Company Bankruptcy Prediction

Ting-yuan Lin, Zhuolin Zheng, Nai Li, Chunlan Ma
CUHK Business School (MSc in Business Analytics)
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
11551{48842, 47401, 53423, 51505}@link.cuhk.edu.hk

Professor: Dr. Haiqin Yang
CUHK Business School (MSc in Business Analytics)
The Chinese University of Hong Kong*)*
Shatin, N.T., Hong Kong
haiqin.yang@gmail.com

Abstract

In these two years, many companies have closed due to the outbreak of COVID-19, including some leading corporates around the world, the concern of corporate bankruptcy has been raised again. A better grasp the internal factors is conductive to the timely observation of company's survival and help it adjust the decision in time. In this project, we use machine learning to better explore the information hidden behind the financial indicators, apply seven different machine learning methods to train the prediction model, then compare their performance and choose best one for better predicting.

Keywords: bankruptcy, machine learning, prediction

Introduction

In the news of these two years, many companies change their business strategies or go bankruptcy directly because of the negative impact from the sudden epidemic COVID-19. The black swan incident led variety of enterprises still have to bear high fixed cost without cash inflow, and the inventory get impairment because enterprises can't operate normally. Some companies with meager financial resources have even gone bankruptcy. Face to pandemic situation, how to reduce the losses of enterprises as much as possible and make them survive the dangerous period safely is an important issue for most enterprises.

The first research on the causes of corporate financial distress is the Univariate Discriminant Approach (UDA) developed by Fitzpatrick in 1932 1. However, due to lack of advanced statistical and computational tools at that time, this approach is an empirical analysis and comparison of a series of financial ratios between normal enterprises and bust enterprises. Since the 1960s, some scholars began to introduce various models to predict the probability of company bankruptcy. Nowadays, with the development of computer science and information technology, western researchers also apply machine learning into predicting.

With the increasingly urgent demand for the study of corporate bankruptcy prediction, companies need to establish an effective economic early warning system through a perfect economic prediction method. The trained model can help companies build warning mechanism by referring to their own financial indicators to dynamically monitor the status quo of company. For companies making decision, when a company chooses a partner to cooperate, it can use this model to analyze the financial situation of the other companies which want to build partnership, and estimate the risks that the company needs to bear. From the perspective of investors, realizing the actual present situation for those listed companies that want to invest has a significant impact on the investment decisions of investors, the prediction model provides an effective tool for securities investors and analysts to predict the bankruptcy distress.

The main object of this project is listed companies, and uses machine learning method to analyze the collected internal financial index parameters of the company, then trains a high accuracy model to predict the bankruptcy rate of the company.

I. DATA INTRODUCTION

A. Dataset

The data set comes from Kaggle.com (https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction), the original data is from Taiwan Economic Journal for the years 1999 to 2009. The dependent variable bankruptcy is defined based on the business regulations of the Taiwan Stock Exchange. There are 95 features in total in this data set, including ROA, operating gross margin, realized sales gross margin and so on.

Details of data set are depicted in the following:

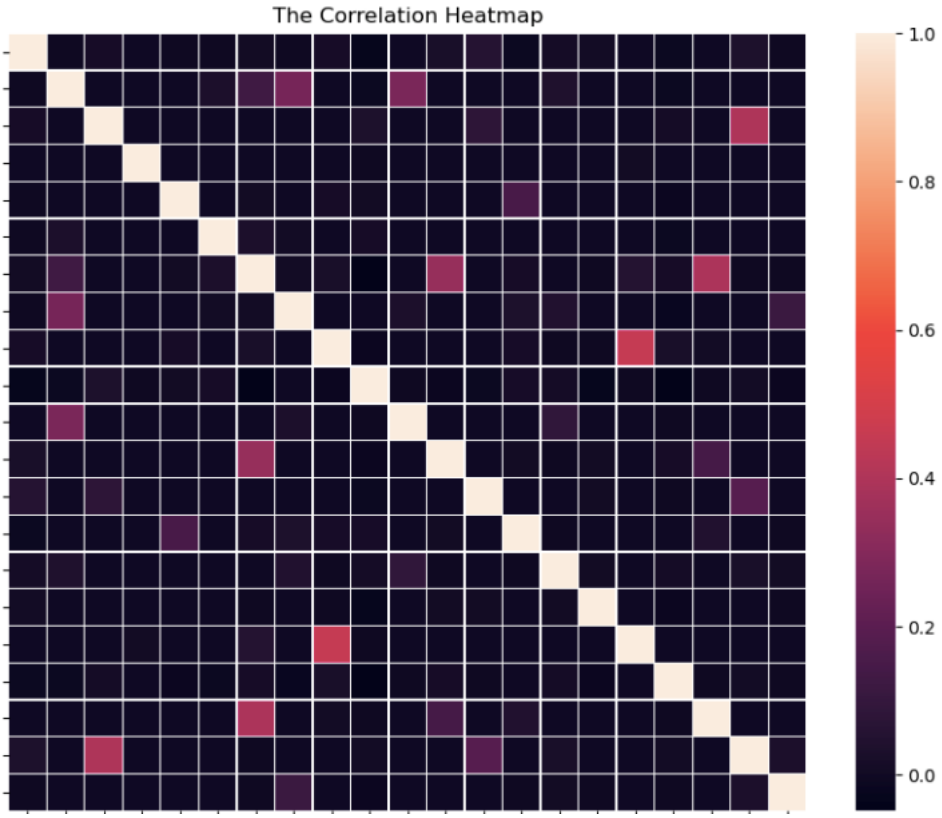
- The data set consists of 6,819 samples, each sample has 95 features and one label.
- The ratio of bankruptcy and non-bankruptcy is 6599:220 (96.77% :3.23% in percentage).
- Most of feature data type is float64.

B. Feature Selection

The features in data set is too much for model training, we first use variance to drop some useless features. We set the threshold at 5 so that we can delete most features with small variances. We treat data in this way because we think features with small variances do not carry to much information we need to train classification model. After this procedure, there are 21 features left. The top 5 largest variance features are current ratio, fixed assets to assets, net value growth rate, revenue per person and quick asset/current liability.

In order to avoid severe correlation problems, we draw correlation matrix and heatmap to check our 21 features. As heatmap shows, there is no severe correlation problems, so we keep all the 21 features. The 21 features are as follows: Interest-bearing debt interest rate, Revenue Per Share (Yuan ¥), Net Value Growth Rate, Current Ratio, Quick Ratio, Total debt/Total net worth, Long-term fund suitability ratio (A), Accounts Receivable Turnover, Average Collection Days, Fixed Assets Turnover Frequency, Revenue per person, Allocation rate per person, Quick Assets/Current Liability, Cash/Current Liability, Inventory/Current Liability, Long-term Liability to Current Assets, Total income/Total expense, Current Asset Turnover Rate, Fixed Assets to Assets, Liability-Assets Flag, Total assets to GNP price.

