Python for Data Science

Machine learning for bankruptcy detection

By

Michel Daher Mansour

04/11/2024

Introduction

This project is realized in the context of the python for data science class of the IAC Masters.

Bankruptcy of companies is evaluated based on several financial indicators such as growth rate, return on investment, debts, etc. This study explores the use of machine learning (ML) to predict if a company is going to be bankrupt or not by analyzing these financial indicators for more than 5000 company. As bankruptcy has significant economic consequences, early detection can help companies, investors, and regulators mitigate potential losses. By applying ML classification techniques to financial datasets, this study aims to develop a model that detects companies at risk of bankruptcy.

In this work, we will leverage various financial indicators such as Continuous interest rate (after tax), Operating Expense Rate, etc, to train models capable of separation between financially stable and potentially bankrupt companies.

Therefore, you will find sections that explain the data treatment made on our dataset, the pipelines used to create the models and finally the results of this work. Additionally, you will find an Annex that resume the features.

Data Processing

a. Data exploration

To use data for any model, we should first understand it. Therefore, we explore and analyze the data as a first step. In this context, our raw dataset consists of **96 features** that represent the companies' financial insights of **6819 company**. For the features we have 93 float type and 3 integer type that include the target which indicates if a company is bankrupted or not.

There is no Nan values presented in this dataset; therefore, there is no need to impute or remove any feature or row, respectively.

All the features are numerical feature, therefore no need for OneHotEncoding or LabelEncoder in order to replace categorical features into numerical ones.

Furthermore, we study the correlation of the features, because for a healthy model the features of the dataset should not be highly correlated. In our case, we have 25 highly correlated features with a Pearson coefficient higher than 0.8. Thus, we drop those highly correlated features to obtain a new reduced dataset of 71 features. The target, 'Bankrupt?', is not highly correlated with the rest of the features. Figure 1 shows the treatment we did and how we kept only one feature of the correlated ones.

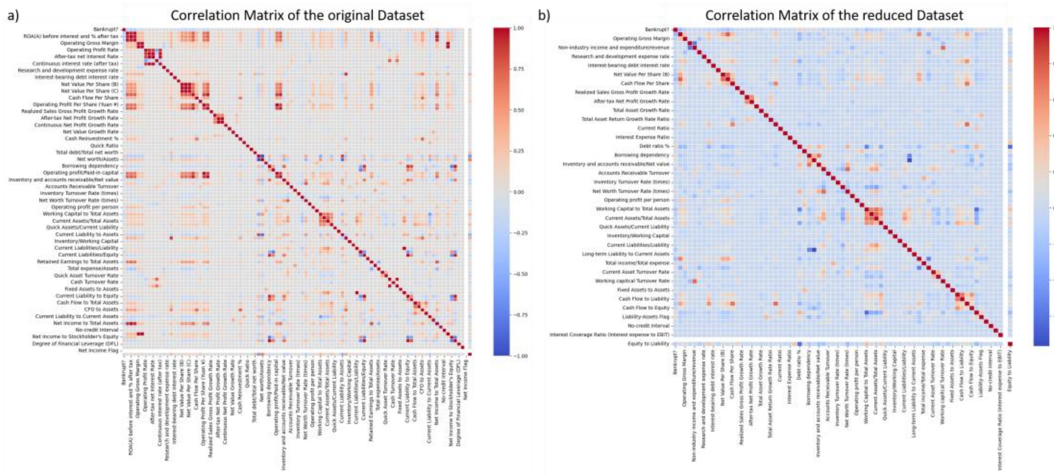

Figure 1: the correlation matrix, before the treatment in (a) and after we remove the highly correlated features in (b).

On the other hand, we show the statistics of the dataset, we have only 3.2% of the dataset with bankrupted companies.

