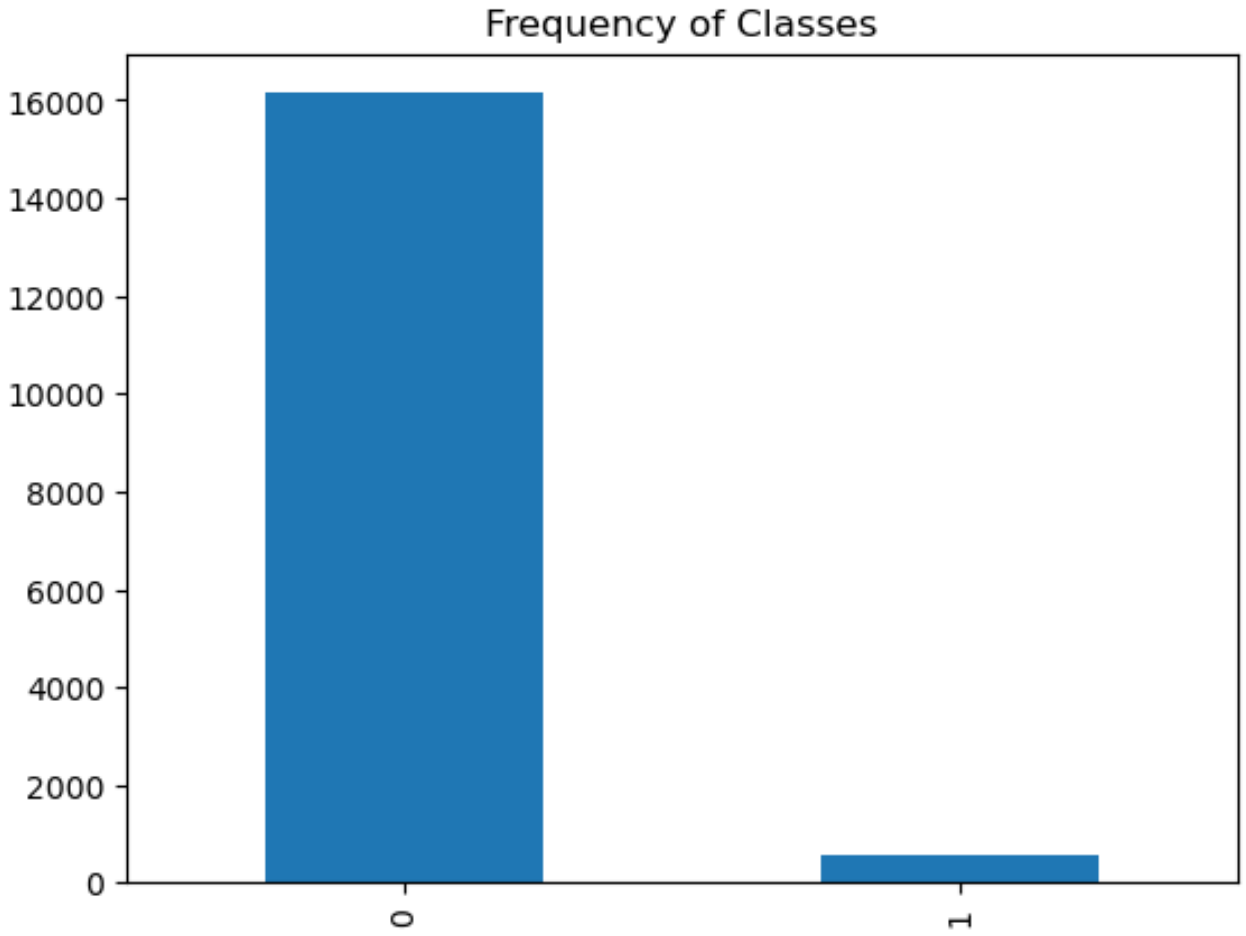
Correlation of the variables to the variable ‘class’

We see that for the companies that have been bankrupted, the net profit /sales and the gross profit /sales were a lot less than for the companies that have not been bankrupted. That is logic because the more a company has profit, the less probability there are that it will bankrupt.

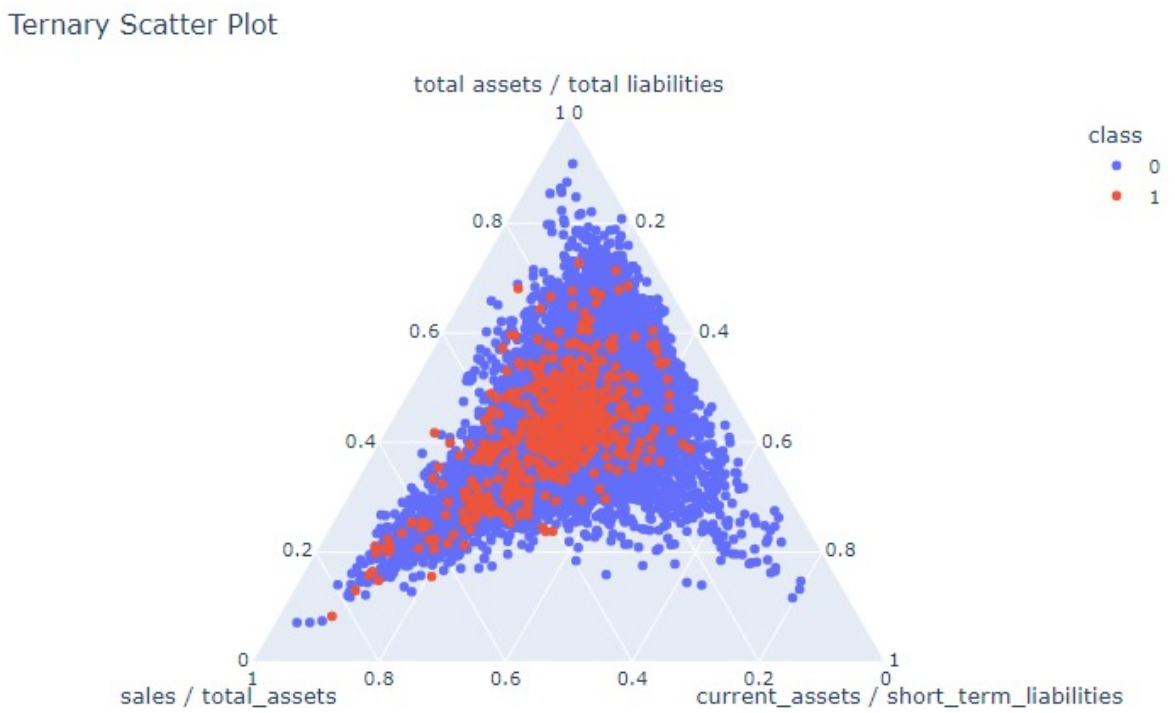
However, the correlations between class and the other variables are very low (between 0.1 and -0.1).