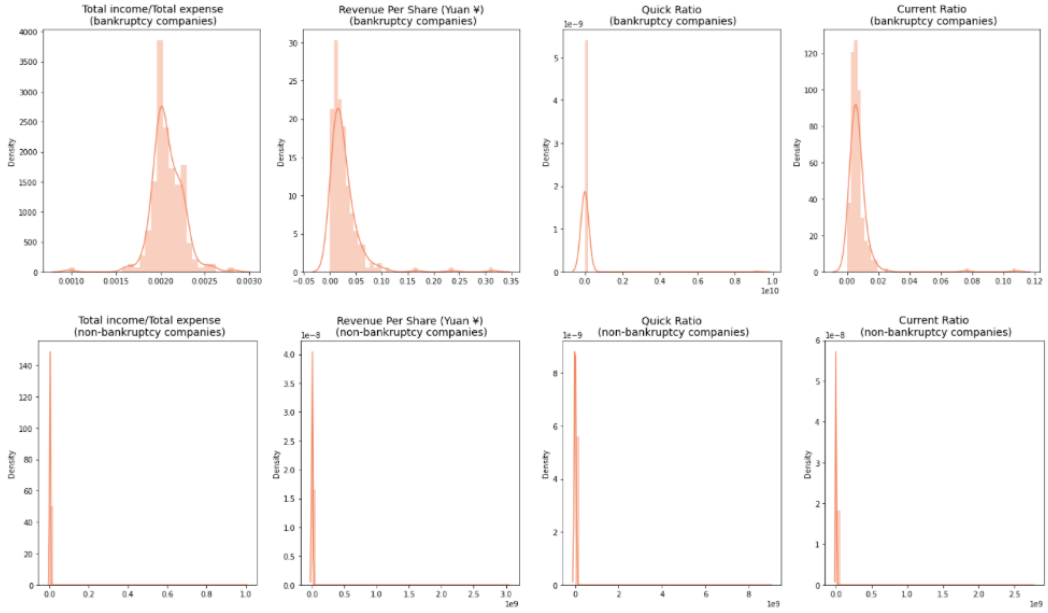
The heatmap result is showed below:



Ⅰ-1 Heatmap for selected 21 features

C. *Visualization*

In the following, we select four features to further draw distribution plot to show the distribution difference between bankrupt companies and non-bankrupt companies.



Ⅰ−2 Features Distribution

The graph shows that the non-bankrupt companies has more centralized data distribution.

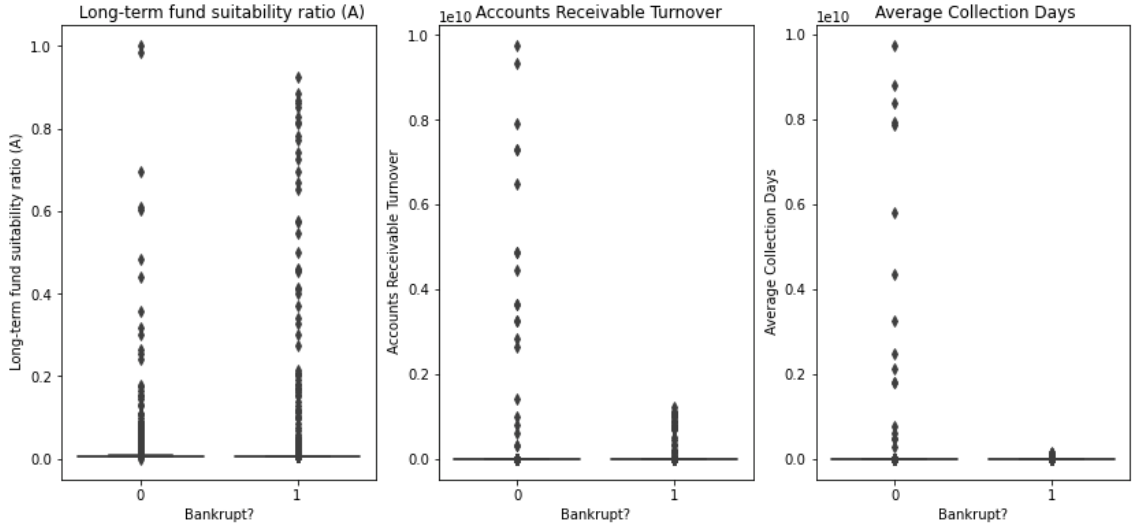
II. DATA PREPROCESSING

A. *Oversampling*

Since the data ratio of bankruptcy and non-bankruptcy is severely unbalance, we need to do some preprocessing work to balance the data. We try SMOTE to balance our data set. After SMOTE, we have 13,198 samples in total and the sample ratio of bankruptcy and non-bankruptcy is 1:1.

B. *Outlier Removal*

In order to avoid the influence of outlier samples, we draw boxplot to detect outlier samples and remove them. Below show parts of boxplot graphs:



II-1 Boxplot

After outlier remove, there are 5,655 non-bankruptcy and 5,544 bankruptcy samples left.

C. *Normalization*

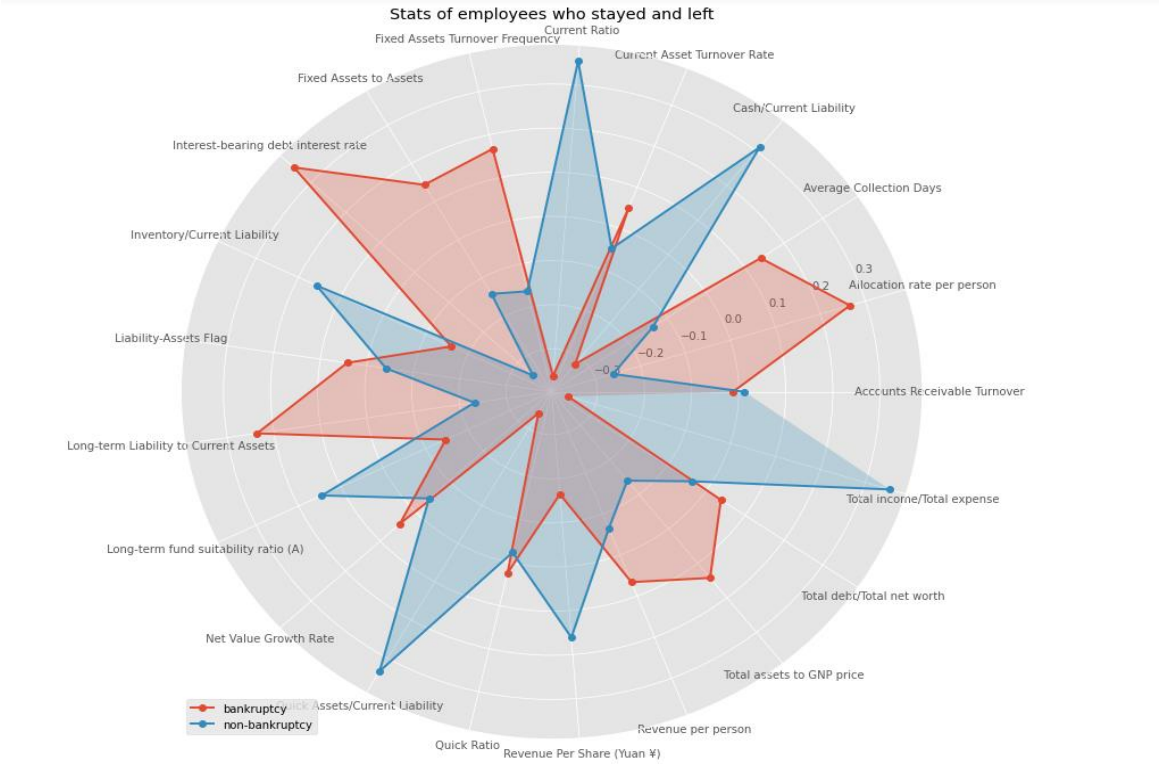
This project includes some distance-based models, so the scaler of features matters a lot, we need to do normalization work to reduce the scaler influence when we train data with distance-based models. The data set after normalization looks as follows:

	Interest-bearing debt interest rate	Revenue Per Share (Yuan ¥)	Net Value Growth Rate	Current Ratio	Quick Ratio	Total debt/Total net worth	Long-term fund suitability ratio (A)	Accounts Receivable Turnover	Average Collection Days	Fixed Assets Turnover Frequency	...	Allocation rate per person	Quick Assets/Current Liability
0	1.224769	-0.454570	-0.030368	-0.859315	-0.027959	-0.046793	-0.503225	0.659880	-1.041193	-0.511443	...	0.901909	-0.657399
1	0.896416	-0.350397	-0.030368	-0.466444	-0.027959	-0.046793	-0.488916	0.100489	-0.690284	-0.265039	...	-0.253107	-0.421861
3	0.062905	-0.547314	-0.030368	-0.656974	-0.027959	-0.046793	-0.493886	0.821243	-1.108063	2.624296	...	0.165349	-0.551209
4	1.060592	-0.102053	-0.030368	-0.465792	-0.027959	-0.046793	-0.387324	0.272610	-0.825281	-0.511443	...	0.289615	-0.406585
5	1.186882	-0.438746	-0.030368	-0.809007	-0.027959	-0.046793	-0.549467	0.356518	-0.881670	2.675702	...	0.701343	-0.768420
...	...	...	...	...	...	...	...	...	...	...	...	...	...
13190	0.231858	-0.546971	-0.030368	0.294634	-0.027959	-0.046793	1.611301	-0.332779	0.686804	-0.511443	...	-0.341598	-0.000307
13191	1.224555	-0.297394	-0.030368	-0.513793	-0.027959	-0.046793	-0.496487	-0.206203	-0.122989	-0.230503	...	0.077466	-0.395428
13194	-0.478934	4.775488	-0.030368	-0.479913	-0.027959	-0.046793	-0.468127	-0.167178	-0.379741	-0.511443	...	-0.378358	-0.467311
13195	0.736432	0.305658	-0.030368	0.025259	-0.027959	-0.046793	0.140697	-0.737445	1.424232	-0.511443	...	0.050434	0.005453
13196	1.502244	3.479182	-0.030368	-0.870032	-0.027959	-0.046793	-0.465254	7.018649	-1.570916	-0.511443	...	0.625516	-0.641806

II-2 Normalized data

D. Radar Chart

In order to have a complete view on data after preprocessing, we draw a radar chart based on the mean value of each features group by bankruptcy or not.
The result shows below:



II-3 Radar Chart

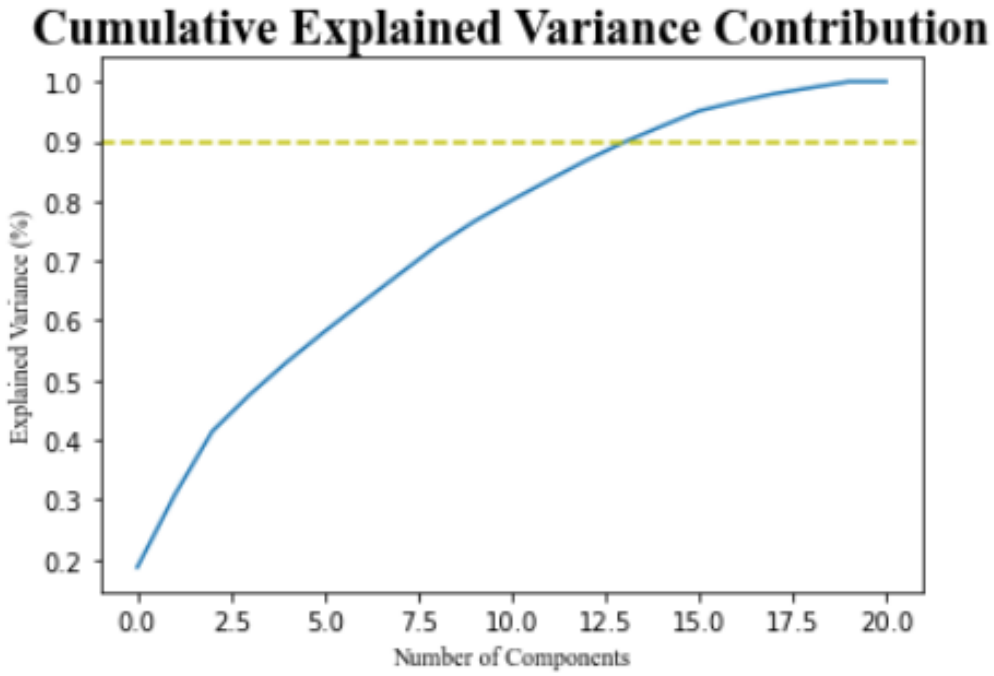
The feature mean of bankrupt companies and non-bankruptcy companies is quite different according to radar chart. Current ratio, Cash/Current Liability, Total income/Total expense and Quick Assets/Current Liability are quite different between two class. In non-bankrupt companies, its value is much larger than in bankrupt companies. In the other side, Interest-bearing debt interest rate, Long-term liability to current assets and allocation rate per person, the mean value of bankrupt companies are much higher than non-bankrupt companies.

III. PRINCIPLE COMPONENT ANALYSIS

A. Determining Components of PCA

Principal Component Analysis (PCA) utilizes linear transformation technique to realize dimensional reduction which preserves essential information from high dimensional space by projecting it onto lower dimensional space.

In this part, we prepare two datasets for model training: the original data and the data after PCA dimension reduction. For PCA part, we first find the explained variance ratio set to be 0.9 to determine the number of components we want, where we can clearly observe n=12 is a good fit.