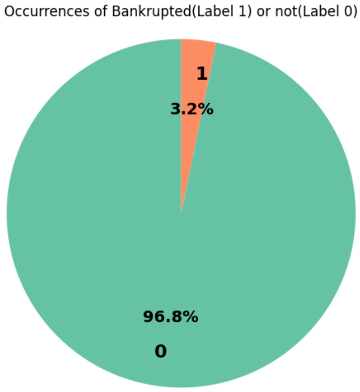

Figure 2: In this figure, we show the distribution of the dataset, in our dataset we have only 3.2% bankrupted companies whereas 96.8% not bankrupted.

Thus, with this imbalanced dataset we have to take two actions:

- For the scoring, we will consider the F1 score to evaluate the model.
- We will resample the dataset with oversampling in order to create a more balanced dataset and check the results.(this step occurs when we split the dataset into train and test sets, in the next subsection)

In the Notebook, we show the distribution of each feature, and we compare for each feature the distribution of the data for bankrupted companies' vs the financially stable ones. In Figure 3, we illustrate these distributions for only two features, the ROA before interest and the Net value per share.

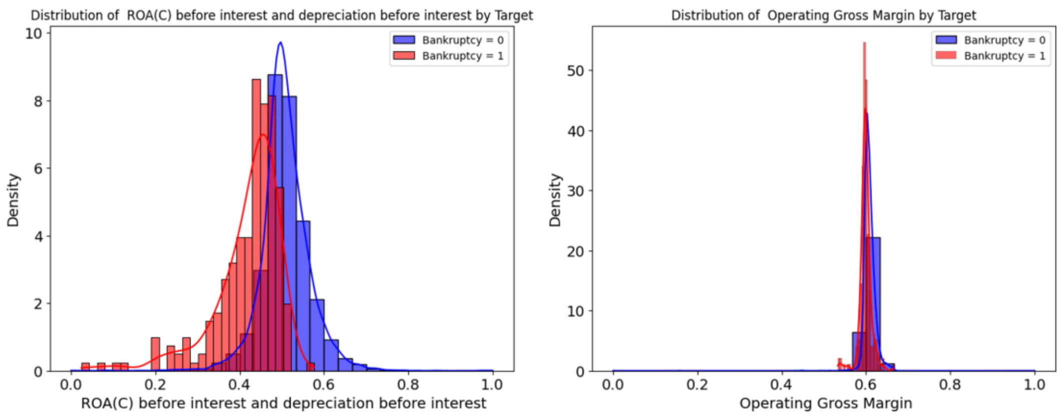

Figure 3: Distribution of the data for the bankrupted (in red) and the not bankrupted(in blue) companies for two features.

b. Data processing

In order to train and test the model, we need to have two sets of dataset, a train and test datasets. Therefore, we split the dataset into a train and test samples. In our case, we split it with a ratio of 30%, thus 70% of the dataset serves to train the model, and we test the model on the 30%.

We used 'stratify' in the train_test_split function in order to have similar proportions of bankrupted companies in both samples.

At this step, we apply on both datasets, the train and test samples, the oversampling in order to get balanced datasets.

Models

In our case, we are in a supervised problem where the dataset labels each company if it is bankrupted or not. Therefore, we used several models known for this case and then we chose the one with the best performance. In our selection, we used the:

- Logistic regression(LR)
- Random Forest(RF)
- Gradient Boosing (GB)
- HistGradient Boosting. (HGB)

We created a pipeline for each model the pipeline consisted on:

1. StandardScaler step that standardizes features by removing the mean and scaling to unit variance.
2. One of the models mentioned earlier. (LR, RF, GB, HGB)

In addition, we applied a GridSearchCV in order to finetune the hyperparameters. Thus, for each model, we applied several parameters and we select the ones with the best performances.

Results & Discussion

In this section, I show the results of the models for both datasets, the one before and after oversampling. This way, we get both results and I show the effect of resampling. We show the results on the test samples.