Indeed, there are about 16000 times more companies that did not bankrupted comparing to those who bankrupted.

So it makes it difficult to define some correlations about the variable class and with other variables.



Other visualisations



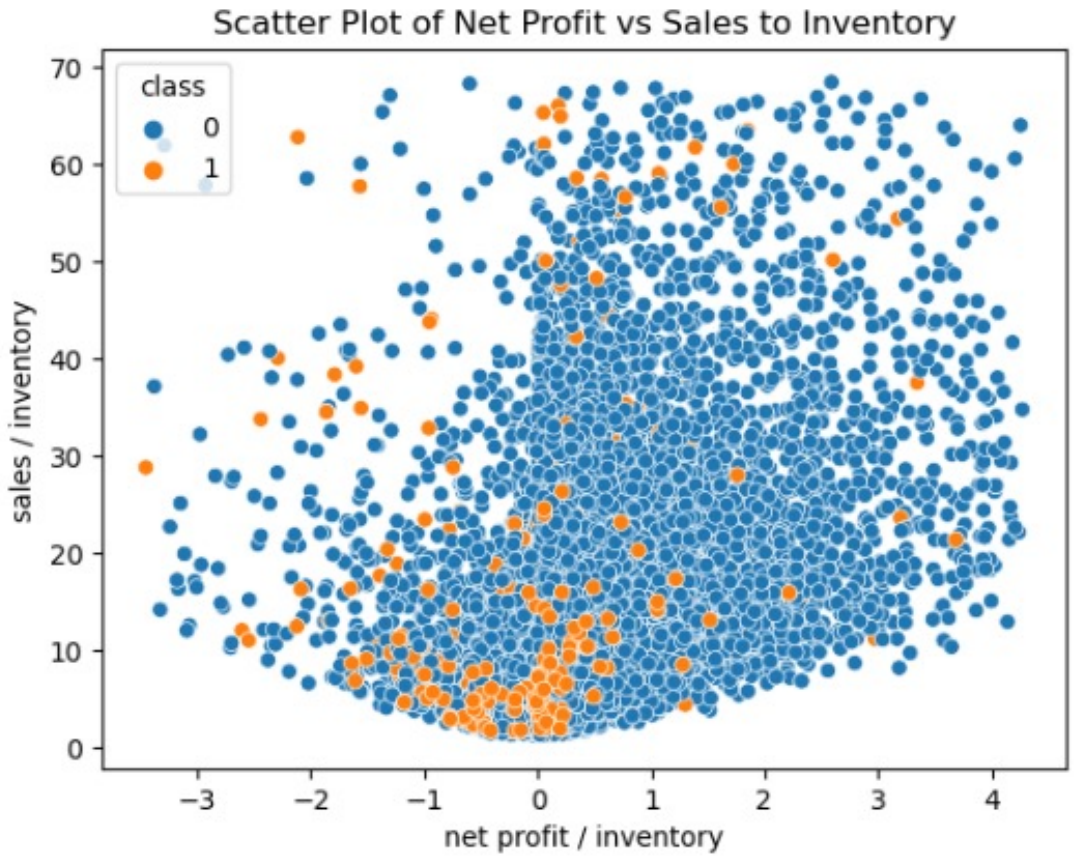
We see that the class 1 is gathered in the middle of the graph.

total assets / liabilities : this ratio measures what the company ‘has’ divided by what the company ‘owes’ at total in year. This ratio will be high for a company with high total assets and low liabilities.

Current assets / short term liabilities : same as the total assets / liabilities but in a short period of time.

Indeed, a company that will bankrupt has ‘total assets / liabilities’ , ‘current assets / short term liabilities‘and ‘sales/total assets’ inferior to a company that won’t bankrupt.

Other visualisations



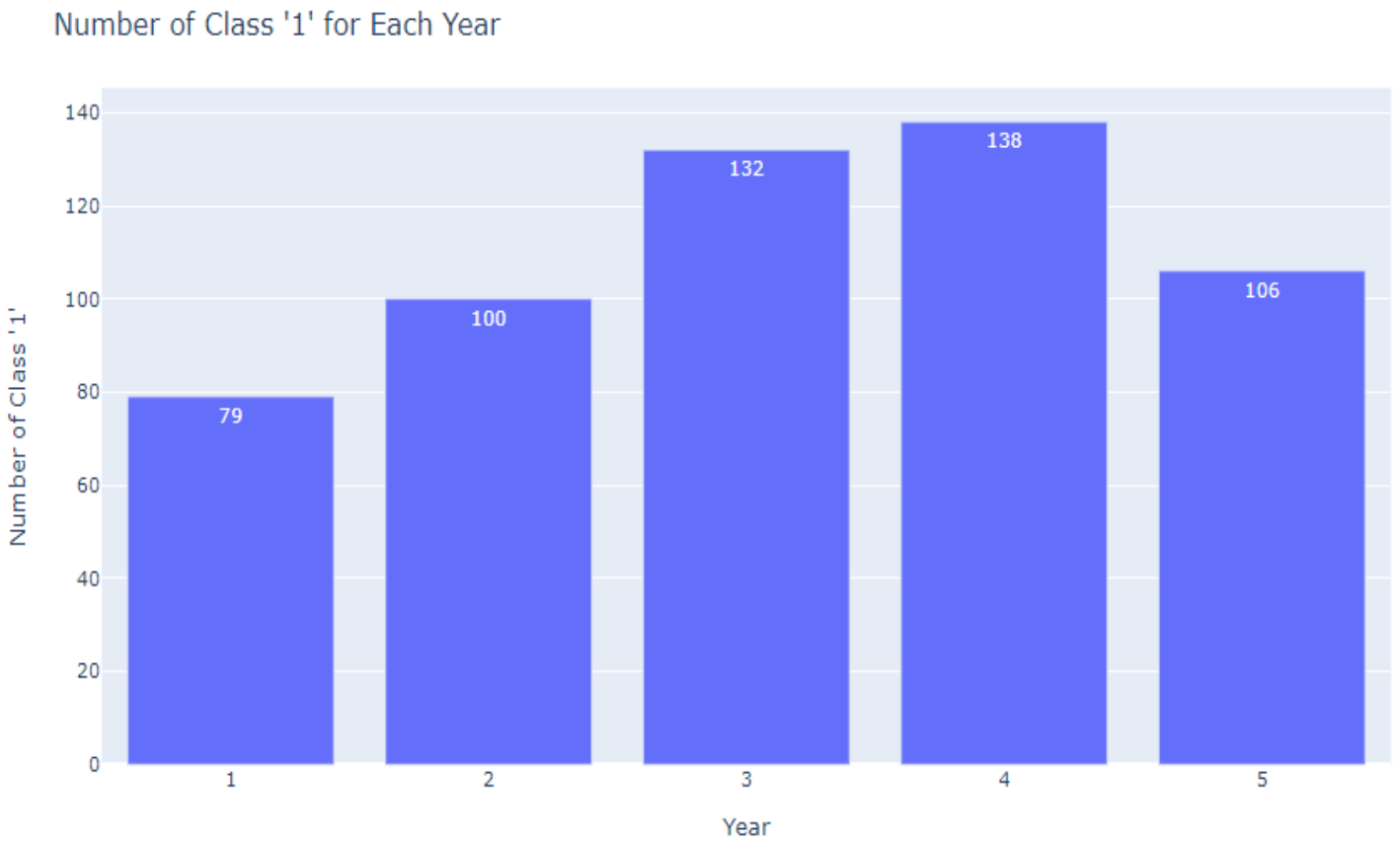
We see that the ‘class 1’ companies sell less (sales/inventory are low) and make less net profit

