Company Bankruptcy Prediction

Wenchenghao Pian

May 11, 2021

Contents

1 Introduction

The finance and accounting data is crucial for a company, stockholders, and investors. It is the best way to observe the performance of a company. We can observe four aspects through those data: Debt solvency indicator, Operational capability, Profitability, and the primary inspection of development capability indicators. Those are the best measurements of the performance of a company. This project is an excellent chance to understand how those finance figures affect a firm. This project aims to build an artificial intelligence neural network model to supervise the learning data set so that it can predict which companies will go bankrupt and which financial indicators will affect the company.

Google Colab Notebook: `https://colab.research.google.com/drive/1nUSOK2_LxGm_ss1mY-4TRSk9wRSneH7F?usp=sharing`.

2 Data Analysis

The dataset "Company Bankruptcy Prediction " from Kaggle. The dataset can be found at https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction. The data source is the "Taiwan Economic Journal" from 1999 to 2009. The definition of company bankruptcy in the data is based on the regulations of the Taiwan Stock Exchange.[4] The dataset has 6819 rows and 96 columns. It contains:

- 95 features
- 1 Vector of labels

The dataset's input features are ratio and rate can measure financial performance such as operating profit rate, cost, expense profit rate, surplus cash guarantee multiple, return on total assets, return on net assets, and capital return.

```
Number of Example (rows): 6819
Number of Features (columns): 96
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6819 entries, 0 to 6818
Data columns (total 96 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   Bankrupt?                                               6819 non-null   int64
 1    ROA(C) before interest and depreciation before interest 6819 non-null   float64
 2    ROA(A) before interest and % after tax                 6819 non-null   float64
 3    ROA(B) before interest and depreciation after tax      6819 non-null   float64
 4    Operating Gross Margin                                 6819 non-null   float64
 5    Realized Sales Gross Margin                            6819 non-null   float64
 6    Operating Profit Rate                                  6819 non-null   float64
 7    Pre-tax net Interest Rate                              6819 non-null   float64
 8    After-tax net Interest Rate                            6819 non-null   float64
 9    Non-industry income and expenditure/revenue            6819 non-null   float64
 10   Continuous interest rate (after tax)                   6819 non-null   float64
 11   Operating Expense Rate                                 6819 non-null   float64
 12   Research and development expense rate                  6819 non-null   float64
 13   Cash flow rate                                         6819 non-null   float64
 14   Interest-bearing debt interest rate                    6819 non-null   float64
 15   Tax rate (A)                                           6819 non-null   float64
 16   Net Value Per Share (B)                                6819 non-null   float64
 17   Net Value Per Share (A)                                6819 non-null   float64
 18   Net Value Per Share (C)                                6819 non-null   float64
 19   Persistent EPS in the Last Four Seasons                6819 non-null   float64
 20   Cash Flow Per Share                                    6819 non-null   float64
 21   Revenue Per Share (Yuan ¥)                             6819 non-null   float64
```

Figure 1: raw and colums

Figure 2 enormously informative for us: There are no missing values (Nan) among the data. Further consideration needs to be done on the pos sibility to have duplicates in our data. It shows that we do not have duplicates in our dataset.

```
# Checking Nan presence

#dataset.isna().sum().max()
[print(col) for col in dataset if dataset[col].isna().sum() > 0]
```

```
[]
```

```
# Check the presence of duplicates
dataset.duplicated().sum()
```

```
0
```

Figure 2: check Nan

2.1 Data Visualization

Data visualization allows us to understand better and analyze data and build a platform for us to observe data. Figure 3 shows the first five rows of data. We can have an idea of our data.

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	Continuous interest rate (after tax)	Operating Expense Rate	Research and development expense rate	Cash flow rate
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809	0.302646	0.780985	1.256969e-04	0.0	0.458143
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301	0.303556	0.781506	2.897851e-04	0.0	0.461867
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388	0.302035	0.780284	2.361297e-04	25500000.0	0.458521
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966	0.303350	0.781241	1.078888e-04	0.0	0.465705
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304	0.303475	0.781550	7.890000e+09	0.0	0.462746

5 rows × 96 columns

Figure 3: first five raw of data

Figure 4 provides us with variance, mean, maximum, minimum, standard deviation, and median. In this way, we can understand the limits of the data, the degree of dispersion, and the degree of deviation.