The results of the models with the dataset **before oversampling**:

Model	Precision	Recall	F1 score
Logistic Regression	0.67	0.58	0.60
Random Forest	0.78	0.58	0.62
Gradient Boosting	0.67	0.59	0.62
HistGradient Boosting	0.81	0.63	0.68

The results of the models with the dataset **after oversampling**:

Model	Precision	Recall	F1 score
Logistic Regression	0.84	0.84	0.84
Random Forest	0.86	0.83	0.83
Gradient Boosting	0.87	0.84	0.83
HistGradient Boosting	0.89	0.87	0.87

By resampling the dataset, we improve the model performance. We pass from an F1 score average of 63% for the original dataset to 84% for the oversampled dataset. This happens because the model is trained on a balanced dataset, which makes it easier for the model to detect the bankrupted cases.

In our case, HistGradient Boosting has the best performance with 87% for the F1 score. This is due to its internal functioning (its architecture makes it perform better and faster on large dataset).

Not all models have the attribute feature importance; in our case, we can use it only for the Random Forest model.

Thus, the most important features in the case of the RF with the best parameters are:

For the original dataset(before resampling):

- Net Value Growth Rate
- Net Value Per Share (B)
- Persistent EPS in the Last four Seasons
- Borrowing dependencies
- ROA(C) before interest and depreciation interest

For the oversampled dataset(after resampling):

- Persistent EPS in the Last four Seasons
- Borrowing dependencies
- ROA(C) before interest and depreciation interest
- Retained Earnings to Total Assets
- Total debt/Total net worth

The features are explained in the Annex section1. The change of the important features can be explained by the change of the dataset insights due to the oversampling.

Conclusion

In this work, we trained a model in order to identify if a company is bankrupted or not. We used a dataset of financial insights of 6819 companies.

Firstly, we proceed with the exploration of the dataset, where we identified the highly correlated features, we selected the features correlated with a Pearson coefficient less than 80%. Then, we made statistical study on the dataset where we identified that only 3.2% of the dataset indicated information of bankrupted companies, which highlighted the fact that we need to do a resampling of the dataset and select the F1 score as a metric to evaluate the model. In addition, we showed the different distribution for each feature in both cases bankrupted or not.
Then, we explained the split we did on the dataset in order to train and test the model.

Secondly, we described the pipelines, consisted of two steps: (i) the StandardScaler that we applied on the dataset and (ii) the model after it. For the model, we selected a group of models that fit our supervised modeling case. In this work, we selected the LR, RF, GB and HGB.

Finally, we showed the results for each model we tested and selected the one with the best performances. The HGB presented the best score with 87% on F1 score on resampled dataset instead of 68% for the original dataset. With this result, we show that oversampling helped the model to identify the bankrupted companies.

As a perspective, the score could be improved furthermore by applying statistical techniques as PCA to reduce the dimensionality or outlier detection on the datasets before the application of the pipeline. In addition, other techniques might be interesting; since we only have 3.2% of the dataset as bankrupted we could drop the label('Bankrupt?') and treat the problem as an unsupervised case and use anomaly detection algorithm such as Isolation Forest or OneClassSVM.
To apply, the notes we used in class, I grouped the different procedures in one Class that I present in the Annex of the Notebook. In addition, I present a dataset on which we can apply the OneHotEncoding function in order to transform categorical data into numerical one.

