Using ML algorithm to predict bankrupt of company
Data 1030 Final Project Report
Supervised by Prof. Andras Zsom
Github Link: https://github.com/shijiemao/data1030project
Shijie Mao

Introduction

Bankrupt is the situation when company is unable to pay outstanding debts. The dataset for this report is the bankrupt data of Taiwan companies under the Taiwan Stock Exchange derived from Kaggle[1]. The target variable is whether a company is bankrupt or not. So, the problem I investigate is a classification. And the features are the financial ratios of these companies. Financial ratio is a relative magnitude of two financial indexes in financial statements. According to accounting, financial ratios are directly related to the company bankrupt. But we usually use quality method to analyze the relationship between financial ratios and bankrupt. Thus, using quantity method to analyze the impact of different financial ratios and use financial ratios to predict the bankrupt situation is important.

The dataset is from 6819 Taiwanese companies. And it has 89 features which are different financial ratios. Most of them are continuous and well-document data except Liability-Assets Flag and Net Income Flag are categorical data and a little bit ambiguous. For the Liability-Asset Flag variable, it has value of 0 and 1. 1 means that Total Liability exceeds Total Assets, 0 otherwise. If Net Income Flag 1 then the company income is Negative for the last two years, 0 otherwise.

My data is from Kaggle, and it was collected on Taiwan Economic Journal for the years from 1999 to 2009. Some projects posted in Kaggle give me some valuable reflection. First is the "Bankruptcy Analysis" designed by Ginelle D'souza[2]. The summary of its EDA reveals some features of the data set like the relationship between different variables. Author tries different models and the final best model this analysis choose is random forest classifier and the result f-score is 98.75% to prediction whether company is bankrupt.

Exploratory Data Analysis

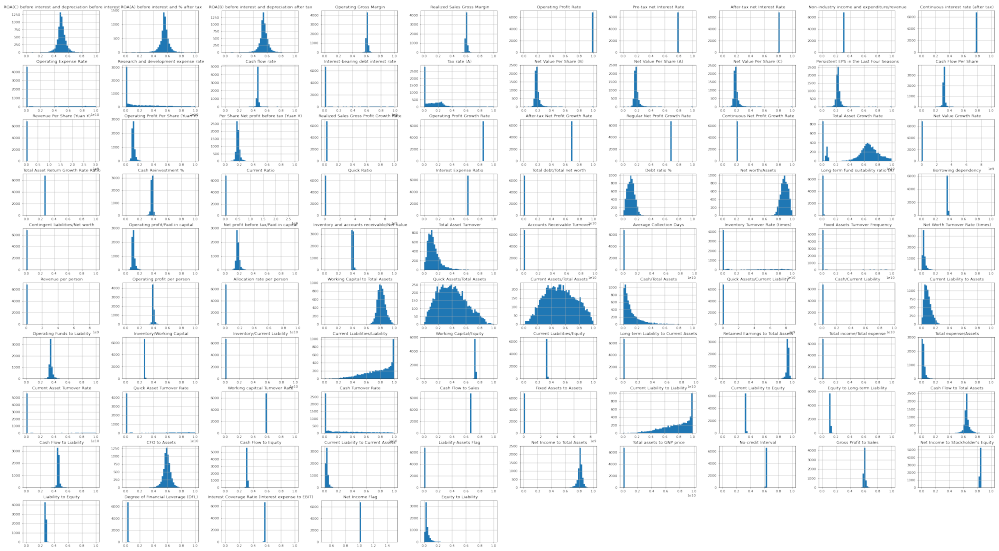

Figure 1 is the overview of all features. This overview told us that most features have extreme value of 1 or 0 to enlarge the bin_range and make the histogram extremely group at the middle. To fix it, the later picture would set a bin_range of 1 percentile to 99 percentile to omit the influence of extreme value to the plot. Also, some picture show that the features have a tail and they have to be standardized rather than minmax scaler.