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	Continuous interest rate (after tax)	Operating Expense Rate
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6.819000e+03
mean	0.032263	0.505180	0.558625	0.553589	0.607948	0.607929	0.998755	0.797190	0.809084	0.303623	0.781381	1.995347e+09
std	0.176710	0.060686	0.065620	0.061595	0.016934	0.016916	0.013010	0.012869	0.013601	0.011163	0.012679	3.237684e+09
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	0.303466	0.781567	1.566874e-04
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	0.303525	0.781635	2.777589e-04
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	0.303585	0.781735	4.145000e+09
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	9.990000e+09

Figure 4: variance, mean, maximum, minimum, standard deviation, and median

Figure 5 visualizes the distributions of each input feature. We can observe that many inputs in the plot are single values. However, if you carefully observe the complete dataset, you will find no single value in the dataset. Those inputs that seem to be single values are some highly approximate values. So it will look like a single value in the plot.

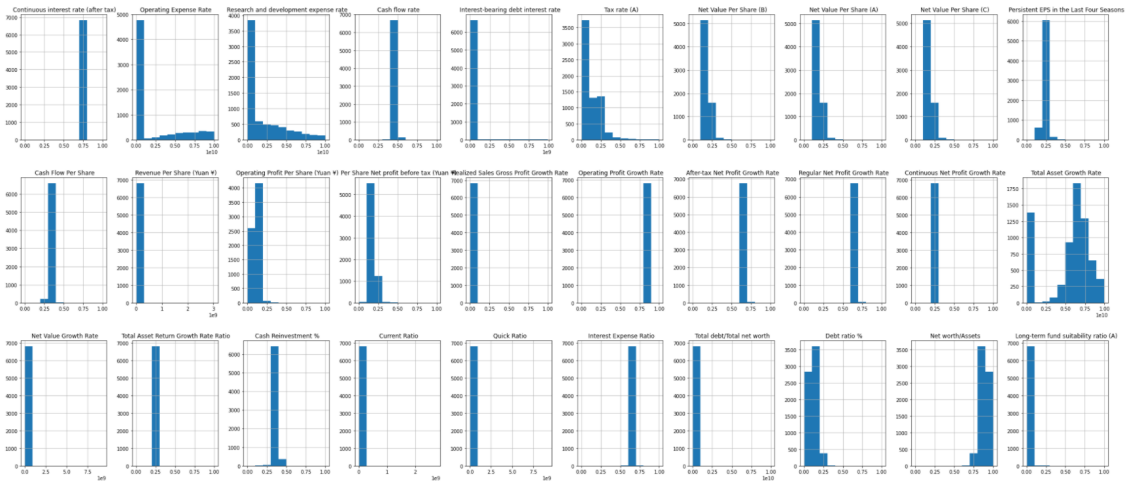

Figure 5: distributions of each input feature

2.2 Distribution of the Output Labels

From the figure below, we can see the dataset is imbalanced which means the model will have high baseline accuracy.