III−1 Cumulative Explained Variance Contribution curve

B. *Model Evaluation Before PCA*

One keynote to point out about PCA is that it is an unsupervised learning technique, though we can cluster the similar data points based on the feature correlation between them without any labels, we expected the performance to be low because predicting company bankruptcy is considered a supervised learning type. Regardless of the machine learning type, we will train our PCA data and original data with Decision Tree, Random Forest, K-Nearest Neighbors, Naive-bayes, Support Vector Machine, Logistics Regression, Gradient Boosting Classifier, and Neural Network models. Model performance details can be in the attached code section.

```
original_results = []

for name, model in original_models.items():
    result = model.score(X_test, y_test)
    original_results.append(result)
    print(name + ": {:.2f}%".format(result * 100))
```

```
Decision Tree: 92.51%
Random Forest: 96.77%
K-Nearest Neighbors: 92.94%
Naives-bayes: 50.36%
SVM: 92.29%
Logistic Regression: 85.84%
Gradient Boosting: 92.40%
Neural Network: 97.67%
```

III-2 Model Evaluation before PCA

C. *Model Evaluation After PCA*

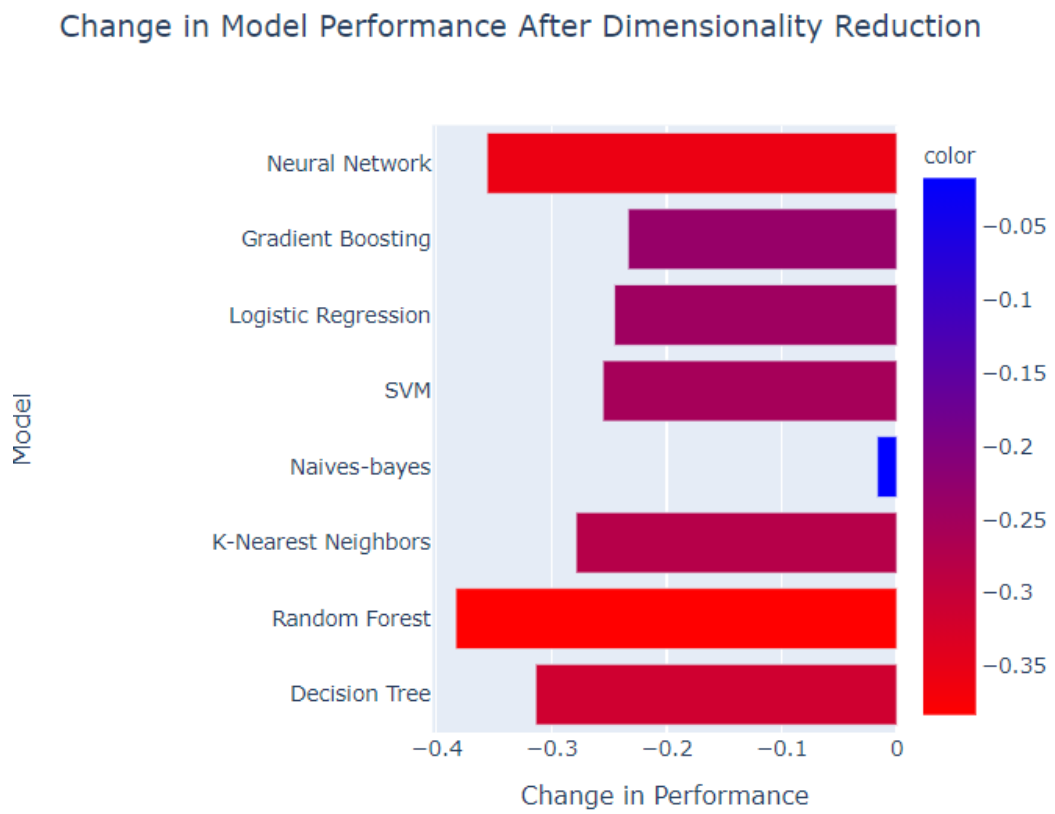
```
reduced_results = []

for name, model in reduced_models.items():
    result = model.score(X_test_pca, y_test)
    reduced_results.append(result)
    print(name + ": {:.2f}%".format(result * 100))
```

```
Decision Tree: 57.31%
Random Forest: 62.01%
K-Nearest Neighbors: 65.27%
Naives-bayes: 54.87%
SVM: 66.27%
Logistic Regression: 66.70%
Gradient Boosting: 63.05%
Neural Network: 64.80%
```

III-3 Model Evaluation after PCA

With not much surprise, PCA data performance even worse compared to the original dataset. The following demonstrates the change in performance of models before and after PCA reduction:



III−4 PCA comparison

From our experimental outcomes, we see the overall performance has decreased. Therefore, we decide to keep using original dataset and train our model for the rest of our lab.

IV. MODELING AND EVALUATION

The problem of company bankruptcy detection is to use some features to predict whether the company will face bankrupt in the future. In this part, we try to find good ways to make predictions. We evaluate model performance by comparing the score. Also, we can get further insights about the ranking of features importance of 21 features after filtering, which can help companies to establish a dynamic indicator monitoring system.

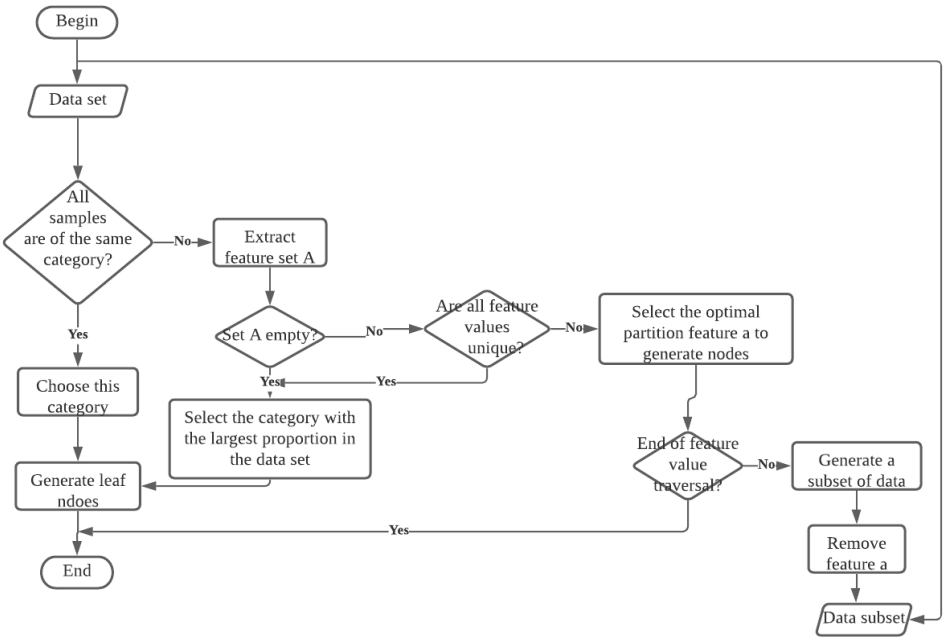
When selecting models, we first consider the supervised learning and unsupervised learning. Because all of our samples have labels, we choose supervised learning methods. The popular models used for binary classification have Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machine, Naive Bayes. Among them, Logistic Regression and Support Vector Machine are well designed for binary classification. Considering the shortcomings of decision trees that are prone to overfitting, Random Forest uses a voting mechanism of multiple decision trees to improve the decision tree. Like Random Forest, Gradient Boosting Decision Tree is also a decision tree model based on ensemble thinking. However, Gradient Boosting Decision Tree uses the boosting algorithm to build a weak learner at each step of the iteration to make up for the deficiencies of the original model. It builds a learner along the fastest direction of gradient descent at each iteration. We also use Random Forest and Gradient Boosting Decision Tree to fit data. For above models, only k-Nearest Neighbors and Support Vector Machine need to use normalized data Because their algorithm involves distance calculation. The rest models algorithm based on probability. So, we use data without standardized processing.