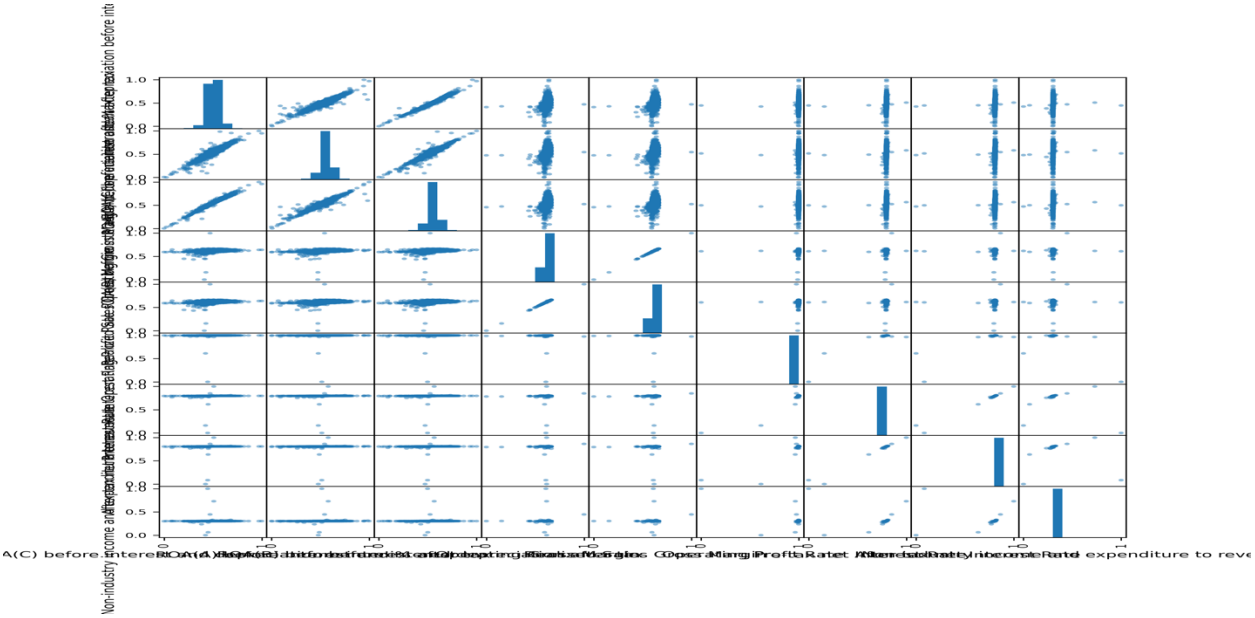

Figure2: Here is a scatter of first 10 features. We could see that some of them are highly related and some of them have not any correlation.

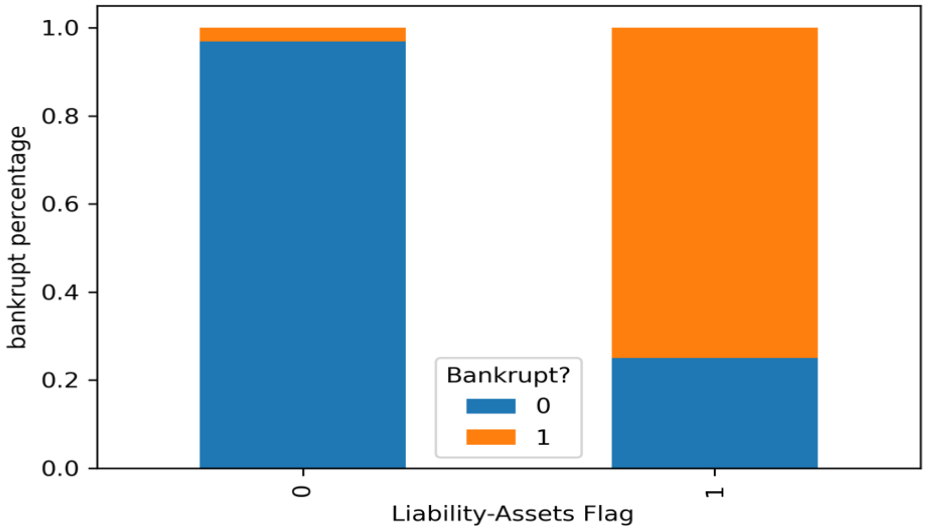

Figure3: This bar plot shows the percentage of bankrupt company among companies with different Liability-Asset Flag. 1 means that the Total Liability exceeds Total Assets. And we could clearly see that the companies with 1 Liability-Asset Flag have an extremely high percentage of bankrupt. This conclusion is reasonable and easy to see. When the company with a high Liability. If the debt of a company is larger than its asset, the company would have no equity and it would be easy to be bankrupt.