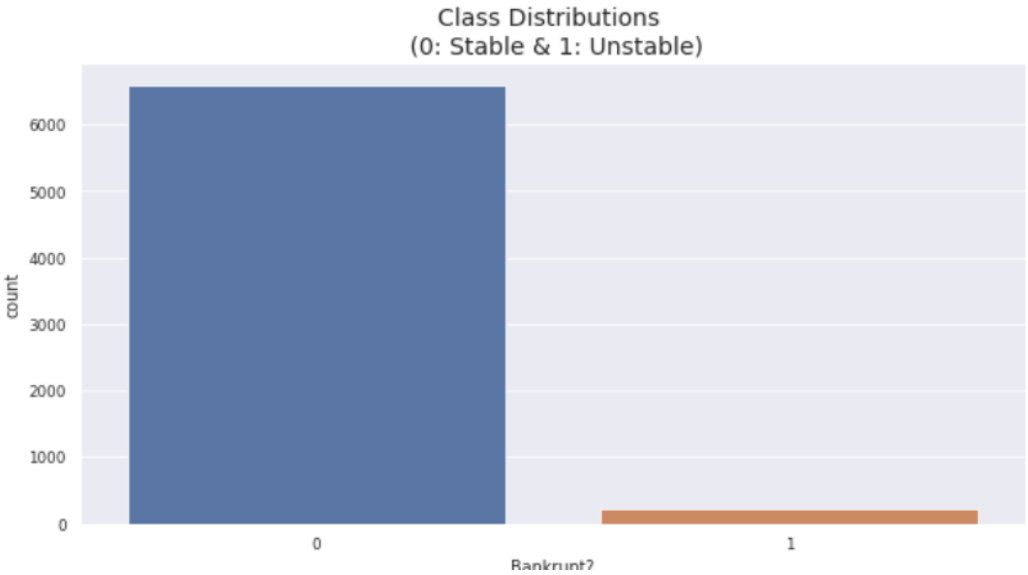

Figure 6: Distribution of the output labels

2.3 Split and Normalization

In regression problems and some machine learning algorithms and training neural networks, it is usually necessary to perform Zero-centered or Mean-subtraction processing and Standardization or Normalization processing on the original data. The difference between normalization and standardization is that normalization is to convert the eigenvalues of the sample to the same dimension and map the data to the interval [0,1] or [-1, 1], which is only determined by the extreme value of the variable. Because the interval scaling method is a kind of normalization, standardization is to process data following the columns of the feature matrix, which is converted to a standard normal distribution by finding the z-score related to the overall sample distribution. Each sample point can have an impact on the standardization. Their similarity is that they can cancel the errors caused by the different dimensions; they are both linear transformations. Both compress the vector X according to the proportion and then perform translation. It can be said that standardization is a kind of normalization. In this project, I apply the standardization for the dataset. [1]

The data was shuffled and splitting into 80 % training set and 20 % validation set for the model training. Data splitting can verify whether the model you have made is correct. Segmentation is to divide the data randomly. For example, 80 % of training data, 20 % of test data, part of the data is used to train the model, and part of the data is used to verify the correctness of the model.

3 Modeling

First, I divided the data into three models to train linear regression, logistic regression, and binary classification using NN. Because of the imbalance of the dataset, the baseline accuracy is very high, which is 97%

3.1 Linear Regression

Models use the minor square function called linear regression equation to model the relationship between independent and dependent variables. A method to determine the quantitative relationship between two or more variables.

The one layer and one neuron's linear model I trained have 99.78% accuracy, MSE is 0.064. Because my baseline accuracy is very high so my linear regression model's accuracy is very high. Figure 7 is the best linear model's learning curve

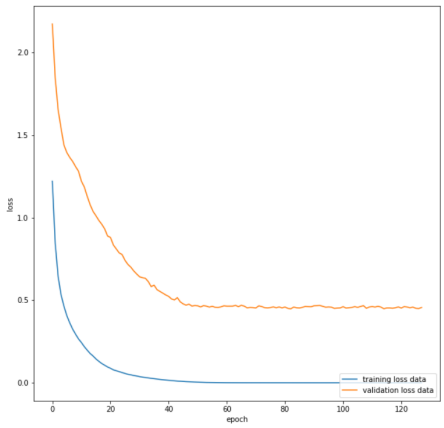

Figure 7: linear regression

3.2 Logistic Regression

Logistic regression is similar to linear regression, and it is suitable for situations where the dependent variable is not a numerical word. For example, a yes or no response. Although it is called regression, it is based on classification based on regression, which divides the dependent variables into two categories. As mentioned above, logistic regression is used to predict the output of two categories. For example, if a credit card company builds a model to decide whether to apply for a credit card issued by a customer, it will predict whether the customer's credit card will default.