The class 1 is gathered next to 0. Indeed, the net profit and sales are low for a company that is likely to go bankrupt.

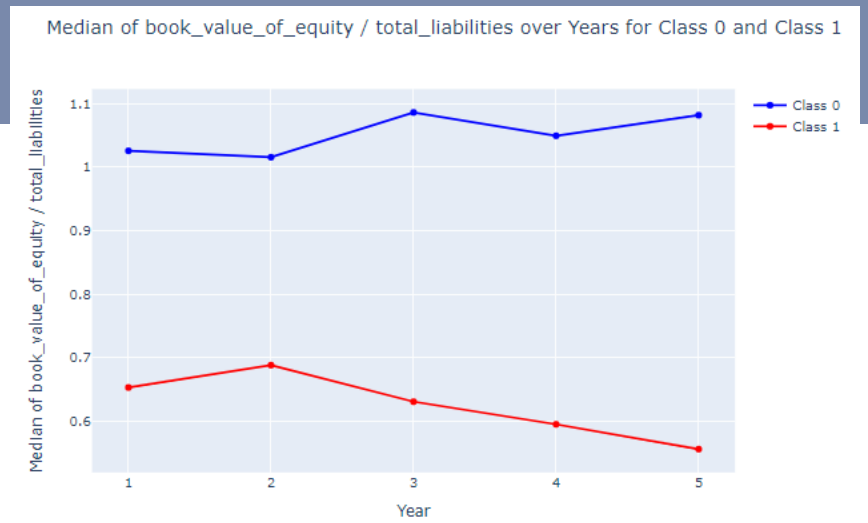
Other visualisations

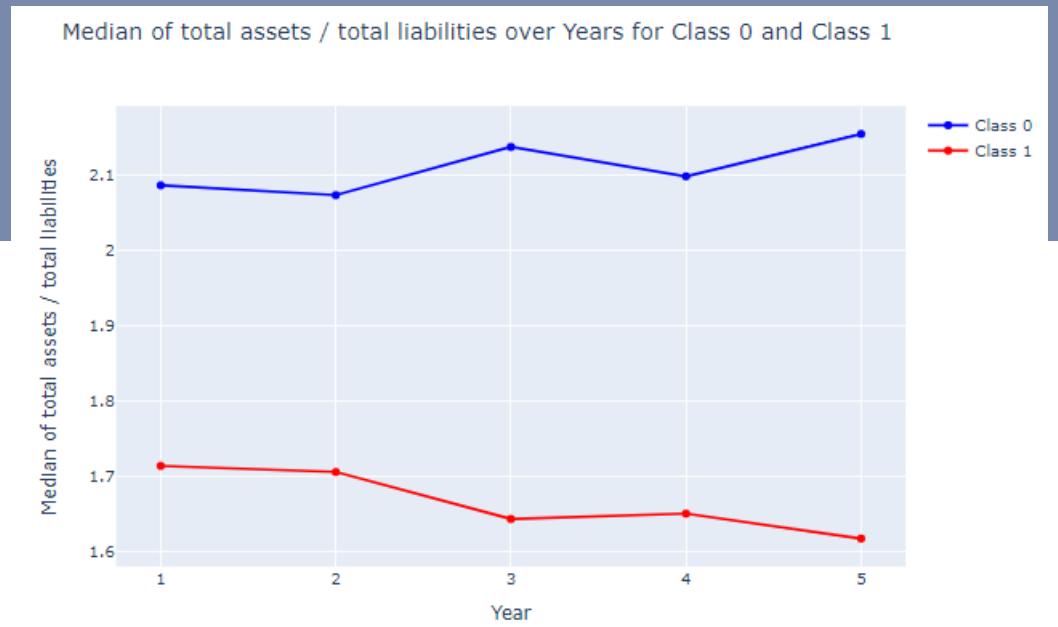
Let’s see the influence of the year regarding the bankruptcies.

There is a peak during year 4.



Other visualisations





Machine learning

- In our case the target is the class so it will be th Y and the features will be the X.
- Then we apply the method train_test_split to have our x_train,x_test,y_train and y_test
- Encode the data: We need to encode in binary the class (in our case it is almost the case we have b’0 and b’1 we just need to convert the type) but in some case we can use LabelEncoder.
- Normalisation of the data: We have chose firstly to predict without scaler to see the influence of normalisation then with:

a) Standardscaler:scales the data in a way that the average of each feature becomes 0, and the standard deviation becomes 1 (z-score normalization). It assume that the data fills a Gaussian distribution. Standardization is robust to outliers because it uses the statistical state of data with mean and standard deviation, which are less affected by extreme values. It is commonly used when applying machine learning algorithms that assume normally distributed features or when you want to scale your data to unit variance. How to calculate : xstd = (x−mean) /std

b) Minmax scaler: scales each feature individually where the values fall within a specified range, usually [0, 1]. This process does not assume any particular distribution of the data. The resulting normalization is sensitive to outliers because it uses the minimum and maximum values to perform the scaling. This method is commonly used when you want to bring all features to a common scale, especially when using algorithms that rely on distances or gradients, such as k-nearest neighbors (KNN) or gradient descent-based methods. How to calculate : xmin−max =(x−min)/(max−min)

In our case we have seen first of all that we have a bad precision,recall and f1 score for the class 1.It is due to the number of value for class 1(less companies in bankruptcy case).So after that it will be complicated to predict if a companies will bankrupt.or not.