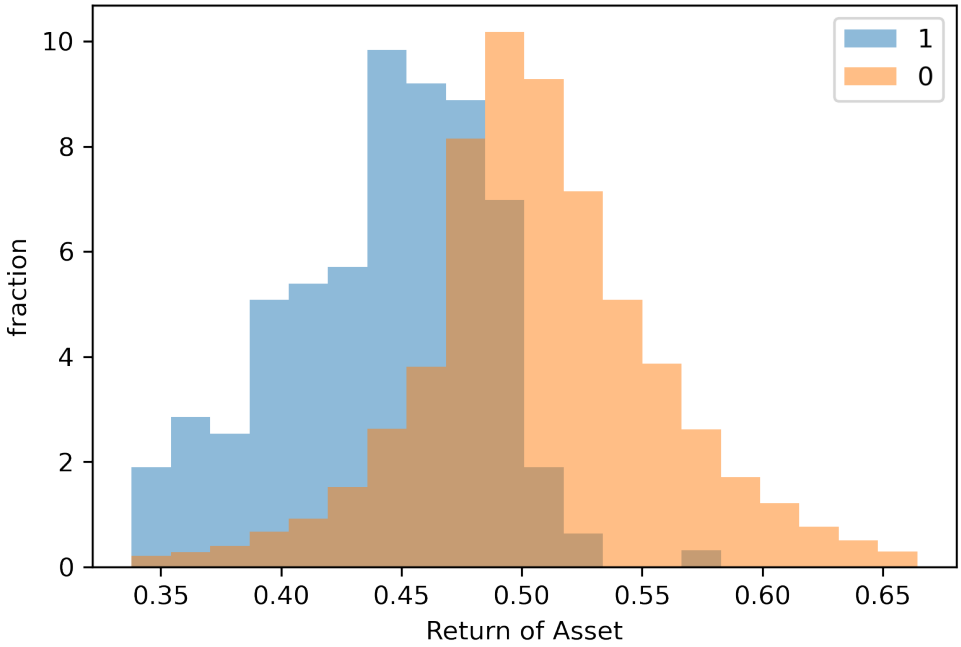

Figure 4 is the relationship between the feature ' ROA(C) before interest and depreciation before interest' and the target variable whether bankrupt or not. ROA is the ability of measure the ability of earning profit. Surely, a lower ROA would lead to a higher probability to bankrupt.

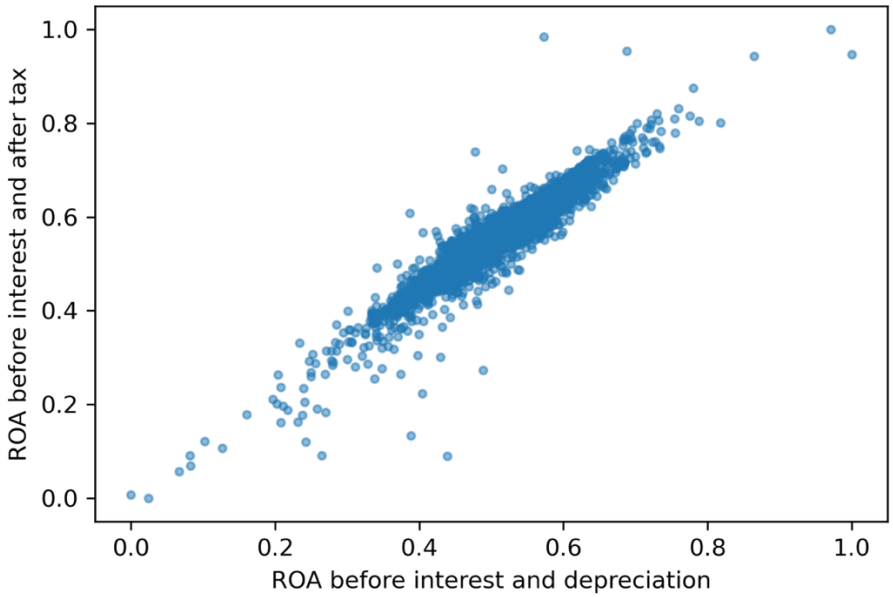

Figure 4 is the relationship between two features ROA (C) and ROA (A), From the scatterplot, these two variables are highly linear related. This need to be noticed and fixed in the later process feature engineering.

Method

Splitting Strategy. I use ratio of 0.8 and random state of 5 random numbers between 1 and 100 using a random.seed = 11 to make each model has 5 same random numbers to do the split. 80% of the data set are train and validation data. 20% are the test data. Then apply Kfold = 4 to train and validation data to do Cross-Validation within the random state of 5 random number mentioned above to find the best splitted model with a highest validation F1_score_macro. Kfold = 4 would set 60% of the data into training data 20 % of the data as validation data each fold to find the best model in each fold.