Here, we use GridSearchCV algorithm to find the suitable parameters for each model to make sure the accuracy score is as high as possible, except for Naive Bayes model. Because the code "GaussianNB()" don't have parameters. GridSearchCV contains two parts. One is Grid Search, which loops through all candidate parameters, trying every possibility. But its disadvantage is that it takes time. The other is CV, which means cross validation. We set CV = 5 here, meaning we perform 5 divisions on the original data set, and perform training and evaluation for each division, and finally get the average evaluation results after 5 divisions as the evaluation score. This method will return us the highest score and its corresponding parameters among all the candidate parameters. This can help us reduce fitting or overfitting problems.

A. Decision Tree

The Tree is consisting of root node (the first selection point), non-leaf nodes in branches (intermediate process), leaf node (the final decision result). The generation process of a decision tree is mainly divided into the following three parts: ① Feature selection: feature selection refers to selecting a feature from the many features in the training data as the splitting criterion for the current node. There are many different quantitative evaluation criteria for how to select a feature. ② Decision tree generation: according to the selected feature evaluation criteria, child nodes are generated recursively from top to bottom, and the decision tree stops growing until the data set is indivisible. ③ Pruning: decision trees are prone to overfitting. Generally, pruning is needed to alleviate overfitting. There are two types of pruning techniques: pre-pruning and post-pruning.

We use pre-pruning mothed. According to set several candidate max depth parameters, we get the highest accuracy score and the suitable max_depth = 18.[1] The classification algorithm flow of Decision Tree is as IV-1.

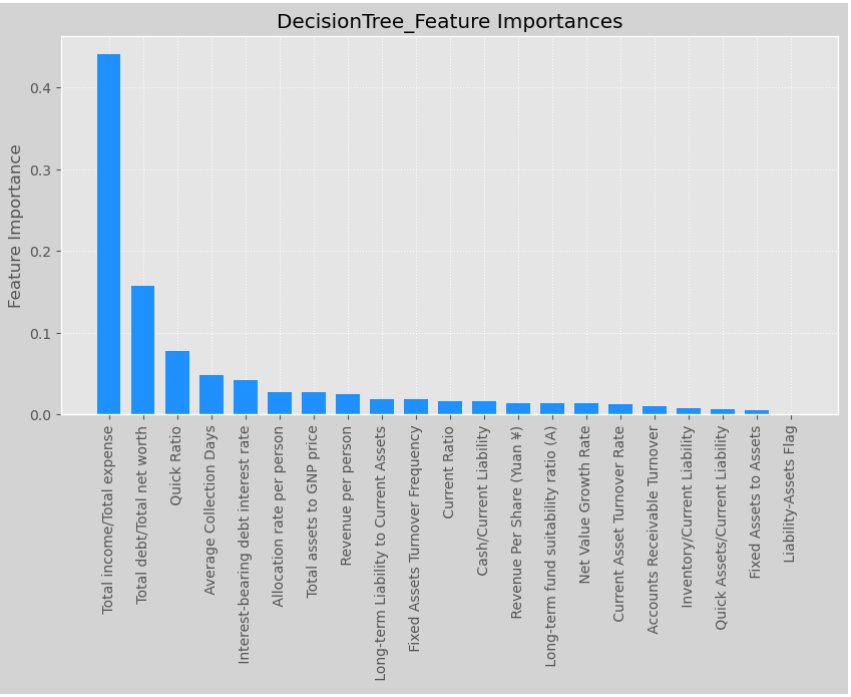


IV-1 Algorithm Flow of Decision Tree

After fitting the data, we get the ranking of feature importance. (IV-2 and IV-3)

feature	importance
Total income/Total expense	0.441139
Total debt/Total net worth	0.157993
Quick Ratio	0.077972
Average Collection Days	0.047917
Interest-bearing debt interest rate	0.041890
Allocation rate per person	0.027712
Total assets to GNP price	0.027596
Revenue per person	0.024303
Long-term Liability to Current Assets	0.019025
Fixed Assets Turnover Frequency	0.018140
Current Ratio	0.016272
Cash/Current Liability	0.015910
Revenue Per Share (Yuan ¥)	0.014284
Long-term fund suitability ratio (A)	0.014156
Net Value Growth Rate	0.013615
Current Asset Turnover Rate	0.012073
Accounts Receivable Turnover	0.010449
Inventory/Current Liability	0.007855
Quick Assets/Current Liability	0.006409
Fixed Assets to Assets	0.005291
Liability-Assets Flag	0.000000

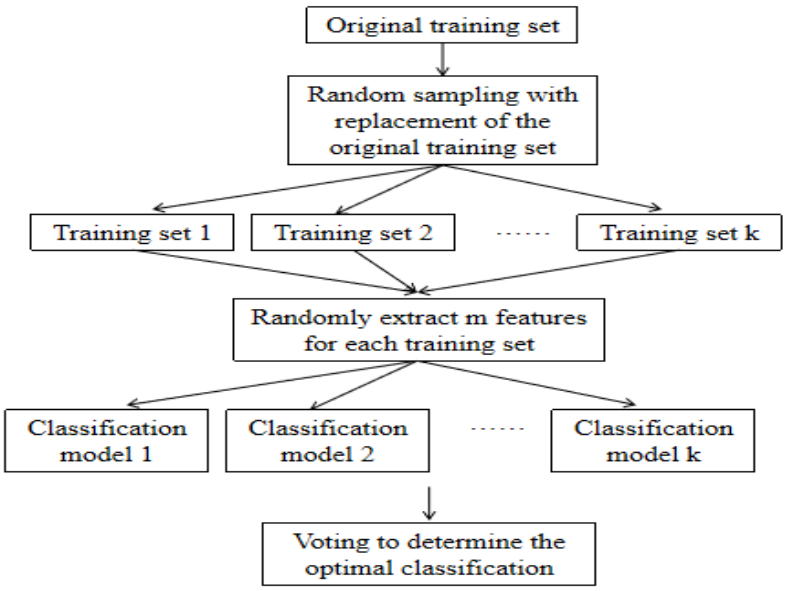
IV-2 Feature Ranking(DT)



IV-3 Feature Ranking Visualization(DT)

B. *Random Forest*

Random forest is a combination of decision tree and bagging algorithm. It is a kind of cluster classification model. It builds a forest in a random manner. The forest is composed of many decision trees, and each decision tree is independent. After the random forest model is obtained, each decision tree in the random forest is judged separately when a new sample comes in. Bagging method usually uses voting for classification problems. And the category or one of the categories that gets the most votes is the final model output. The classification algorithm flow of Random Forest is as Figure 4. According to set several candidate max_depth and n_estimators parameters, we get the highest accuracy score and the max_depth = 21 and n_estimators = 40.[1]

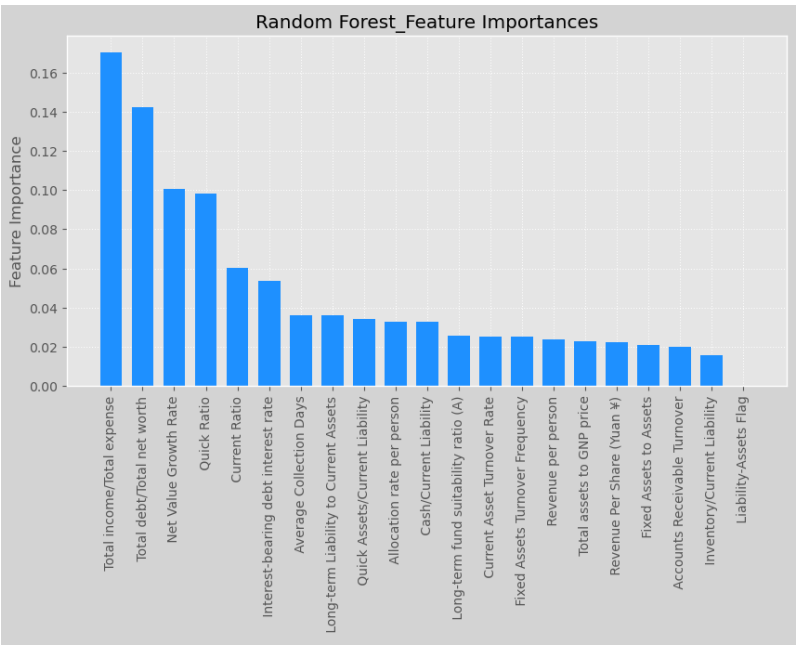


IV-4 Algorithm Flow of Random Forest

After fitting the data, we get the ranking of feature importance. (IV-5 and IV-6)

feature	importance
Total income/Total expense	0.170345
Total debt/Total net worth	0.142102
Net Value Growth Rate	0.100713
Quick Ratio	0.098431
Current Ratio	0.060310
Interest-bearing debt interest rate	0.053653
Average Collection Days	0.036117
Long-term Liability to Current Assets	0.036017
Quick Assets/Current Liability	0.034301
Allocation rate per person	0.032925
Cash/Current Liability	0.032648
Long-term fund suitability ratio (A)	0.025771
Current Asset Turnover Rate	0.025301
Fixed Assets Turnover Frequency	0.025101
Revenue per person	0.023925
Total assets to GNP price	0.022912
Revenue Per Share (Yuan ¥)	0.022384
Fixed Assets to Assets	0.020884
Accounts Receivable Turnover	0.020217
Inventory/Current Liability	0.015867
Liability-Assets Flag	0.000077

IV-5 Feature Ranking(RF)



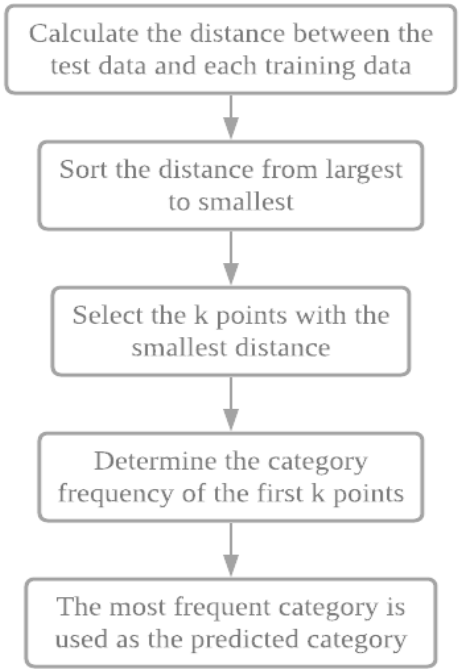
IV-6 Feature Ranking Visualization(RF)

C. *k-Nearest Neighbors*

k-Nearest Neighbors algorithm is a famous statistical method for pattern recognition. An instance has K most similar (ie nearest neighbors in the feature space) instances in the feature space. Most of these instances belong to a certain category, and the instance also belongs to this category. The selected neighbors are all instances that have been correctly classified. For distance measurement, we have many distance measurement methods. But the most commonly used is Euclidean distance, that is, for two n-dimensional vectors x and y, the Euclidean distance between the two is defined as:

$$D(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots\ldots+(x_n-y_n)^2}$$

The classification algorithm flow of k-Nearest Neighbors is as Figure 3. According to set several candidate n_neighbors and weights parameters, we get the highest accuracy score and the n_neighbors = 2 and weights = uniform.



IV-7 Algorithm Flow of k-Nearest Neighbors

D. Naive Bayes Classifier

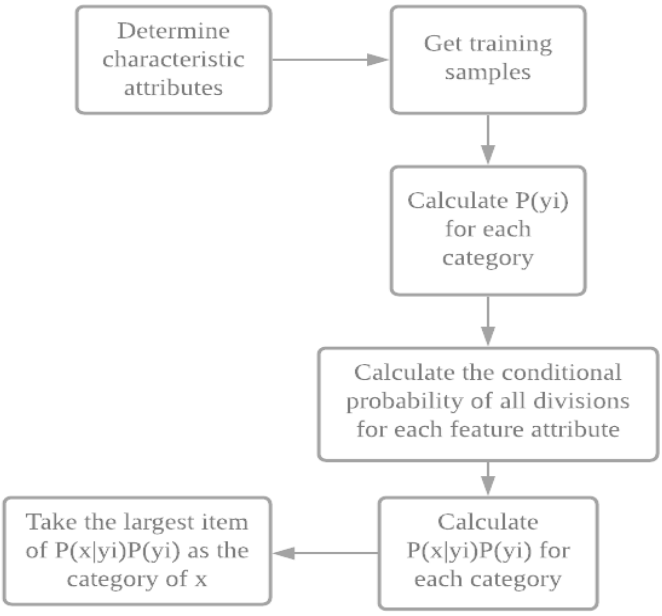
Naive Bayes Classifier is a generative probabilistic method. The naive Bayes classifier uses the "attribute conditional independence hypothesis", that is, for known categories, all attributes are assumed to be independent of each other. Then Bayesian formula is:

$$P(c \mid x) = \frac{P(c)P(x \mid c)}{P(x)} = \frac{P(c)}{P(x)} \prod_{i=1}^{d} P(x_i \mid c)$$

Where d represents the number of attributes, xi represents the value of x on the i-th attribute. Because P(x) is uniquely determined by the sample set, that is, P(x) is the same for all categories, so Naive Bayesian classification Expression:

$$h_{nb}(x) = \arg\max_{c \in y} P(c) \prod_{i=1}^{d} P(x_i \mid c)$$

The classification algorithm flow of Naive Bayes Classifier is as IV-8.

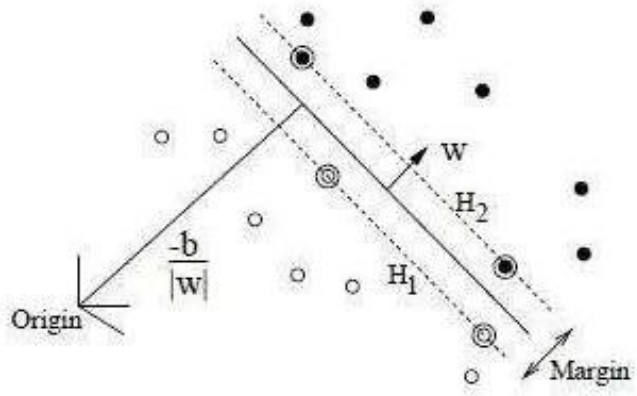


IV- 8 Algorithm Flow of Naive Bayes Classifier

We use cross validation to calculate the accuracy score instead of GridSearchCV since there is no parameters in our python function module.

E. Support Vector Machine

SVM is to find a hyperplane for given training samples to separate as many positive and negative examples as possible. The choice of the optimal hyperplane is based on positive and negative examples as far as possible from this hyperplane. From the below figure, if we can ensure that the points closest to the hyperplane are as far away as possible from the hyperplane, we can ensure that all positive and negative examples are as far as possible from the hyperplane. Therefore, we define these points closest to the hyperplane as support vectors (as the points crossed by the dotted line in the figure above). And define the distance between the positive and negative support vectors as Margin.[2]



IV- 9 Support Vector Machine model

According to set several candidate C and tol parameters, we get the highest accuracy score and the C= 3 and tol = 0.06.

F. Logistic Regression

Logistic regression simply adds certain features to the Sigmoid function as input values. Output the result of two classifications through the Sigmoid function. The advantage of logistic regression is that the amount of calculation is not large, and it is easy to understand and implement. The disadvantage is that it is easy to cause under-fitting and the accuracy of classification is not high.

The Sigmoid function is a common sigmoid function in biology, also called sigmoid growth curve. The Sigmoid function is defined by the following formula:

$$S(x) = \frac{1}{1+e^{-x}}$$

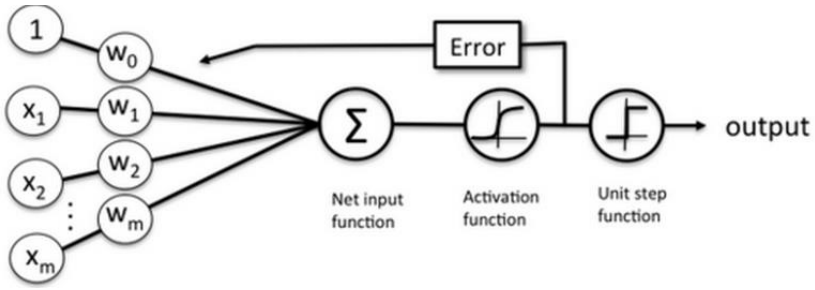
Sigmoid function is symmetrical according to the center of 0.5, only the data whose output is greater than 0.5 is regarded as "Class 1", and the data less than 0.5 is regarded as "Class 0", thus the problem of two classifications is realized. Each feature is multiplied by a regression coefficient, and then all the result values are added together to define the sigmoid function input as z.

$$z = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots\ldots + w_n x_n$$

Where xi is features and wn is parameters by training models. So the form of Logistic regression model can be written as:

$$h_w(x) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$$

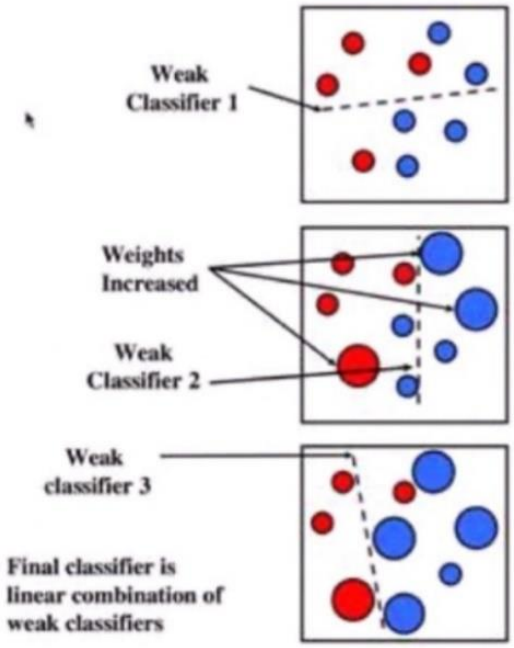
Logistic regression model is as figure 5. According to set several candidate C and tol parameters, we get the highest accuracy score and the C= 0.01 and tol = 0.0001.



IV-10 Logistic regression model

G. Gradient Boosting Decision Tree

Gradient Boosting Decision Tree is an iterative decision tree algorithm, which consists of multiple decision trees, and the conclusions of all trees are added together to make the final answer. The boosting is a bias-reduction technique. Like the below figure, it will increase the weight of last bias and to improve next weak classifier.

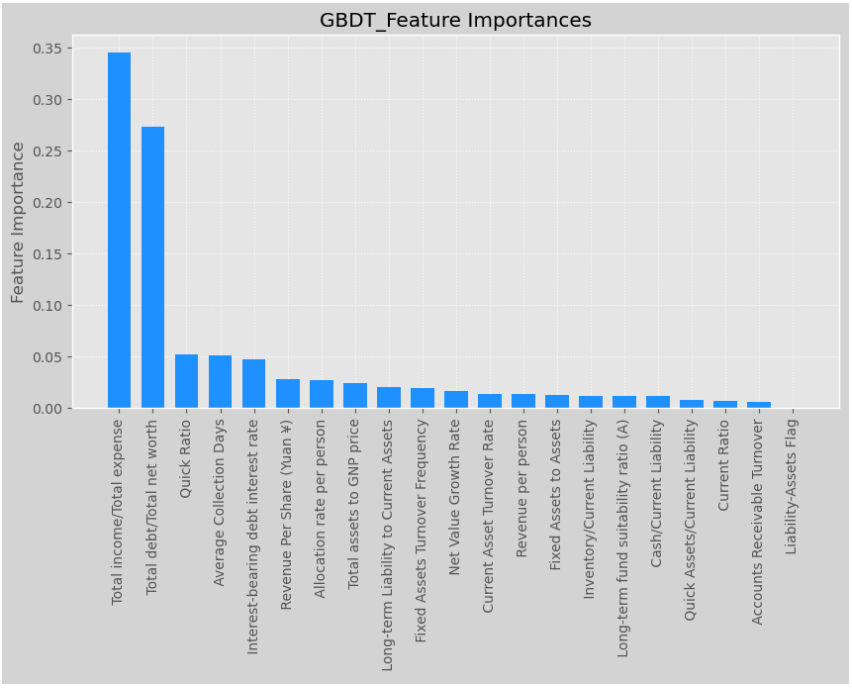


IV-11 Boosting Technique

According to set several candidate max_depth and min_samples_split parameters, we get the highest accuracy score and the max_depth= 17 and min_samples_split= 200. After fitting the data, we get the ranking of feature importance. (IV-12 and IV-13)

feature	importance
Total income/Total expense	3.455333e-01
Total debt/Total net worth	2.727681e-01
Quick Ratio	5.233018e-02
Average Collection Days	5.086298e-02
Interest-bearing debt interest rate	4.720710e-02
Revenue Per Share (Yuan ¥)	2.801534e-02
Allocation rate per person	2.689127e-02
Total assets to GNP price	2.436190e-02
Long-term Liability to Current Assets	1.996961e-02
Fixed Assets Turnover Frequency	1.936301e-02
Net Value Growth Rate	1.687648e-02
Current Asset Turnover Rate	1.366934e-02
Revenue per person	1.358829e-02
Fixed Assets to Assets	1.295351e-02
Inventory/Current Liability	1.198445e-02
Long-term fund suitability ratio (A)	1.142355e-02
Cash/Current Liability	1.134321e-02
Quick Assets/Current Liability	7.576905e-03
Current Ratio	6.995074e-03
Accounts Receivable Turnover	6.286261e-03
Liability-Assets Flag	1.850156e-07

IV-12 Feature Ranking (GBDT)



IV-13 Feature Ranking Visualization(GBDT)

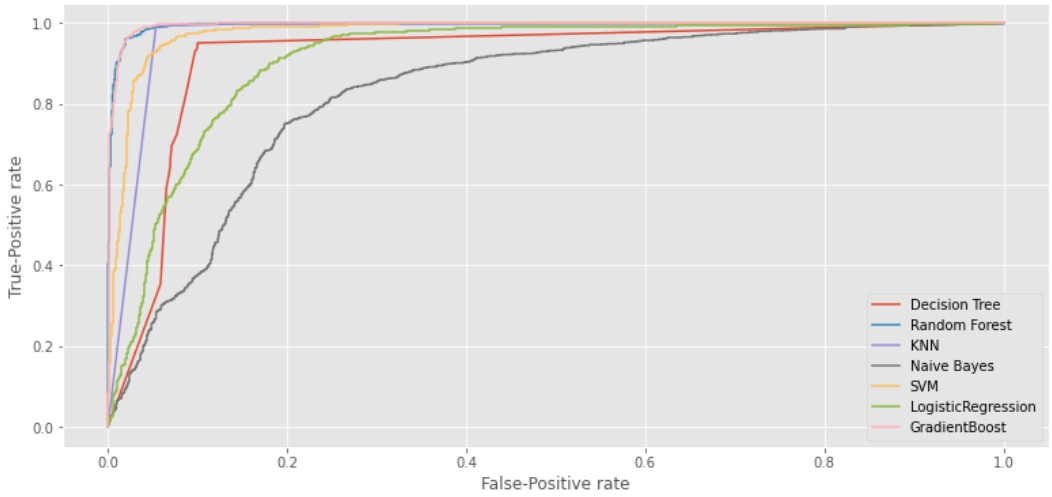
From the feature importance ranking from Decision Tree, Random Forest and Gradient Boosting Decision Tree, we can know that total income/total expense, total debt/total net worth, quick ratio, average collection days, interest-bearing debt interest rate, allocation rate per person are the most importance features that companies need to care about.

The suitable parameters and accuracy score are summary as below:

Model	Accuracy Score	Selected Parameters
Decision Tree	0.9403	max_depth = 18
Random Forest	0.9689	max_depth = 21 and n_estimators = 40
k-Nearest Neighbors	0.9475	n_neighbors = 2 and weights = uniform
Naive Bayes Classifier	0.5118	--
Support Vector Machine	0.9262	C = 3, tol = 0.06
Logistic Regression	0.4859	C = 0.01, tol = 0.0001
Gradient Boosting Decision Tree	0.9624	max_depth = 17, min_samples_split = 200

H. Receiver Operating Characteristic Curve

To better evaluate the performance of each model, we draw the ROC curve of them.



IV-14 ROC curve comparison

Combine the summary table and ROC curve, we get the conclusion that Gradient Boosting Decision Tree, Random Forest are the best 2 models to predict whether the company will face bankrupt in the future.

V. Hypothesis Testing For Model Result

After modeling and evaluating several classification models, we try to use hypothesis test for Gradient Boosting Classifier and Random Forest models result, the two models that perform the best overall. We performed 10 rounds of 10-fold cross-validation on the two models. The cross-validation score of Decision Tree and Random Forest Models are shown as followed.

Gradient Boosting Decision Tree:

```
from sklearn.model_selection import cross_val_score
gradientboost_new = GradientBoostingClassifier(max_depth = 16,min_samples_split=200)
CV_score = cross_val_score(gradientboost_new,X_smote, y_smote,cv=10)
print("The cross-validation score is ",CV_score)
print("The mean score is ",CV_score.mean())
```

```
The cross-validation score is  [0.96363636 0.95984848 0.95606061 0.95757576 0.97954545 0.98939394
 0.98863636 0.98257576 0.9909022  0.9833207 ]
The mean score is  0.9751495623406162
```

V-1 Gradient Boosting Decision Tree

Random Forest:

```
rfc_new = RandomForestClassifier(max_depth = 17, n_estimators = 42)
RF_CV_score = cross_val_score(rfc_new,X_smote, y_smote,cv=10)
print("The cross-validation score is ",RF_CV_score)
print("The mean score is ",RF_CV_score.mean())
```

```
The cross-validation score is  [0.95151515 0.95454545 0.95984848 0.96136364 0.9
7651515 0.98636364
 0.98484848 0.98106061 0.9893859  0.9787718 ]
The mean score is  0.972421830128426
```

V-2 Random Forest

Hypothesis test is conducted at significant level of 5%, and we are trying to figure out if there exists significant difference between the two models on predicting bankruptcy, specifically which model is better in predicting. So set the null hypothesis and alternative hypothesis: H0: the cross-validation score of Gradient Boosting Decision Tree is greater than or equal to the score of Random Forest. The alternative assumes the opposite. We want to see if Random Forest model has statistical evidence that are better than Gradient Boosting Classifier. At the end, we did negate the null hypothesis and conclude Random Forest model performs better.

VI. Conclusion

In our paper, we have exhaustively trained various models with some meaningful approaches. We choose several models to predict bankruptcy risk. To compare the power of these models, we conduct Receiver Operating Characteristics curve and calculate the Area Under the Curve (AUC). The larger the Area Under the Curve, the better the prediction power of the model. It turns out that for all firms, Gradient Boosting Classifier and Random Forest models have the best prediction power. On the other hand, the Naïve model is less effective in predicting company bankruptcy. Our data, extracted from an economical journal for the years 1999–2009 on Kaggle, covers around seven thousand records of representing company bankruptcy based on some business regulations features. Among them, around 66 hundred companies were in good position while about 220 companies eventually went bankrupt.

Since pervious financial crisis and current Covid-19 situation, default risk prediction becomes more and more important for financial institutions and company owners. We adopt two most prevalent and predictive models including Random Forest and Decision Tree and answer the question: which models have the best power in forecasting.

VII. REFERENCE

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

	SID	Full Name	Class	Role	Job description
1	1155148842	LIN, Ting-yuan	DSME6650BA	Team Leader	Hypothesis Testing for Model Results, Data Preprocessing, Modeling training, Feature selection (25%)
2	1155147401	ZHENG, Zhuolin	DSME6650BA	Member	Model selection, PCA, Comparison of the models (25%)
3	1155153423	LI, Nai	DSME6650BA	Member	Abstract, Introduction, Task definition (25%)
4	1155151505	MA, Chunlan	DSME6650BA	Member	Pipeline, Data Introduction, Feature selection, Conclusion, Visualization (25%)