Logistic regression is more fit for this dataset because the bankruptcy column is either 0 or 1. The one layer and one neuron model's accuracy of the logistic model is the same as the linear model, which is 99.78%

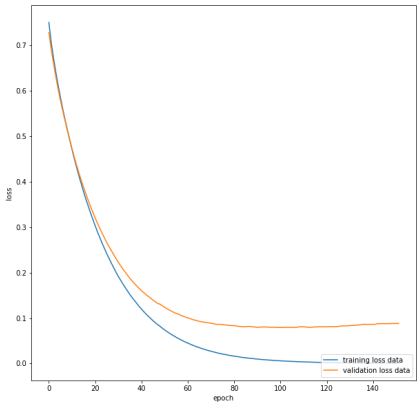

Figure 8: logistic regression

3.3 Binary Classification using NN

Binary classification using NN is the same as logistic(instance of only one layer, binary classification using NN multiplying layers using the relu and sigmoid activation function). However, it should be more accurate than a basic logistic regression model. This is because a neural network model has more weights and biases to learn the patterns in the data.

In the experiment, there are a total of 6 models of 0-5 layers. The two-layer model has the highest accuracy. Nevertheless, combined with the learning curve, the most accurate model should be a 3-layer model because it has a fitting learning curve than a 2-layer model.

	Accuracy	Precision	Recall	F1-score
0 layer	99.78%	93.02%	100.00%	0.96
1 layer	99.78%	93.02%	100.00%	0.96
2 layer	99.78%	95.12%	97.50%	0.96
3 layer	99.71%	90.91%	100.00%	0.95
4 layer	99.78%	97.44%	95.00%	0.96
5 layer	97.07%	0.00%	0.00%	0

Table 1: different layers

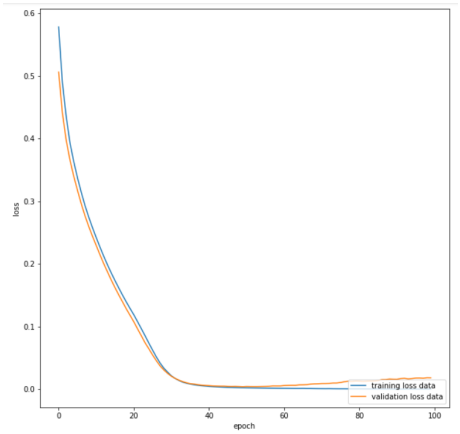

Figure 9: 1 layer loss

It can be observed that the more layer a binary classification model have the learning curve become more fit.

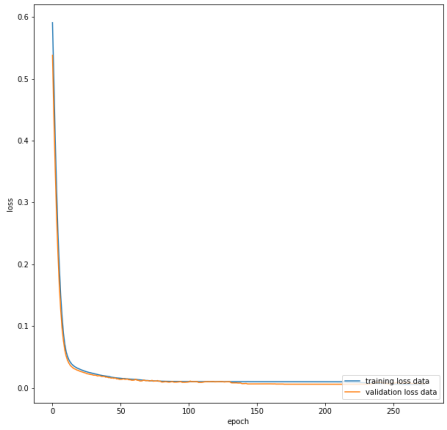

Figure 10: 2 layer loss

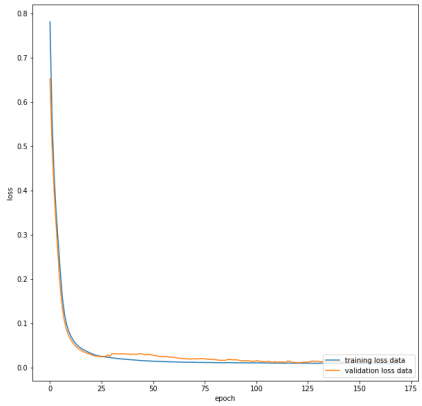

Figure 11: 3 layer loss

The best model of using binary classification using NN is that the model has three layers, and the accuracy is 99.71%. After finding the best model in binary classification using NN, I compared different numbers of neurons to observe the impact on the model. Then I found out that the best model of using binary classification using NN is the model has three layers with five neurons in the first layer, and the accuracy is 99.93%.