Solution:
We use SMOTE method,it works by generating synthetic examples for the minority class, helping to balance class distributions. SMOTE selects instances from the minority class, creates synthetic examples by interpolating(Linear interpolation calculates the values of the new synthetic examples at regular intervals along these lines) between them and their nearest neighbors, and then adds these synthetic examples to the original dataset.

Then, we need to find the best parameters by using grid search for each of the model that we want to try.
We have choose to train with various existing models then compare the with metrics.
We have choose the following ones:

- Random forest classifier:A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- SVC:Support Vector Classifier is a machine learning algorithm used for classification working by finding the hyperplane that best separates different classes in the feature space.
- GradientBoostingClassifier:ensemble method, that works by building a series of weak learners sequentially. Each tree corrects the errors of the previous one, focusing on instances that were misclassified.
- LogisticRegression:used for binary classification tasks. It models the probability of the occurrence of an event by fitting a logistic curve to a linear combination of input features.

Then we predict by fitting ou training and use predict function.

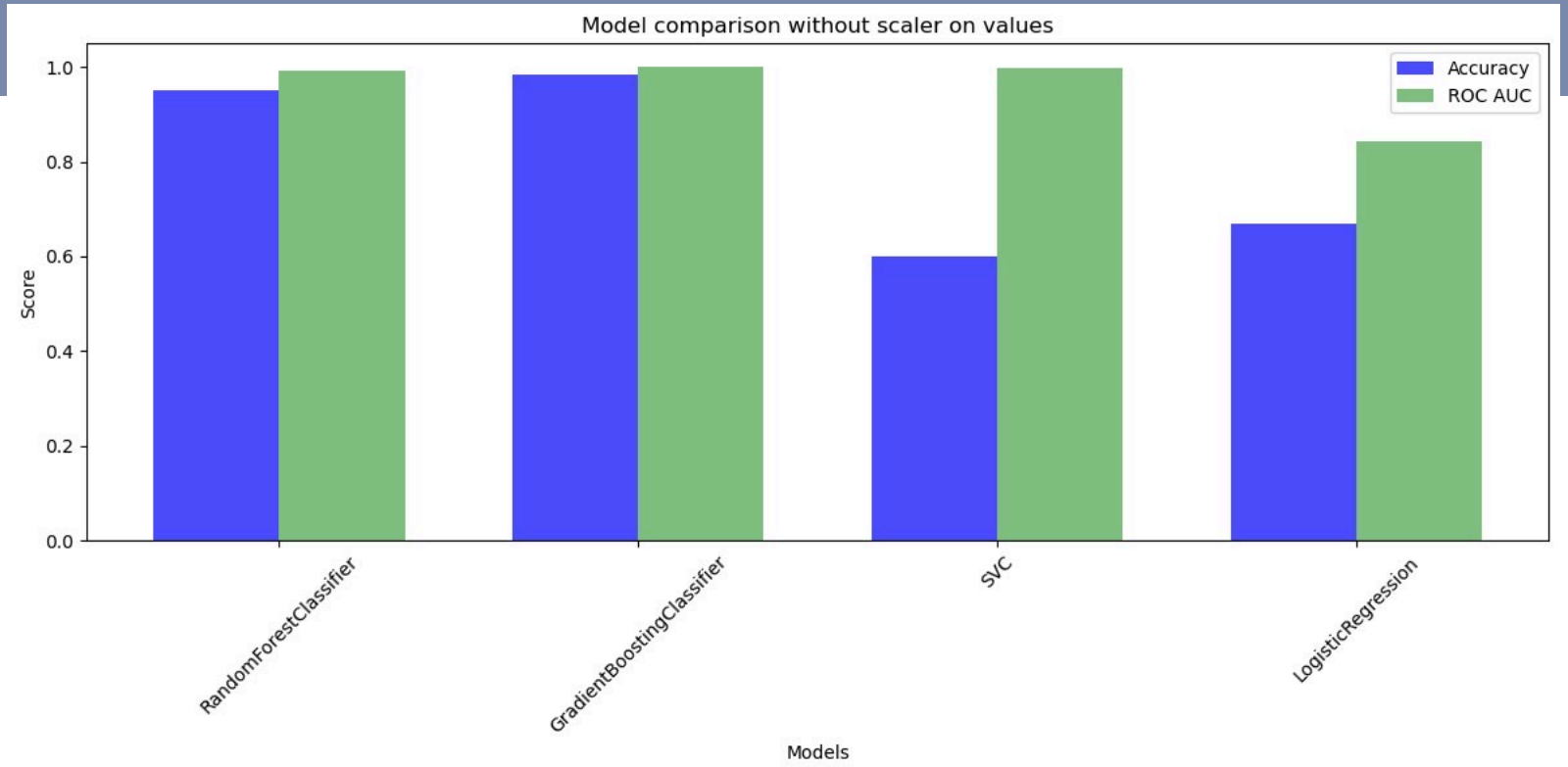
Metrics

Let note 0 as a negatif prediction and 1 as a positive one. We can obtain 4 measures: true positive (TP :true 1), true negative (TN: true 0), false positive (FP: false 1) and flase negative (FN : false 0). Let detail now each of them.
TP = (y_predict = 1 when y_true = 1) = NB_1_predict/NB_1_true
TN = (y predict = 0 when y _true = 0) = NB_0_predict/NB_0_true
FP = (y predict = 1 when y_true = 0) = NB_1 _Falsepredict/NB_0 _true
FN = (y predict = 0 when y_true = 1) = NB 0_predict/NB_1 _true
Precision (P) = TP /(TP +FP)
Recall (R) = TP /(TP +FN)
F-score = 2* (PR/ (P +R))
The global accuracy is calculated as: TP +TN /len(y test)
Accuracy values range from 0 to 1, where: An accuracy of 1 (100%) indicates that the model's predictions are entirely correct, and there are no prediction errors. An accuracy of 0 (0%) indicates that the model's predictions are entirely incorrect, and none of the predictions are correct.