In bankrupt data, there are two categorical variables, and the rest are continuous variables. Thus, I use one hot encoding to deal with the categorical variable and standard scale for the rest. However, one categorical data has only one class. Thus, the number of features after the preprocessing is 95+1 = 96 rather than the 97 by intuition

After deciding the splitting and preprocessing method. I build four Machine Learning Model to find the model with each best parameter. Because my problem here is a classification problem. Thus the four models I choose are Logistic Regression, Support Vector Machine Classification I use gridsearch CV in sklearn to tune the hyperparameters. Evaluation metric here uses F1_score_macro.
First reason is according to the pie chart below of the response variable of my data set is an imbalanced dataset.

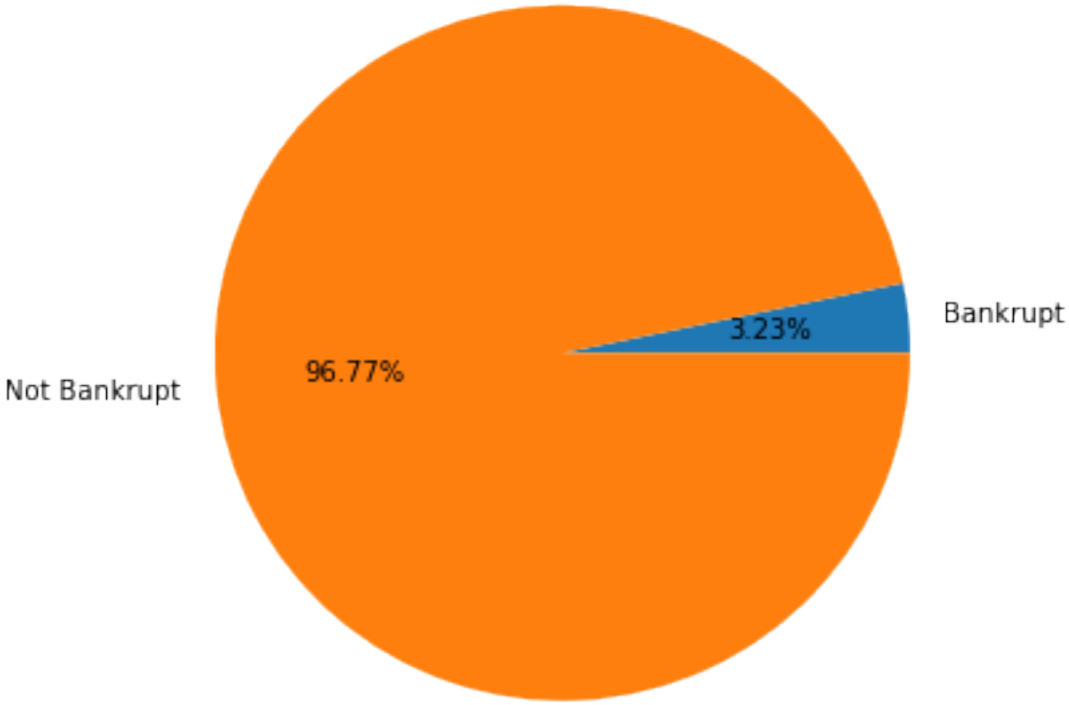

Figure 5 is the percentage of two classes of target variable.

Only 3.23% of the company is bankrupted, which categorical data is my response variable. In addition, I am more interested in the minority class which are the bankrupt companies and I want my model to predict the bankrupt companies, the minority accurate. Thus, I choose f1_score _macro as my evaluation metric to avoid the problem of imbalanced data.

The 4 ML algorithm

1. Logistic Regression
   Using solver = default, max_iter = 100000 to converge
   Tuned parameter: 'C' and 'Penalty'
   'C' is from 0 to 1. 1 is default value
   'Penalty' of 'l2' and 'l1'

2. Support Vector Machine Classification
   Kernel is default which is rbf
   Tuned parameter: 'C' and 'Gamma'
   'Gamma' is selected from [1e-2, 1e-1, 1e0, 1e1, 1e2]
   'C' is selected from np.logspace(-1, 1, 5)
3. KneighborsClassifier
   Tuned parameter: 'n_neighbors' and 'weights'
   'n_neighbors' is selected from [1, 3, 10, 30, 50, 100]
   'weights' is choosen from ['uniform', 'distance']
   All best model choose n_neighbors = 1 and weights = uniform
4. Random Forest Classifier
   Set the Random State at 11 to make it reproducible.