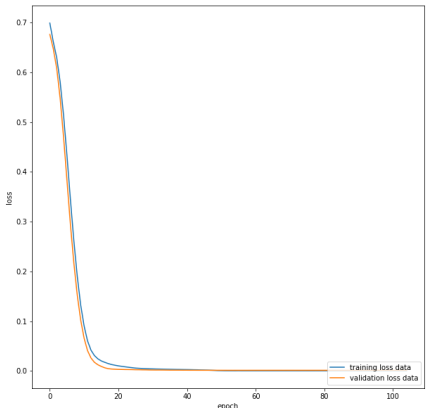

Figure 12: best model

numbers of neurons	Accuracy	Precision	Recall	F1-score
4-3-2-1	99.71%	90.91%	100.00%	0.95
5-4-3-1	99.93%	100.00%	97.50%	0.99
6-4-3-1	99.85%	95.24%	100.00%	0.98
5-3-2-1	99.71%	92.86%	97.50%	0.95

Table 2: different neurons

Also, it can be seen that increasing the neuron can improve the accuracy, but from the learning curve, it can be seen in the learning curve that if the neurons in the first layer increased to 6, and the model will overfitting.

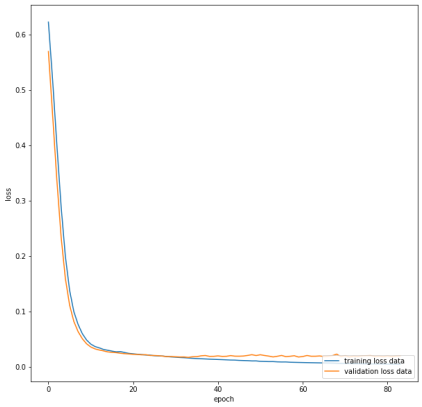

Figure 13: 4 layers overfitting

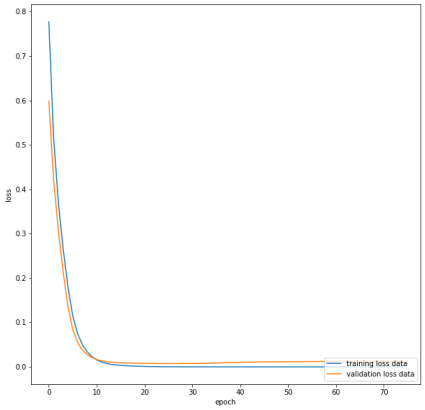

Figure 14: 6 neurons over fitting

3.4 Overfitting

The data of the model is highly imbalanced. The baseline accuracy is also high. So it is accessible to over-fit during the training, so I use the EarlyStopping method to help me avoid overfitting. As a result, when I do not use the EarlyStopping method to stop the training, overfitting will occur. For example, under the control of EarlyStopping, the best model will stop training at epoch 175. However, if I do not use EarlyStopping, let the model continue to the 512 epoch, the overfit will occur. Although the accuracy is increased to almost 100%, the learning curve is not fit as before.

Moreover, the best model has a 3-layer neural network. If one more layer to four layers is added, overfitting will occur. Alternatively, increasing the number of neurons in the first layers to 6 will also cause overfitting.

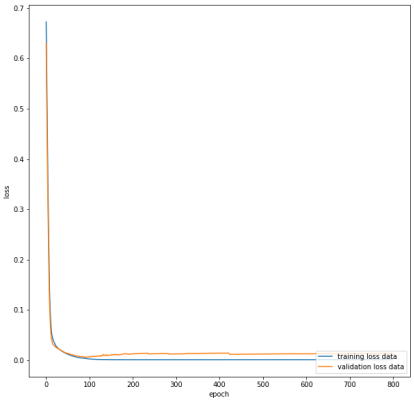

Figure 15: without EarlyStopping

3.5 The Best Neural Network Architecture the Performance difference when Linear Activation is Used Instead of Sigmoid

By replacing the linear activation instead of sigmoid, the accuracy is a decline from 99.93% to 99.71%. Also, the learning curve becomes unreasonable.

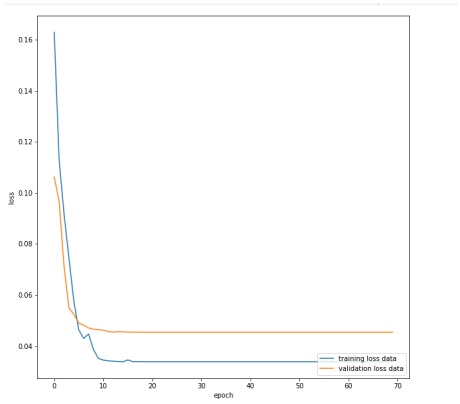

Figure 16: the best neural network architecture the performance difference when linear activation is used instead of sigmoid

4 Model Evaluation

Generally, accuracy, precision, recall, and F-1 value are used to evaluate a model.