We can use also confusion matrix it's a table that summarizes the performance of a machine learning classification model by breaking down predictions into categories: true positives, true negatives, false positives, and false negatives.

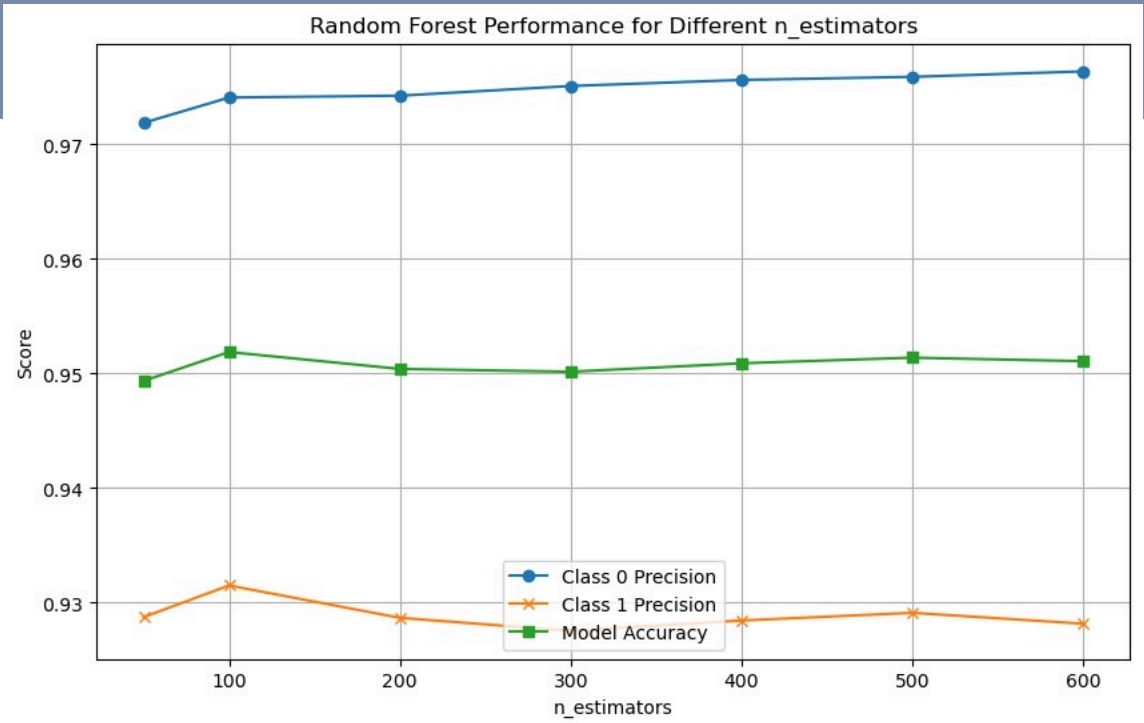
Comparaison between models without scaler

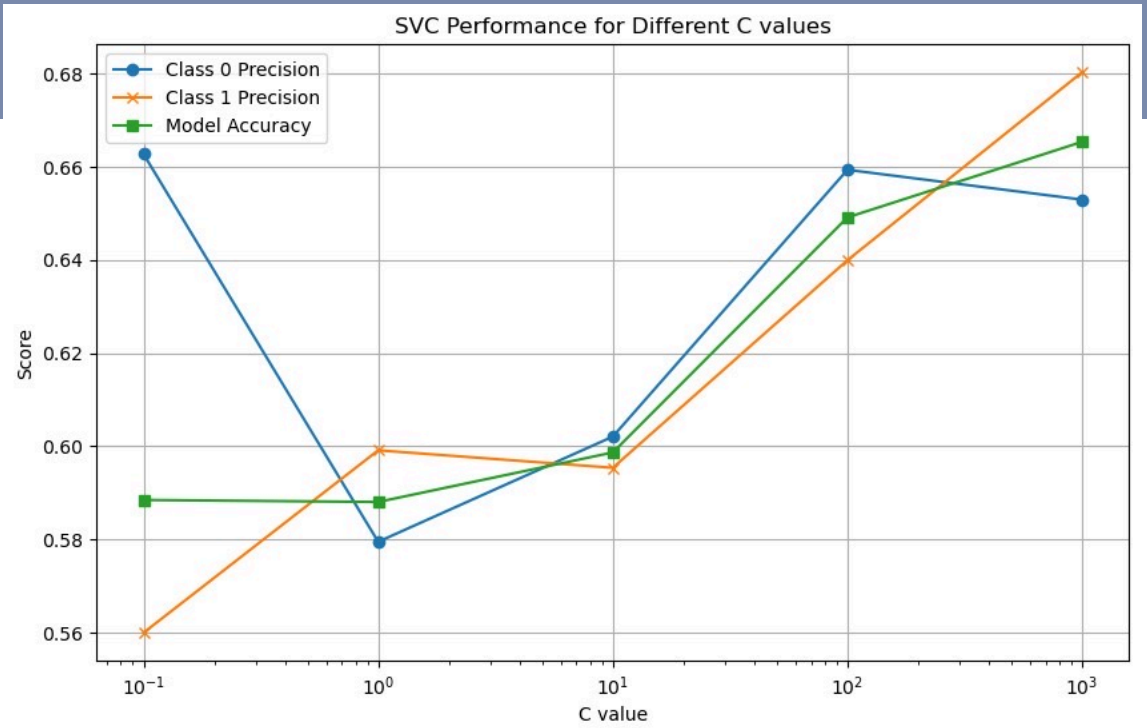
```
Model: RandomForestClassifier
  Accuracy: 0.9504
  ROC AUC: 0.9909
  Class 0: Precision: 0.9742, Recall: 0.9256, F1: 0.9493
  Class 1: Precision: 0.9287, Recall: 0.9754, F1: 0.9514
------------------------------------------------------------
Model: GradientBoostingClassifier
  Accuracy: 0.9879
  ROC AUC: 0.9994
  Class 0: Precision: 0.9931, Recall: 0.9827, F1: 0.9879
  Class 1: Precision: 0.9828, Recall: 0.9931, F1: 0.9879
------------------------------------------------------------
Model: SVC
  Accuracy: 0.5987
  ROC AUC: 0.6450
  Class 0: Precision: 0.6021, Recall: 0.5903, F1: 0.5961
  Class 1: Precision: 0.5954, Recall: 0.6071, F1: 0.6012
------------------------------------------------------------
Model: LogisticRegression
  Accuracy: 0.6684
  ROC AUC: 0.7136
  Class 0: Precision: 0.6758, Recall: 0.6519, F1: 0.6636
  Class 1: Precision: 0.6615, Recall: 0.6850, F1: 0.6730
------------------------------------------------------------
```



We see that the best model is for each metrics is the GradientBoostingCLassifier and the worst is the SVC. So, after we will focus on the influence of the parameter for both models.

Influence of the parameters (without scaler)



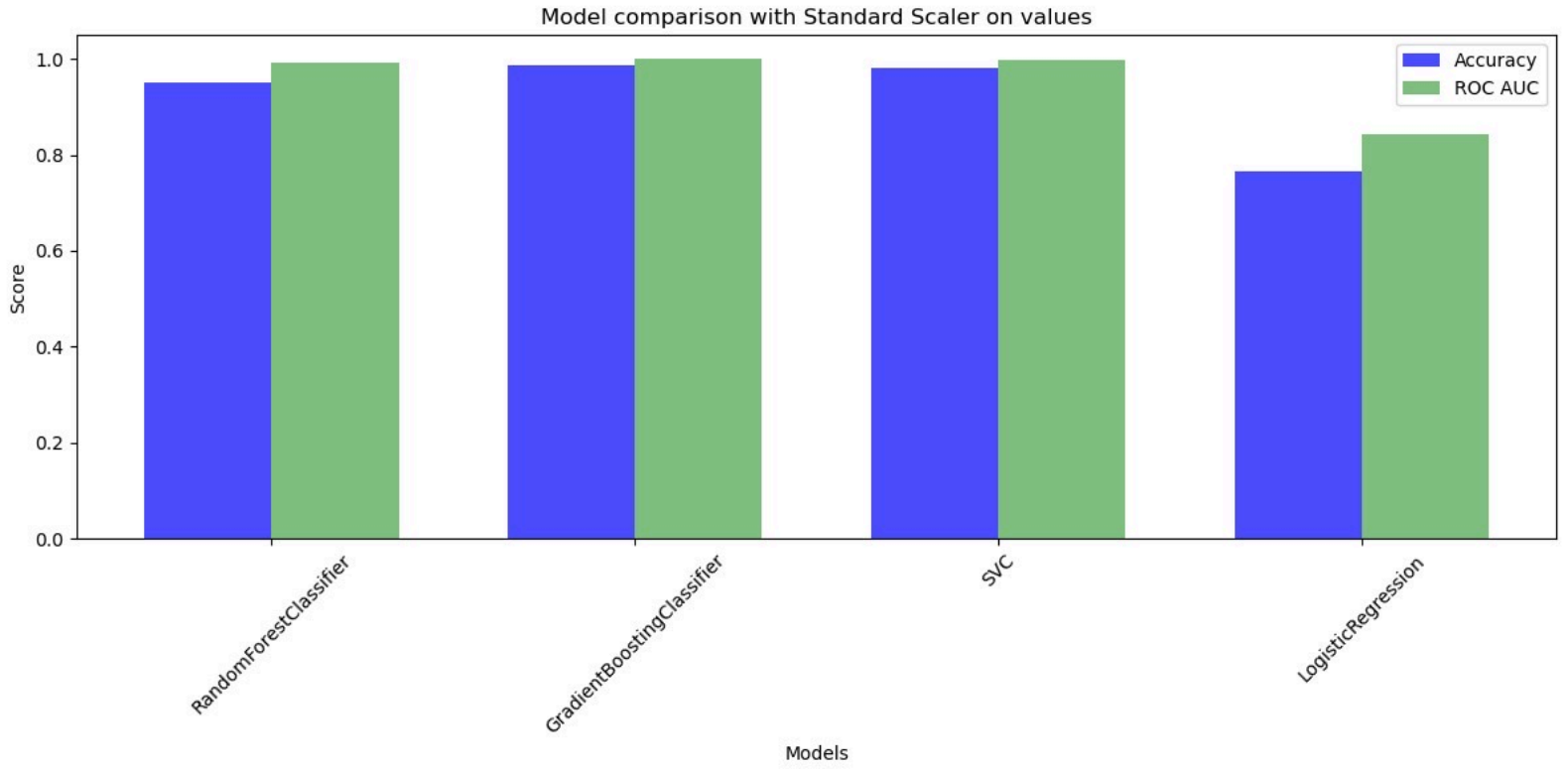


We see that the accuracy increase with the number of estimators this may be due to the increasing complexity of the model which does not necessarily translate into better generalization. It is possible that the model becomes too complex and begins to overfit to the training data, harming its ability to perform well on unseen data (test set).

The curves shows the impact of the modification of the C value for a SVC model. Different values of C imply a different trade-off between model complexity and error tolerance on the training data. Fluctuations in scores indicate that the model responds differently to regularization, where some values of C lead to better performance than others, reflecting the trade-off between bias and variance.

Comparaison between models with standard scaler

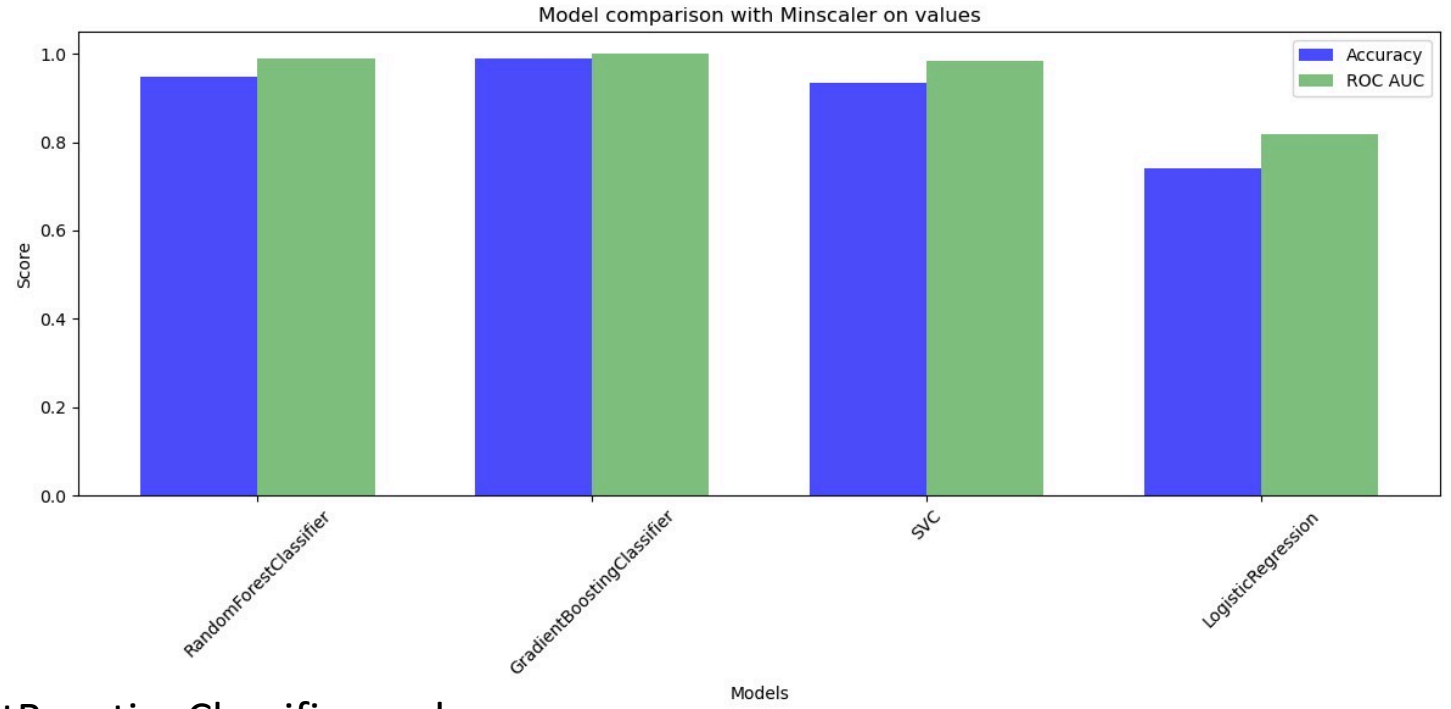
```
Model: RandomForestClassifier
  Accuracy: 0.9506
  ROC AUC: 0.9909
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: GradientBoostingClassifier
  Accuracy: 0.9877
  ROC AUC: 0.9994
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: SVC
  Accuracy: 0.9810
  ROC AUC: 0.9982
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: LogisticRegression
  Accuracy: 0.7644
  ROC AUC: 0.8438
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
```



We see that the best model here is also GradientBoostingClassifier and the worst is LogisticRegression.

Comparaison between models with minmax scaler

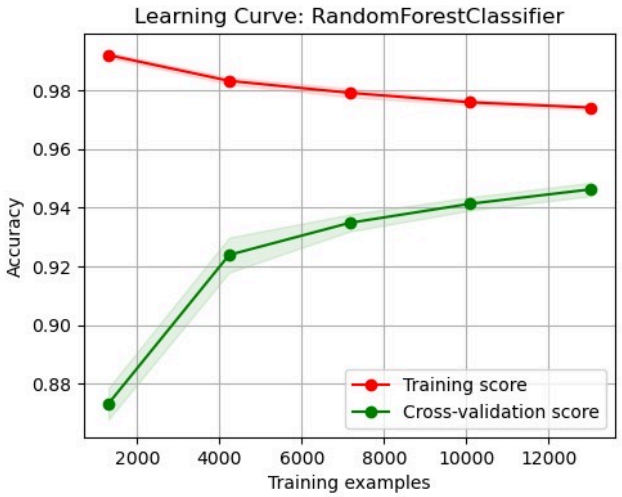
```
Model: RandomForestClassifier
  Accuracy: 0.9487
  ROC AUC: 0.9908
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: GradientBoostingClassifier
  Accuracy: 0.9885
  ROC AUC: 0.9994
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: SVC
  Accuracy: 0.9332
  ROC AUC: 0.9836
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
Model: LogisticRegression
  Accuracy: 0.7418
  ROC AUC: 0.8184
  Class 0: Precision: 0.9764, Recall: 0.9249, F1: 0.9499
  Class 1: Precision: 0.9282, Recall: 0.9774, F1: 0.9522
------------------------------------------------------------
```

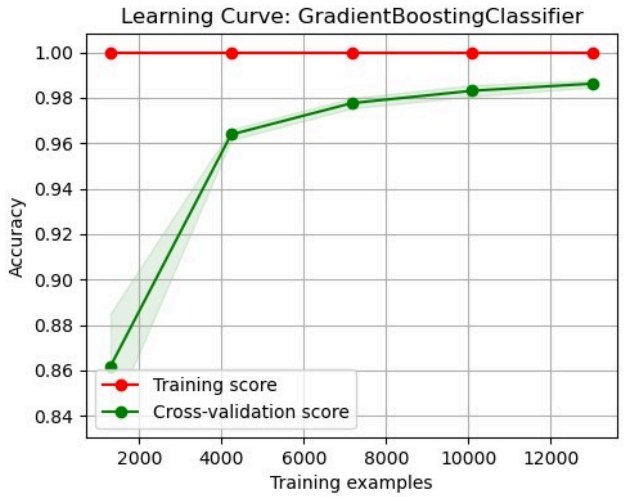


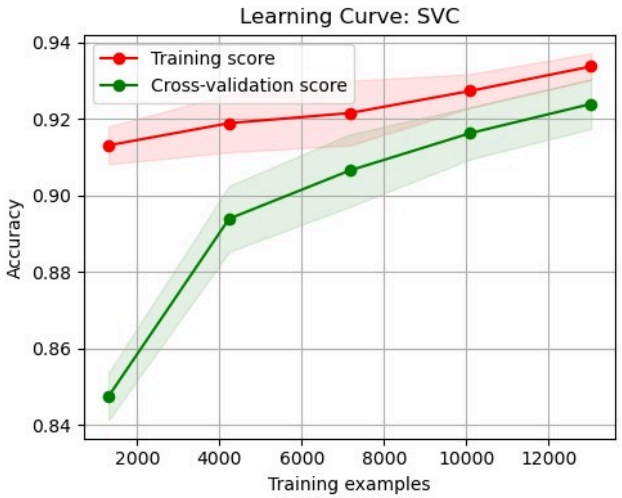
We see that the best model here is also GradientBoostingClassifier and the worst also LogisticRegression

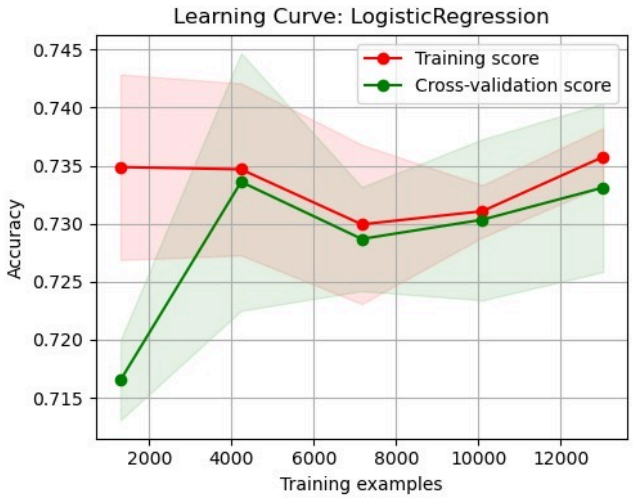
In the absence of scaling, model performances vary, with GradientBoostingClassifier achieving the highest accuracy and ROC AUC. However, SVC and LogisticRegression exhibit lower performance, suggesting sensitivity to data scale. When applying StandardScaler, which normalizes data, SVC shows a substantial improvement, emphasizing its benefit from scaled data. Other models see modest enhancements or maintain similar performance compared to non-scaled data. MinMaxScaler, scaling within a specified range, positively impacts SVC and slightly influences other models. To summarize, scaling significantly benefits scale-sensitive models like SVC, while models with internal normalization mechanisms, such as RandomForest and GradientBoosting, are less affected. LogisticRegression also benefits from scaling, albeit less dramatically than SVC.

Learning curve







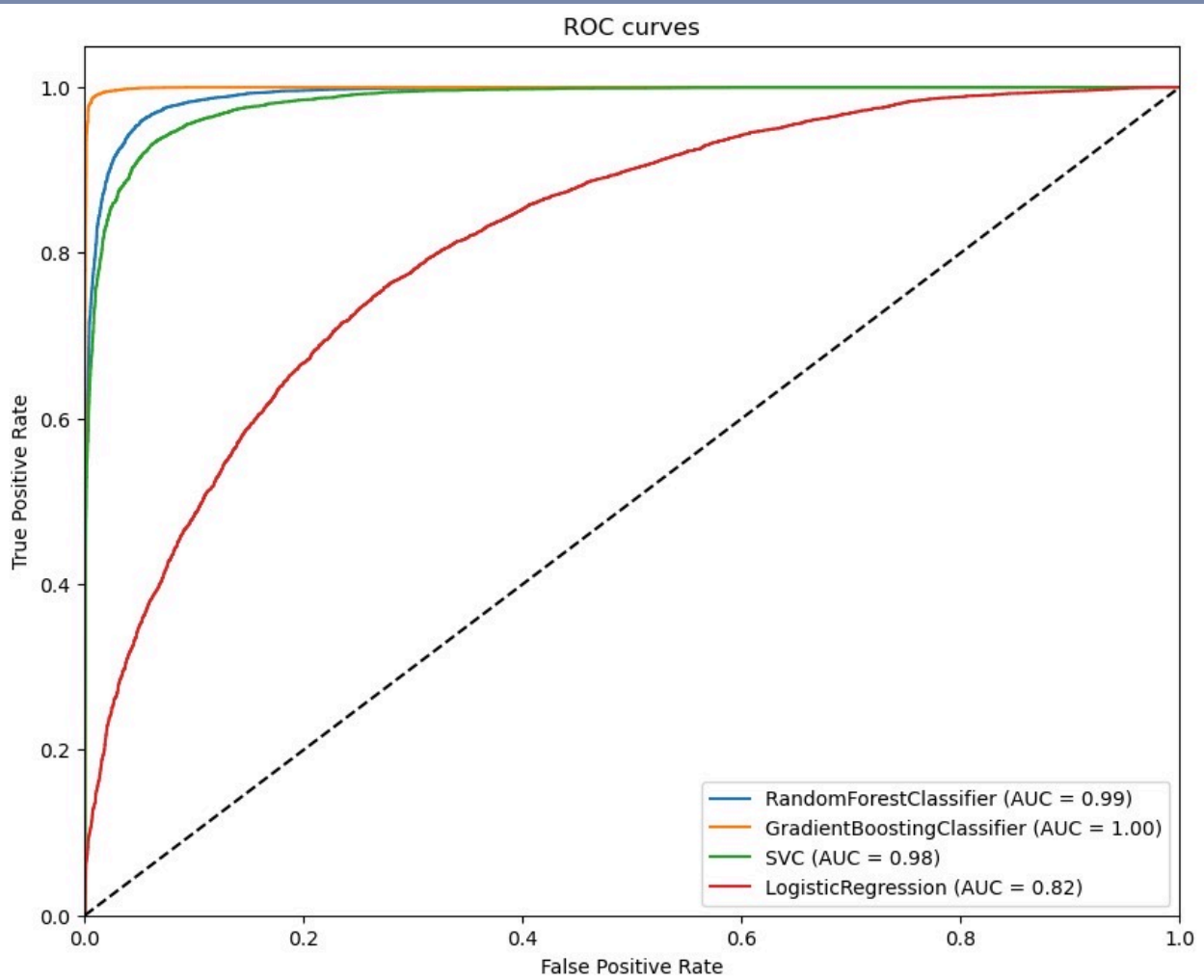


RandomForestClassifier et GradientBoostingClassifier : The accuracy on both the training set and cross-validation increases with more training examples, indicating that these models benefit from more data. However, the gap between training and validation scores remains constant, which could indicate a good balance between bias and variance.

SVC:The training and validation accuracy also increases, but the gap between them decreases slightly, suggesting that the model might be slightly overfitting when less data is available.

LogisticRegression : Training accuracy starts higher but converges to validation accuracy with increasing number of examples. This may indicate initial overfitting that decreases with more data, but the overall accuracy is lower than for other classifiers.

ROC curves



The ROC curves show what we have said that the best model here is GradientBoostingClassifier (it's the perfect classifier) and the worst is the LogisticRegression which is the closest to the random classifier.

Conclusion

In the initial phase of our machine learning project, we successfully overcame the challenge of having a dataset with two extremely imbalanced classes, which made machine learning unfeasible. Subsequently, we analyzed the impact of different scalers and modifications to certain learning parameters on our learning results. We observed that scalers had a significant impact on the SVC model, which was until then the least performing. However, for the already high-performing models, the addition of scalers did not have a significant impact. Finally, we displayed ROC curves and learning graphs, which confirmed that our models were performing well. We were able to highlight that all our models were functioning; some did not need scaling to achieve good results while others required it for coherent outcomes. Conversely, we saw that modifying the learning parameters had a minimal impact on the learning results compared to the choice of model, having balanced classes, and potentially using a scaler depending on the model.