Annex

1. Feature in the dataset

Bankrupt?: **Class label**
ROA(C) before interest and depreciation before interest: Return On Total Assets(C)
ROA(A) before interest and % after tax: Return On Total Assets(A)
ROA(B) before interest and depreciation after tax: Return On Total Assets(B)
Operating Gross Margin: Gross Profit/Net Sales
Realized Sales Gross Margin: Realized Gross Profit/Net Sales
Operating Profit Rate: Operating Income/Net Sales
Pre-tax net Interest Rate: Pre-Tax Income/Net Sales
After-tax net Interest Rate: Net Income/Net Sales
Non-industry income and expenditure/revenue: Net Non-operating Income Ratio
Continuous interest rate (after tax): Net Income-Exclude Disposal Gain or Loss/Net Sales
Operating Expense Rate: Operating Expenses/Net Sales
Research and development expense rate: (Research and Development Expenses)/Net Sales
Cash flow rate: Cash Flow from Operating/Current Liabilities
Interest-bearing debt interest rate: Interest-bearing Debt/Equity
Tax rate (A): Effective Tax Rate
Net Value Per Share (B): Book Value Per Share(B)
Net Value Per Share (A): Book Value Per Share(A)
Net Value Per Share (C): Book Value Per Share(C)
Persistent EPS in the Last Four Seasons: EPS-Net Income
Cash Flow Per Share
Revenue Per Share (Yuan ¥): Sales Per Share
Operating Profit Per Share (Yuan ¥): Operating Income Per Share
Per Share Net profit before tax (Yuan ¥): Pretax Income Per Share
Realized Sales Gross Profit Growth Rate
Operating Profit Growth Rate: Operating Income Growth
After-tax Net Profit Growth Rate: Net Income Growth
Regular Net Profit Growth Rate: Continuing Operating Income after Tax Growth
Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth
Total Asset Growth Rate: Total Asset Growth
Net Value Growth Rate: Total Equity Growth
Total Asset Return Growth Rate Ratio: Return on Total Asset Growth
Cash Reinvestment %: Cash Reinvestment Ratio
Current Ratio
Quick Ratio: Acid Test
Interest Expense Ratio: Interest Expenses/Total Revenue
Total debt/Total net worth: Total Liability/Equity Ratio
Debt ratio %: Liability/Total Assets
Net worth/Assets: Equity/Total Assets
Long-term fund suitability ratio (A): (Long-term Liability+Equity)/Fixed Assets
Borrowing dependency: Cost of Interest-bearing Debt
Contingent liabilities/Net worth: Contingent Liability/Equity
Operating profit/Paid-in capital: Operating Income/Capital
Net profit before tax/Paid-in capital: Pretax Income/Capital
Inventory and accounts receivable/Net value: (Inventory+Accounts Receivables)/Equity
Total Asset Turnover
Accounts Receivable Turnover
Average Collection Days: Days Receivable Outstanding
Inventory Turnover Rate (times)
Fixed Assets Turnover Frequency
Net Worth Turnover Rate (times): Equity Turnover
Revenue per person: Sales Per Employee

Operating profit per person: Operation Income Per Employee
Allocation rate per person: Fixed Assets Per Employee
Working Capital to Total Assets
Quick Assets/Total Assets
Current Assets/Total Assets
Cash/Total Assets
Quick Assets/Current Liability
Cash/Current Liability
Current Liability to Assets
Operating Funds to Liability
Inventory/Working Capital
Inventory/Current Liability
Current Liabilities/Liability
Working Capital/Equity
Current Liabilities/Equity
Long-term Liability to Current Assets
Retained Earnings to Total Assets
Total income/Total expense
Total expense/Assets
Current Asset Turnover Rate: Current Assets to Sales
Quick Asset Turnover Rate: Quick Assets to Sales
Working capitcal Turnover Rate: Working Capital to Sales
Cash Turnover Rate: Cash to Sales
Cash Flow to Sales
Fixed Assets to Assets
Current Liability to Liability
Current Liability to Equity
Equity to Long-term Liability
Cash Flow to Total Assets
Cash Flow to Liability
CFO to Assets
Cash Flow to Equity
Current Liability to Current Assets
Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise
Net Income to Total Assets
Total assets to GNP price
No-credit Interval
Gross Profit to Sales
Net Income to Stockholder's Equity
Liability to Equity
Degree of Financial Leverage (DFL)
Interest Coverage Ratio (Interest expense to EBIT)
Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise
Equity to Liability