1. Accuracy: Accuracy is the part that the model predicts correctly. When the data set is unbalanced, when the number of positive samples and negative samples is significantly different, the accuracy of the model alone cannot be used to evaluate the model's performance. Precision and recall are better indicators for measuring unbalanced data sets.

2. Precision: precision refers to the effect of the degree of correctness of the prediction as a positive example in all categories where the prediction is positive. The higher the precision, the better.

3. Recall: The recall refers to the classification samples that are correctly predicted to be accurate and that are not correctly predicted but are true. The recall rate refers to the degree of correctness of the prediction. Also called sensitivity or true rate TPR. The higher the recall rate, the better.

4. F-1 : It is usually practical to combine the accuracy and recall rate into one index F-1 value, especially when you need a simple method to measure the performance of two classifiers. The F-1 value is the harmonic average of precision and recall. The expected average value treats all values equally, while the average harmonic value gives higher weight to lower values, thereby punishing extreme values more. Therefore, if the accuracy and recall are both high, the classifier will get a high F-1.

The table below compares the different models

	Accuracy	Precision	Recall	F-1
Linear	99.78%	96.08%	98.00%	0.97
Logistic	99.78%	93.02%	100.00%	0.96
Best model of Binary Classification	99.93%	100.00%	97.50%	0.99
Overfitting	99.85%	100.00%	95.00%	0.97
Contrast	99.71%	97.37%	92.50%	0.95

Table 3: Model Evaluation

Combining the comparison table and the learning curve, it can be found that the model's performance is the best in the case of binary classification using NN.

5 Feature Importance

Feature importance and selection are a crucial link for deep learning and machine learning. Even if the algorithms and models have achieved the perfect results, they still cannot achieve maximum accuracy without good data and features. Feature importance and selection aim to maximize the extraction of features from the original data for use by algorithms and models.

Now that the best model has been found, what we need to do now is to use the Recursive Feature Elimination (RFE) technique to remove redundant or insignificant input features.

The following figure shows the importance of each feature to the model. It can be found that almost all the features are on the same horizontal line, and there is no prominent feature. Only three features are not on the same horizontal line as other features.

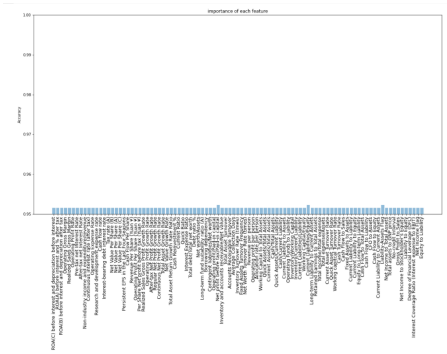

Figure 17: importance of each feature

Then according to the requirements of the Recursive Feature Elimination (RFE)

algorithm, removing less important features one at a time, the graph presented that all features are on the same level.

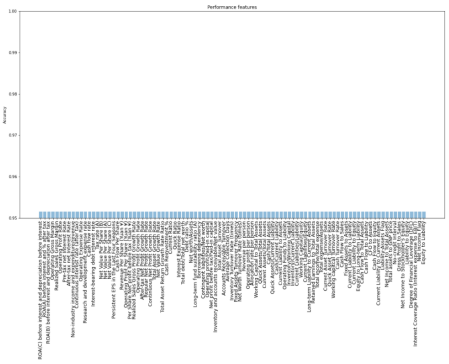

Figure 18: removing less important features one at a time

5.1 OversamplingSMOTE

In phase 3 I found out that features of the dataset are not obvious. Most of the features are at the same level in the plot. The reason why it happens is that the dataset is imbalanced cause the baseline accuracy is high. To solve this I use the oversampling technique to balance the dataset.

Oversampling uses an algorithm to artificially generate a part of the data when the original data is unevenly distributed and randomly sample the newly generated data. The number of the minority set in the original data is finally the same as the majority set.