Tuned parameter: ‘max_depth’
‘max_depth’ values are [1, 3, 10, 30, 100]
The best 5 models choose parameter of four 30s and one 10.

Baseline score is the situation I predict all data as no bankrupt. Confusion Matrix is plotted below. The f1_score_macro of which confusion Matrix is 0.4918. This score would by my baseline score.

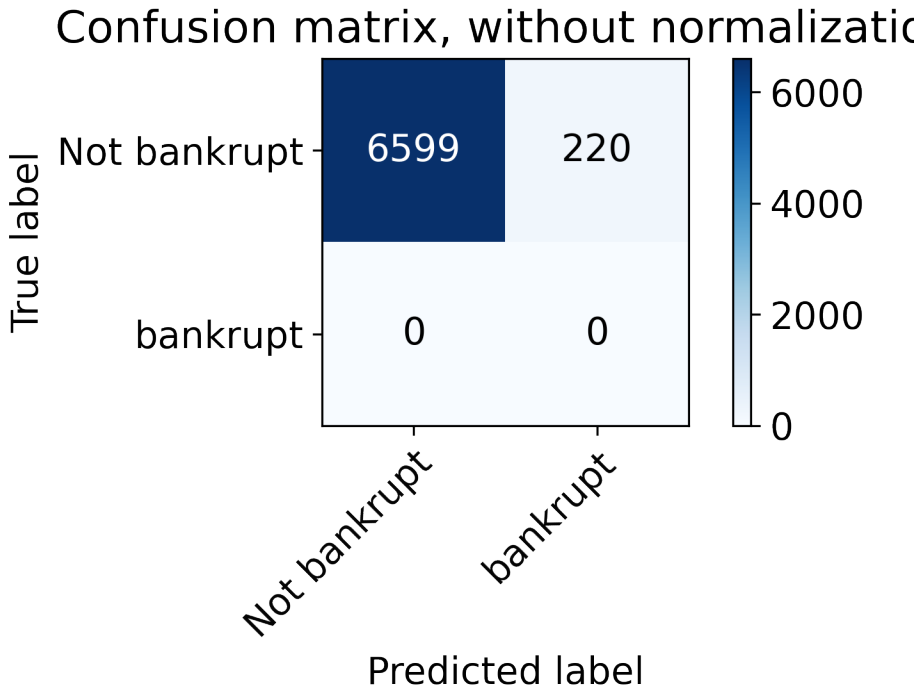

Figure 6 is the confusion matrix of the baseline situation.

Result

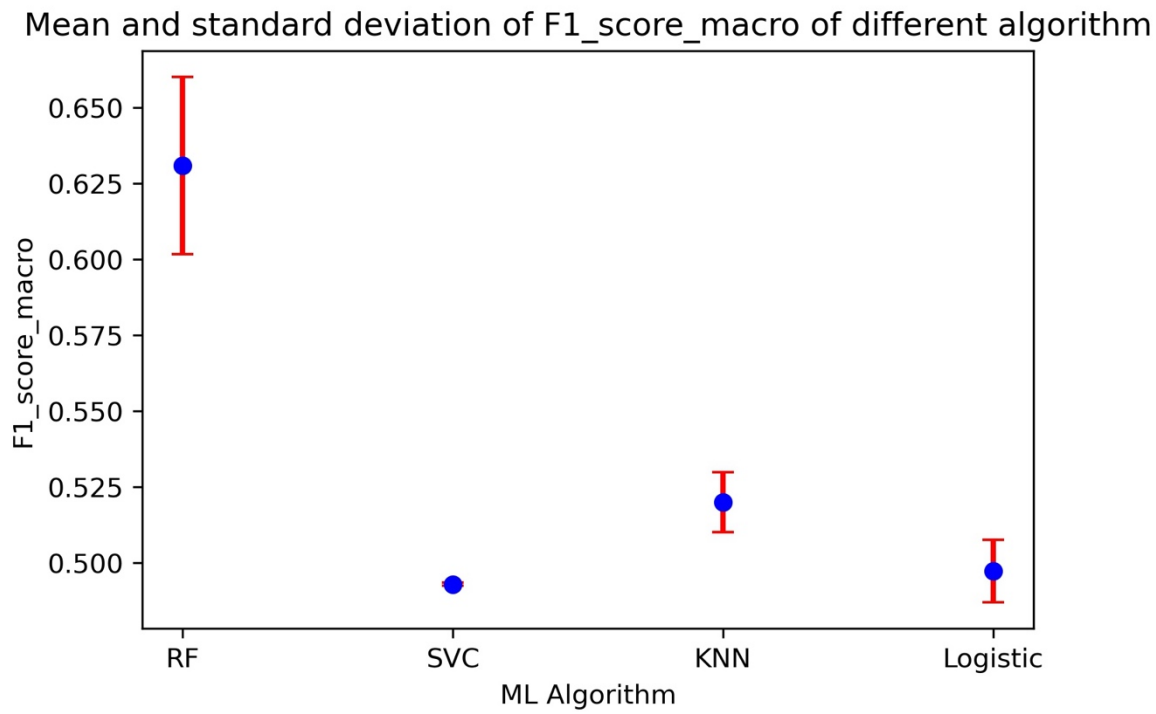

Figure 7 is the figure result, F1_score_macro of four chosen different Machine Learning Algorithms.

```
mean of RandomForest: 0.6308953775242834  std of RandomForest: 0.029119262864157003
mean of svc: 0.49293627087831665  std of svc: 0.0005192008266053108
mean of knn: 0.5199941256142201  std of knn: 0.009839015647903905
mean of Logistic: 0.49729340652119636  std of Logistic: 0.010329744231881375
```

Figure 8 is the quantity result of the performance of four Algorithms.

We can see that Random Forest Classifier is the most predictive model with a score of 0.63 which is higher than the baseline score which is 0.4918. Standard deviation of the scores of Random Forest is 0.029. That means Random Forest Classifier has a roughly 4 standard deviations above the baseline.

Because Random Forest Classification is the most predictive model. I would do more inspection on Random Forest Classification model with the best parameter max_depth = 30. First is the confusion matrix of the best model testing result.

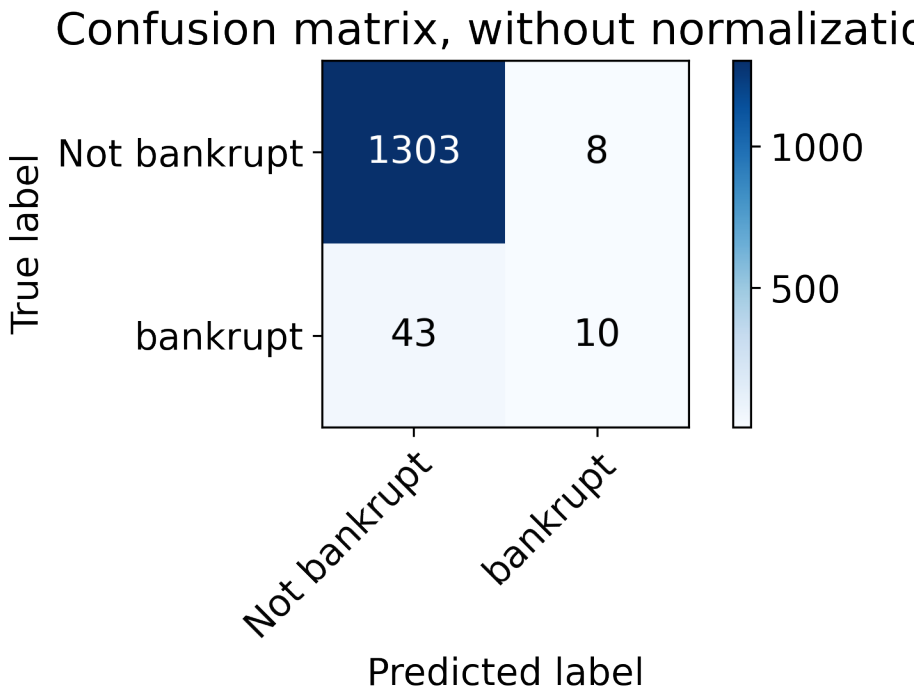

Figure 9 is the confusion matrix of the test set of the best model.

We can see that the best model identifies half (10/18) of the bankrupt companies with the F1_score of 0.63

Then I calculate the global importance of this model and find the five most important and least important features by permutation test using Random Forest Classification.

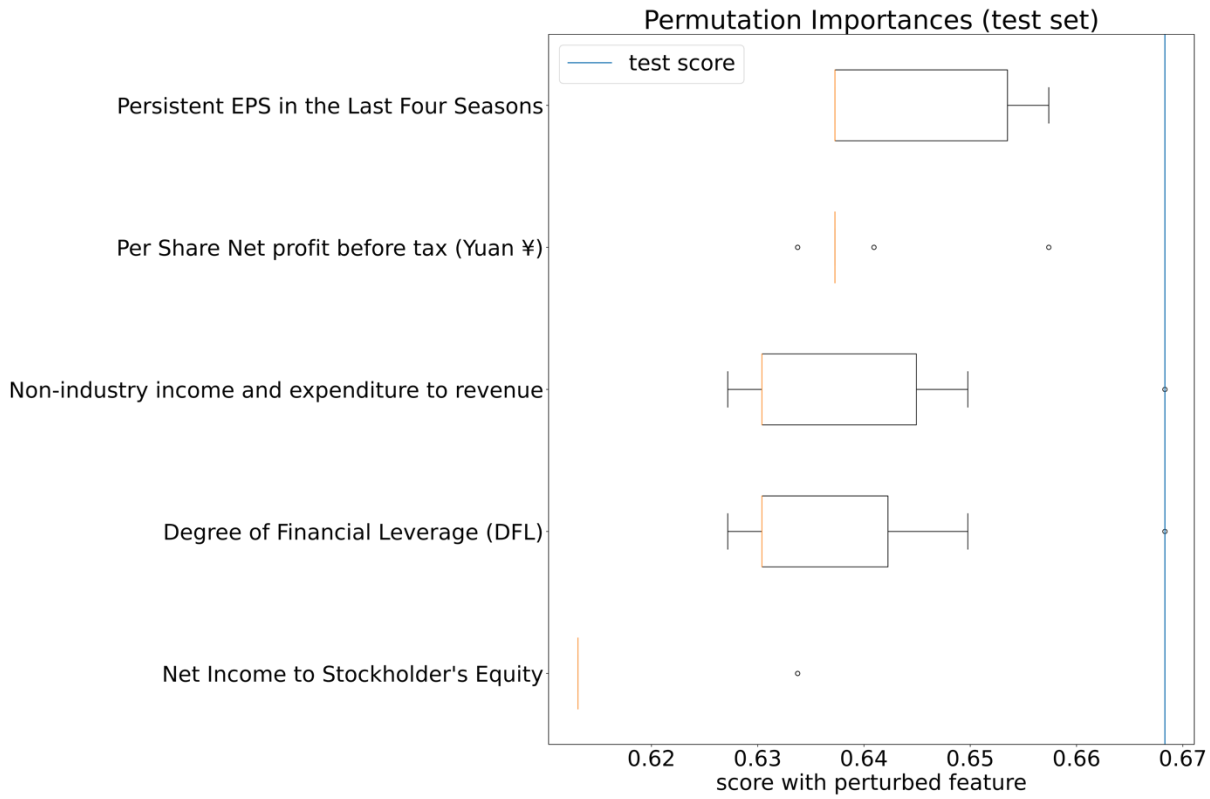

Figure 10 is the five most important features

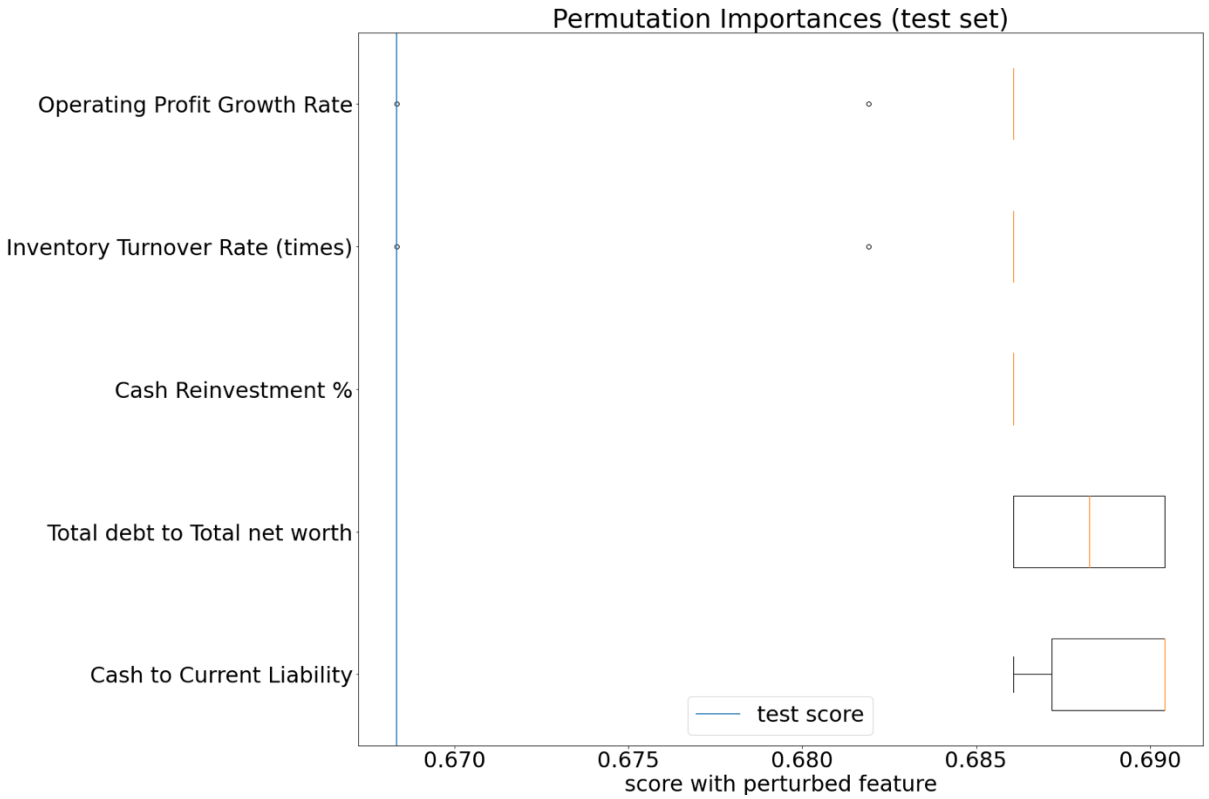

Figure 11 is the five least important features

After find the most and least important feature. I use Sharp to calculate their local importance.

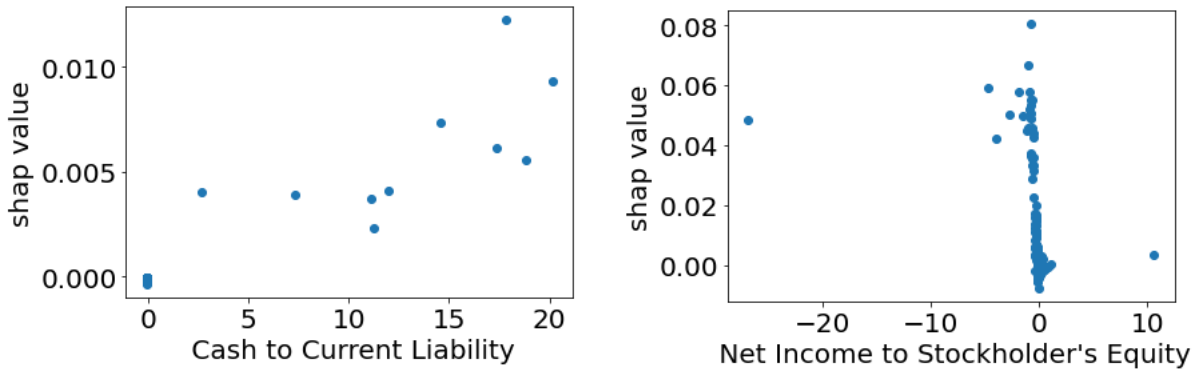

Figure 12 &13 are the local importance of the most and least important feature

We can see that the Cash to Current Liability are positive relationship to the shap value but the Net Income to Stockholder's Equtiy is not, which relationship is not clear. That is the reason that one is most important and the other is not.

Outlook

Because of the limitation of the time and computing ability. Tuning parameter process is not enough especially in SVC. That represents that some of the models are not the optimized model. We can see that the standard deviation of the scores of SVC model in my tuning part is unusually small. The reason of this situation may be that the tuning hyperparameter process fails. I should tune more parameter values which are close to the optimal value.

Second is that the five most important features are not stable. When I change the random state to run the best model. The ranking of the model global importance would change a little bit. I want to use more time on this to identify that it is normal or the problem of my Pipeline.

In addition, the interpretation of local importance analysis of the variables is not enough. I need to focus more on the real-world interpretation of the feature importance.

Reference

1. https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction
2. Previous work. https://www.kaggle.com/marto24/bankruptcy-detection

GitHub Repository

https://github.com/shijiemao/project1030