- For each sample X in the minority class, use the Euclidean distance as the standard to calculate its distance to all samples in the minority class sample set to obtain its K nearest neighbors.[2]
- Set a sampling ratio according to the sample imbalance ratio to determine the sampling magnification N. For each minority sample X, randomly select several samples from its K nearest neighbors, assuming that the selected nearest neighbor is X n.[2]
- Each randomly selected neighbor X n constructs a new sample with the original sample according to the following formula. [2]

$$\chi new = \chi + rand\,(0,1) \times (\widetilde{\chi} - \chi)$$

In fact, for each minority sample a, randomly select a sample b from its nearest neighbors, and then randomly select a point on the line between a and b as the newly synthesized minority sample. As a result, it shows in figure 19 the dataset is balanced after oversampling.[2]

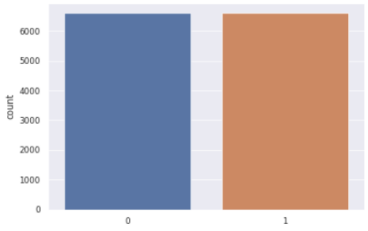

Figure 19: balanced dataset

5.2 Removing Features with Low Variance and RFE

This kind of situation where most of the features are equally important is not common. I doubt the accuracy of the RFE algorithm I wrote, so I used other method and the RFE method provide in sklearn to verify the accuracy. Assuming that the feature value of a feature is only 0 and 1, and in all input samples, 95% of the instances have the feature value 1, then it can be considered that this feature has little effect. If 100% is 1, then this feature is meaningless. This method can only be used when the eigenvalues are all discrete variables. If it is a continuous variable, you need to discretize the continuous variable before it can be used. It can be used as the pre-processing of feature selection. First, remove those features with small value changes and select the appropriate feature selection method from the feature selection methods mentioned next for other feature selection. After removing features with the low variance method, 74 features are removed. There are 22 features left.[3]

5.3 The Performance of Best Model After RFE

After using the RFE method in sklearn and oversampling, I found that the Optimal number of features is 22. Run the best model finds that the accuracy is decrease. Because after oversampling the baseline accuracy will became much lower then before.

	Accuracy	Precision	Recall	F1-score
with RFE	96.48%	52.78%	38.00%	0.44
without RFE	99.93%	100.00%	97.50%	0.99

Table 4: best model after RFE

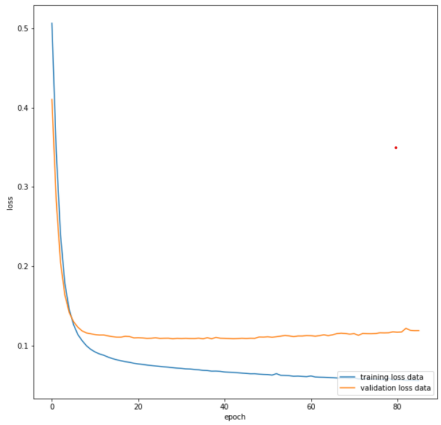

Figure 20: learning curve of the model with RFE

5.4 Ensemble Learning

Ensemble Learning is to build multiple learners and then combine them through a particular strategy to complete the learning task. It is often possible to obtain a significantly superior learner than single learning. According to the relationship between individual learners, they are divided into Bagging, Boosting, and Stacking. In the integrated algorithm, Adaboost, GBDT, and XGBoost belong to the method of boosting, and Random Forest belongs to bagging.

The Bagging algorithm is to randomly sample from the sample with replacement to form N training samples. Each piece trains a model and obtains the result by voting (classification problem) or averaging (regression problem) for the problem to be solved. Each model is independent of the other and can be calculated in parallel.

Boosting algorithm is that the training of the latter model depends on the result of the previous model, and the models need to be generated sequentially. Adaboost and (GBDT, XGBoost) processing methods are slightly different: Adaboost adjusts the sample weight according to the error rate (increase the weight of the wrong sample, reduce the weight of the sample that competes for the correct classification), and join the weak classifier for fusion; GBDT and XGBoost, through Continuously reduce the residual, by constantly adding new trees, to establish a new model in the direction of reducing the residual (negative gradient).

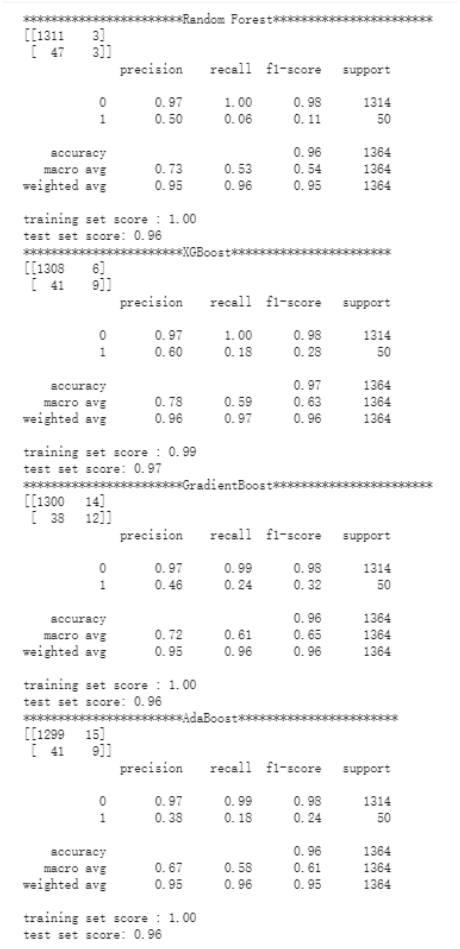

Figure 21: Performance of each algorithms

Whether comparing these four algorithms from various values or ROC curves, the algorithm with the best performance is XGBoost. Because XGBoost draws on a random forest algorithm to sample samples and features, it reduces the amount of calculation while reducing over-fitting. And you can customize the loss function. There is also a normalized regular term so that the trained model is not easy to overfit

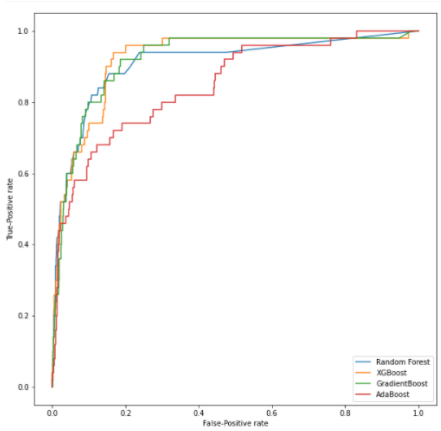

Figure 22: ROC curve of each algorithms

Surprisingly, the plot importance method provided in the XGboost library can use t sort the importance of features. This will make up for the failure of my own method that does not derive the importance of features.

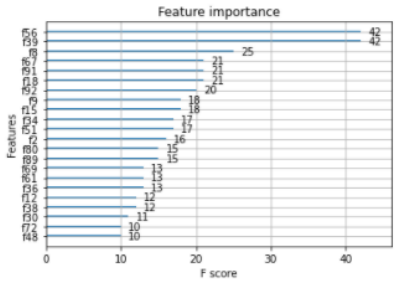

Figure 23: features importance of XGBoost

6 Conclusion

Throughout the project, seven algorithms were used to build neural networks to predict which companies would go bankrupt. Compared with the algorithms, XGBoost is the best performer. It can be concluded that perhaps using over-sampling (or SMOTE) + Regularization model (such as XGBoost) may be more suitable for unbalanced data. However, each model has much room for improvements, such as more parameter adjustments and more data processing. Data enhancement or data synthesis can be performed on the data set because the selected data set is a small and unbalanced data set. But that is a little out of scope for this course, so I don't have time to learn these. I hope I can learn more about artificial intelligence in my future studies or work.

In learning artificial intelligence and doing this project during this semester, I discovered that data is the necessary foundation for building artificial intelligence. Data largely determines the performance and fairness of artificial intelligence. If this bankrupt data set can be more accurate and more prominent, the model's performance will become better.[1]

References

[1] Standardization and normalization,
https://www.programmersought.com/article/23175226881/

[2] Algorithm to deal with sample imbalance problem: SMOTE,
https://www.programmersought.com/article/60703449008/

[3] Sklearn —— Feature Selection,
https://www.programmersought.com/article/21347086501/

[4] Knuth: Computers and Typesetting,
https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction