IBM Data Science Certification Capstone Project –
Predicting Bankruptcy among Tiawanese Companies, 1998-2009

Fred Addy

February 12, 2021

1. Introduction

Being able to predict the likelihood of a company's imminent bankruptcy would provide value to investors, company management, and employees. Investors, because they can choose to sell their stake before losing it (if they are equities-focused) or increase the company's bond yields (if they are fixed-income-focused). Management, because not all parties in the C-suite or on the Board of Directors may be aware of the dire financial financial straits the company finds itself in. And employees, because getting an early warning that their company may soon be insolvent may allow them to exit and find a new job before they are fired.

Bankruptcy is tricky to predict because, while some eagle-eyed investors may be able to spot the warning signs early, the majority of people will not have access to the same data as financial professionals. As a result, our stakeholder would like to build a model that can predict bankruptcy using only readily-available financial datapoints. To this end, we will utilize a financial dataset with far too many features so that we will be able to slim them down to the most relevant ones. Furthermore, we will train our model using only the features that can be easily found in regular company filings, such as earnings reports.

Note:

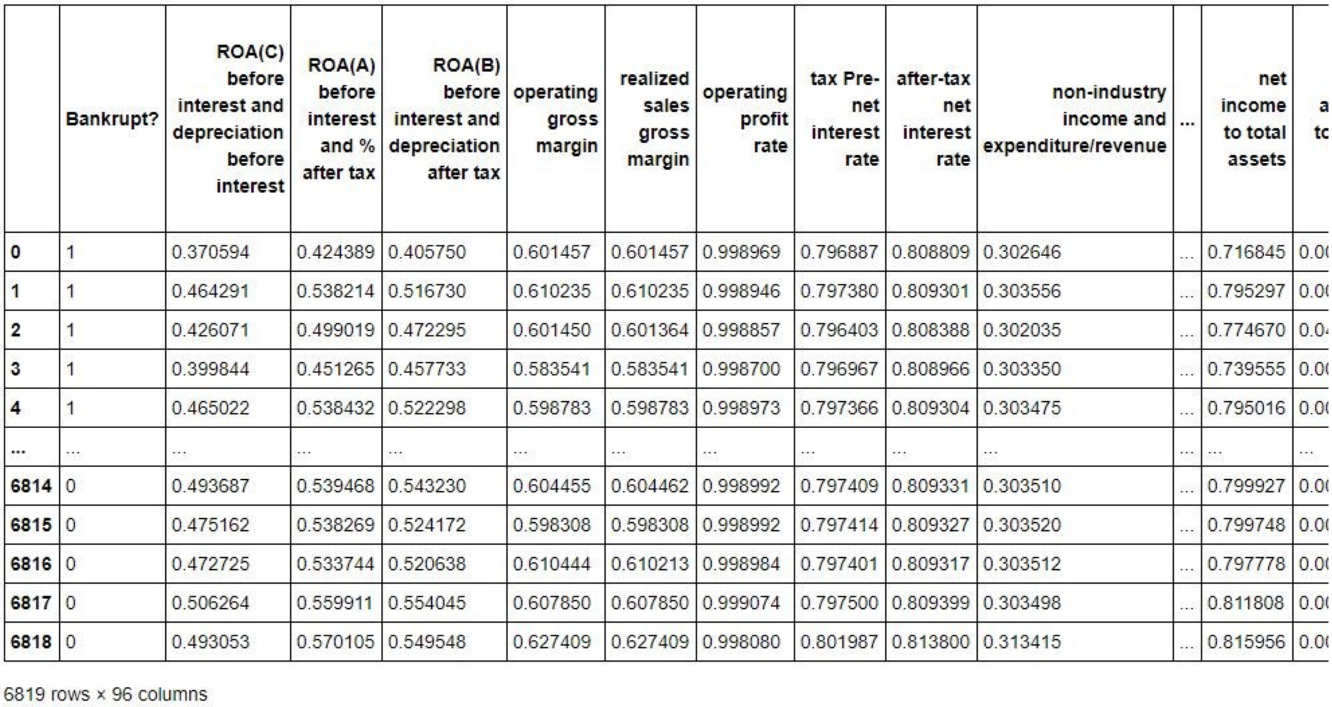
- All code is contained in Jupyter Notebooks located on my Github
    - https://github.com/fredaddy/IBM_Capstone_Project_Predicting_Bankruptcy
- All coding and analysis is done using a Python 3.7 environment initialized with Apache Spark 3.0
- Other relevant libraries include Pandas, Pyspark, Seaborn, Scikit-learn, Numpy, and Matplotlib

2. Data description and Initial Analysis

Based on the stakeholder's criteria, our dataset must have the following:

- High-dimensionality
- Low-sparsity
- A binary classifier for bankrupt vs. not bankrupt
- Among others

Such a dataset is readily available through Kaggle. The dataset consists of ~7000 Tiawanese companies' financial data between 1998 and 2009. There are 95 trainable features, each of which consists of integers or floats and no missing values. Furthermore, the dataset comes prepared with a binary classifier for bankrupt (1) versus not bankrupt (0). See Figure 1 for a look at the dataset.



	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	tax Pre-net interest rate	after-tax net interest rate	non-industry income and expenditure/revenue	...	net income to total assets	a to
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809	0.302646	...	0.716845	0.0
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301	0.303556	...	0.795297	0.0
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388	0.302035	...	0.774670	0.04
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966	0.303350	...	0.739555	0.0
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304	0.303475	...	0.795016	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6814	0	0.493687	0.539468	0.543230	0.604455	0.604462	0.998992	0.797409	0.809331	0.303510	...	0.799927	0.0
6815	0	0.475162	0.538269	0.524172	0.598308	0.598308	0.998992	0.797414	0.809327	0.303520	...	0.799748	0.0
6816	0	0.472725	0.533744	0.520638	0.610444	0.610213	0.998984	0.797401	0.809317	0.303512	...	0.797778	0.0
6817	0	0.506264	0.559911	0.554045	0.607850	0.607850	0.999074	0.797500	0.809399	0.303498	...	0.811808	0.0
6818	0	0.493053	0.570105	0.549548	0.627409	0.627409	0.998080	0.801987	0.813800	0.313415	...	0.815956	0.0

6819 rows × 96 columns

Figure 1

2.1 General Overview of the Data

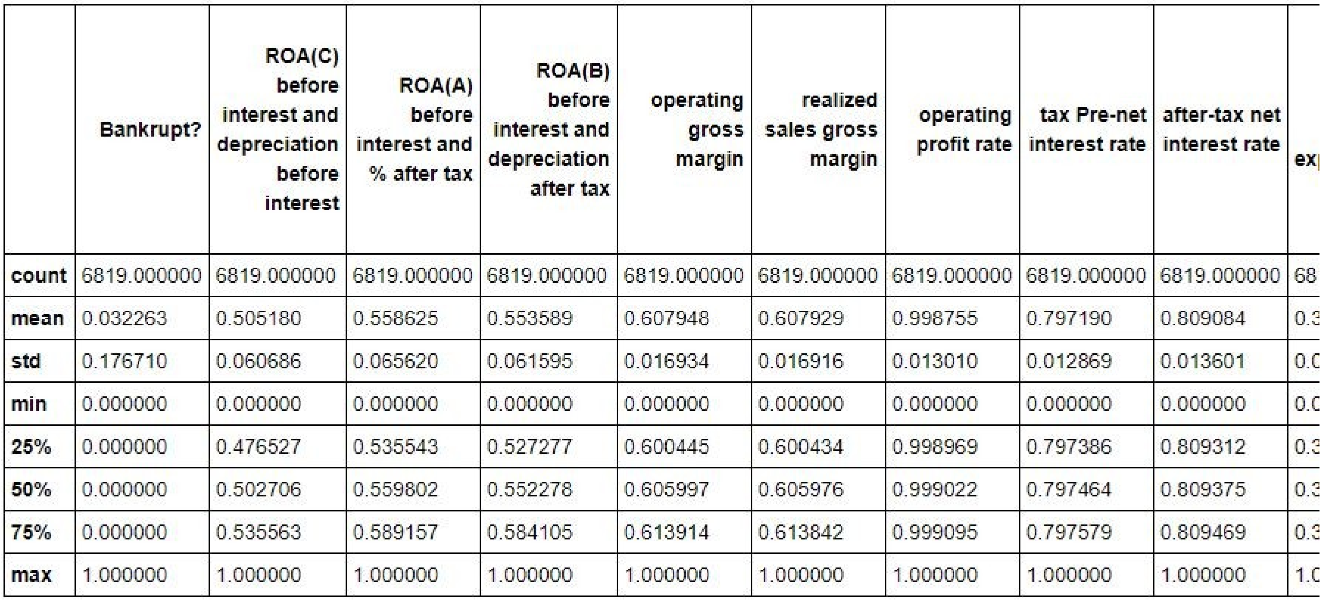
As seen in Figure 1 and through further exploration in the notebook (Predicting_Bankruptcy.data_exp.python-3.7.v1.0), the dataset has the following overarching characteristics:

- 6819 companies
- 95 features per company
- 1 Class label column titled "Bankrupt?"
- All entries are numerical
- There are no NaN (missing) values
- 2 pre-engineered binary features
  - Feature #85: "One if total liabilities exceeds total assets zero otherwise"
  - Feature #94: "One if net income was negative for the last two years zero otherwise"

The main takeaway from these findings is that we will, fortunately, not have to do a significant amount of feature engineering. There are no missing values.

We will, however, have to do some robust feature selection in order to lower the number of features for model training and usability based on stakeholder goals. Many of these features are taken from datasets that are not readily available to the average person. We may also elect to drop the two pre-engineered binary features, as we prefer to have our model work on continuous data only instead of mixing continuous and categorical variables.

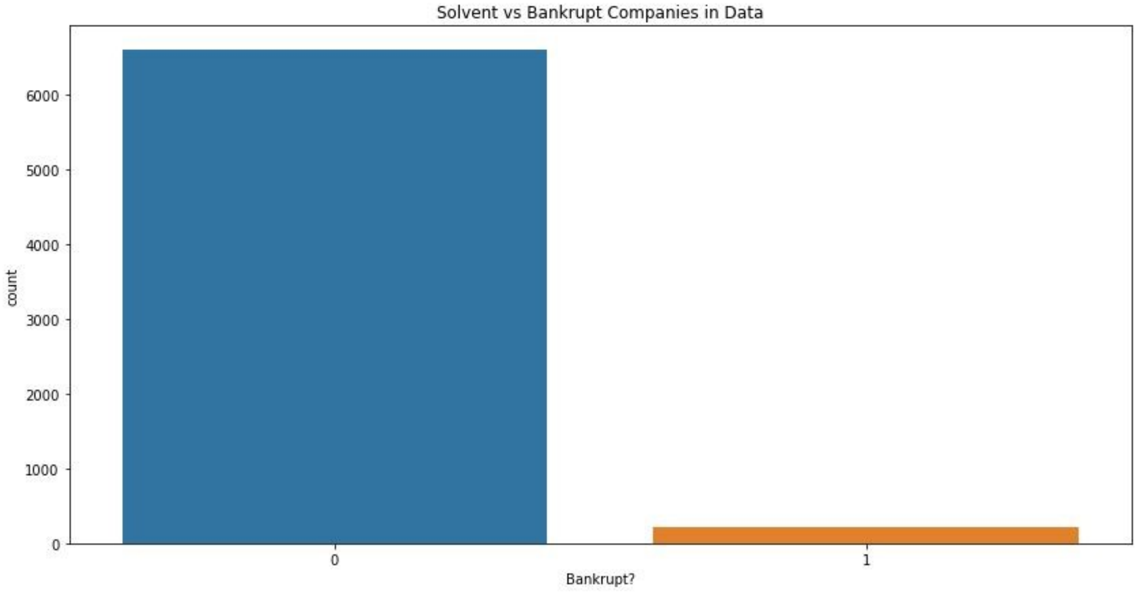
2.2 Statistical Exploration of the Data



	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	tax Pre-net interest rate	after-tax net interest rate	ex
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	68
mean	0.032263	0.505180	0.558625	0.553589	0.607948	0.607929	0.998755	0.797190	0.809084	0.3
std	0.176710	0.060686	0.065620	0.061595	0.016934	0.016916	0.013010	0.012869	0.013601	0.0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	0.3
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	0.3
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	0.3
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

Figure 2

Initial statistical exploration (Figure 2) shows one main potential problem: the mean of our Classification variable is 0.032. This suggests that there are far more financially solvent (not-bankrupt, i.e. “0”) companies than there are bankrupt ones (i.e. “1”) represented in the data. Further exploration (Figure 3) confirms this: we have 6599 solvent companies, and only 220 bankrupt companies in the data. Moving forward, barring reducing the number of observations so that we have more of a 50/50 split between solvent and bankrupt companies, we will need to choose models that have a high likelihood of accounting for this disparity.



```
0    6599
1     220
Name: Bankrupt?, dtype: int64
```

Figure 3

Although we have a large number of features, we still plot a correlation heatmap to see if any particular features are highly-correlated, and thus easy targets for removal (Figure 4). As we can see from the image, there are not enough highly-correlated features to make a significant reduction in the dataset's dimensionality. As a result, we will have to try something else when it comes to feature selection.

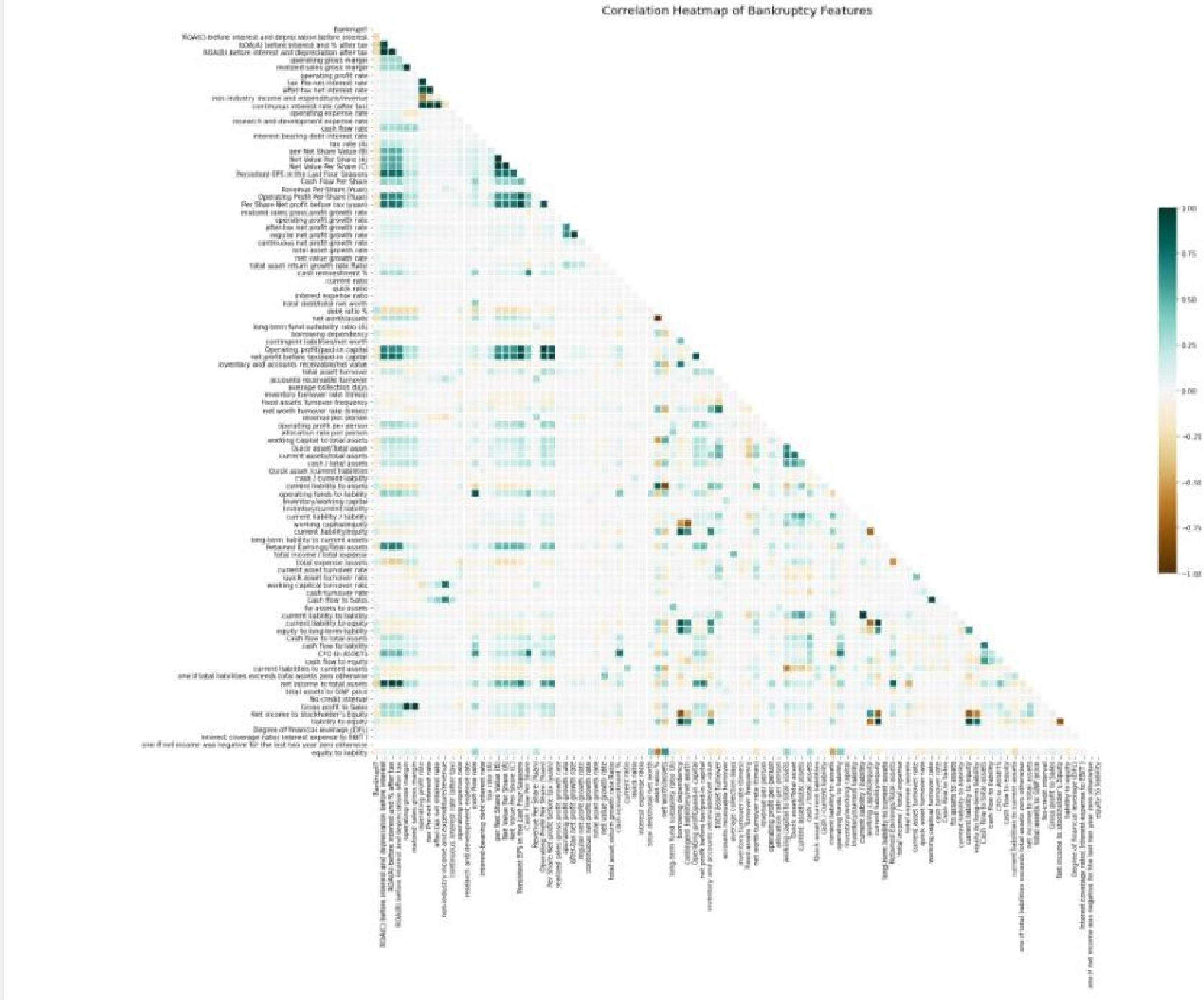


Figure 4

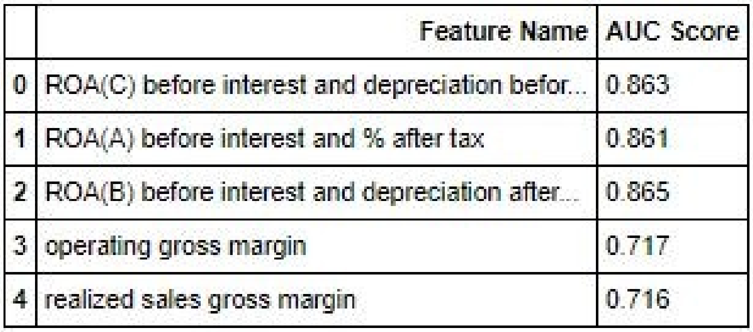
3. Feature Engineering and Selection

In order to effectively select a lower number of features from a dataset with such a high dimensionality, we utilize two metrics:

1. Area Under Curve (AOC)
    - AUC is essentially the integral of the Receiver Operating Characteristic (ROC) curve, which plots true positives versus false positives between the features and the classification variable.
    - A higher AUC implies a higher number of true positives, and therefore more important features.

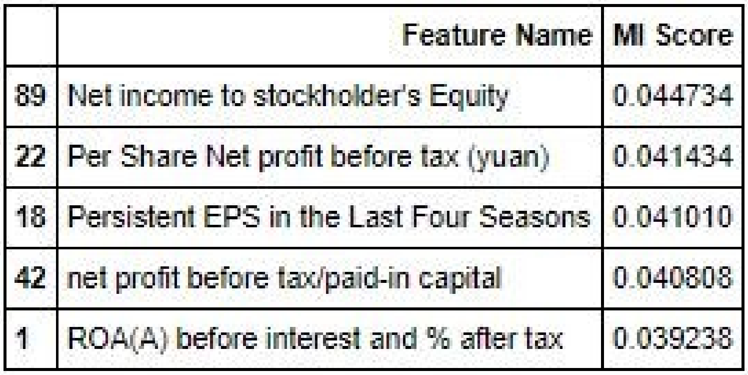
2. Mutual Information (MI)
    - MI is an algorithm that attempts to maximize the amount of information each feature contributes to the classification variable, and minimize the amount of information shared between two features.
    - Again, a higher MI score implies that a feature is highly important to predicting the classification variable and contributes information contained by few (if any) other variables.

The idea is to combine both metrics (auc_score and mi_score) into one consolidated metric (avg_score), and then rank the features by avg_score in order to select the top N features likely to contribute to model effectiveness. N will be a hyperparameter that we will tune during the model training and testing stages. The notebook (Predicting_Bankruptcy.etl_&_feat_eng.pytho3-7.v1.0) contains more information, but here we will just go over the most important aspects of the feature selection process.



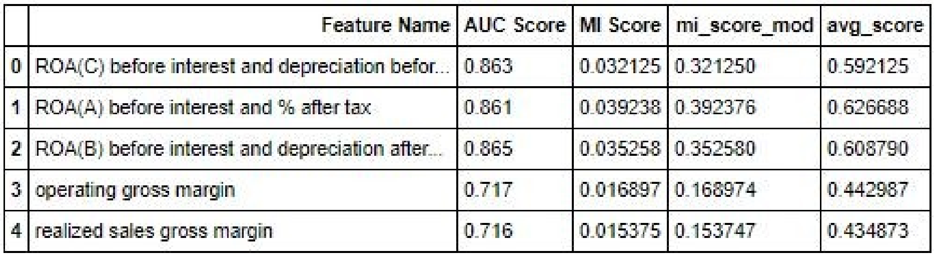
	Feature Name	AUC Score
0	ROA(C) before interest and depreciation befor...	0.863
1	ROA(A) before interest and % after tax	0.861
2	ROA(B) before interest and depreciation after...	0.865
3	operating gross margin	0.717
4	realized sales gross margin	0.716

Figure 5: Area Under Curve Score



	Feature Name	MI Score
89	Net income to stockholder's Equity	0.044734
22	Per Share Net profit before tax (yuan)	0.041434
18	Persistent EPS in the Last Four Seasons	0.041010
42	net profit before tax/paid-in capital	0.040808
1	ROA(A) before interest and % after tax	0.039238

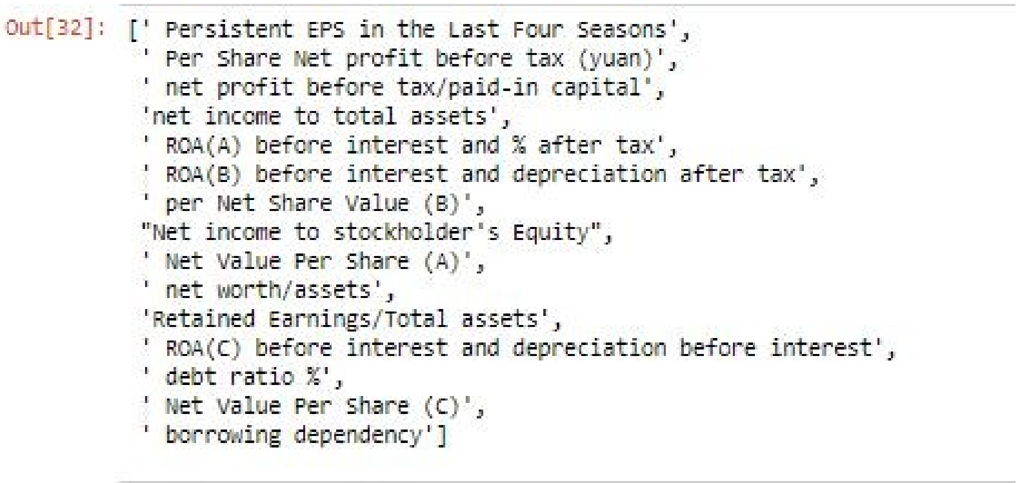
Figure 6: Mutual Information Score



	Feature Name	AUC Score	MI Score	mi_score_mod	avg_score
0	ROA(C) before interest and depreciation befor...	0.863	0.032125	0.321250	0.592125
1	ROA(A) before interest and % after tax	0.861	0.039238	0.392376	0.626688
2	ROA(B) before interest and depreciation after...	0.865	0.035258	0.352580	0.608790
3	operating gross margin	0.717	0.016897	0.168974	0.442987
4	realized sales gross margin	0.716	0.015375	0.153747	0.434873

Figure 7: Combined AUC and MI Scores

We begin by calculating the AUC score for each feature (Figure 5). Next, we calculate the MI score for each feature (Figure 6). Finally, we combine the two dataframes together, multiply the MI score by 10 to put both scores in the same order of magnitude, and average them (Figure 7). The result is a dataframe of features ranked by calculated importance, from which we will select the top N features for our model. The top 15 features (N=15) are seen in Figure 8.



```
Out[32]: [' Persistent EPS in the Last Four Seasons',
           ' Per Share Net profit before tax (yuan)',
           ' net profit before tax/paid-in capital',
          'net income to total assets',
           ' ROA(A) before interest and % after tax',
           ' ROA(B) before interest and depreciation after tax',
           ' per Net Share Value (B)',
          "Net income to stockholder's Equity",
           ' Net Value Per Share (A)',
           ' net worth/assets',
          'Retained Earnings/Total assets',
           ' ROA(C) before interest and depreciation before interest',
           ' debt ratio %',
           ' Net Value Per Share (C)',
           ' borrowing dependency']
```

Figure 8

At first glance, it appears our feature selection method is working well, as most of the features constitute readily available financial data. However, there are some, such as “borrowing dependency” that are not readily available. This suggests that our final model may require N<15. With our feature selection method set as the end of our ETL pipeline, we are ready to move onto modeling.

4. Models

Overview:

- We selected 3 different models to test
  - Support Vector Machine
  - Gradient Boosted Trees
  - Keras-supported Neural Net
- Each model had a 80/20 split of training and testing data
- The evaluation parameter for each was accuracy

4.1 Support Vector Machine

SVM appeared to be a good option as SVMs tend to perform well when there is minimal overlap between features and target classes. Since our ETL procedure attempted to remove any features that did not contribute meaningful information to the model, our hypothesis was that an SVM would have no problem classifying a hyperplane between features. However, the SVM was unable to attain higher than 89% accuracy until N=15 (Figure 9). As a result, it required financial data points that were not readily available in order to achieve reasonable accuracy, and even then the accuracy did not exceed 92%. See the notebook (Predicting_Bankruptcy.model_def_train_eval.SVM) for more detailed information.

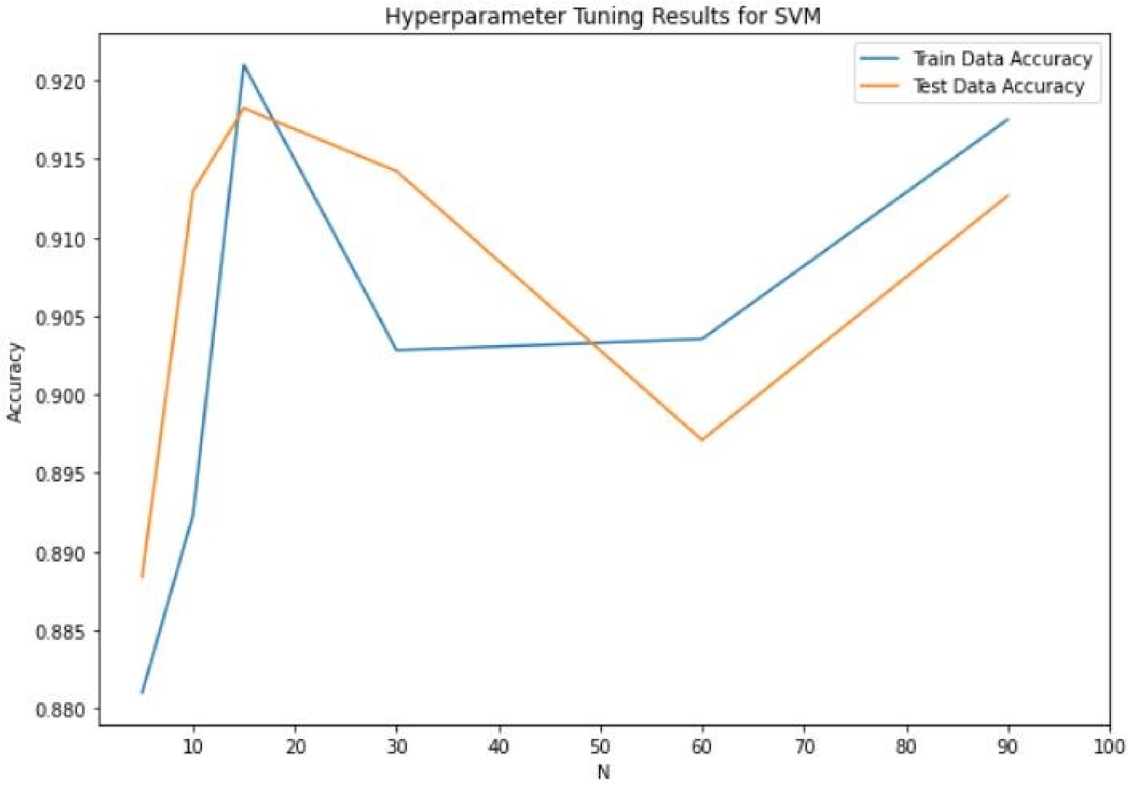


Figure 9

4.2 Gradient Boosted Trees

Upon realizing the SVM model was not sufficient, we moved onto a GBT model. A GBT model was identified as a likely candidate for success due to its reputation of accounting for class labels that were not well-explained by a previously grown "tree." While one iteration may over-focus on the solvent companies, subsequent iterations might be able to hone the model to the insolvent ones. As seen in Figure 10, this theory proved correct. Using a GBT model, we were able to achieve 99% prediction accuracy using only the top 10 features. The model can be found in the notebook (Predicting_Bankruptcy.model_def_train_eval.GBT).

Even more encouraging, the only features needed to achieve our 99% prediction accuracy are:

1. Persistent EPS in the Last Four Seasons
2. Per Share Net profit before tax (yuan)
3. Net profit before tax / paid-in capital
4. Net income / total assets
5. Return on Assets (A) before interest and % after tax
6. Return on Assets (B) before interest and depreciation after tax
7. Per net share value (B)
8. Net income to stockholder's equity
9. Net value per share (A)
10. Net worth / assets

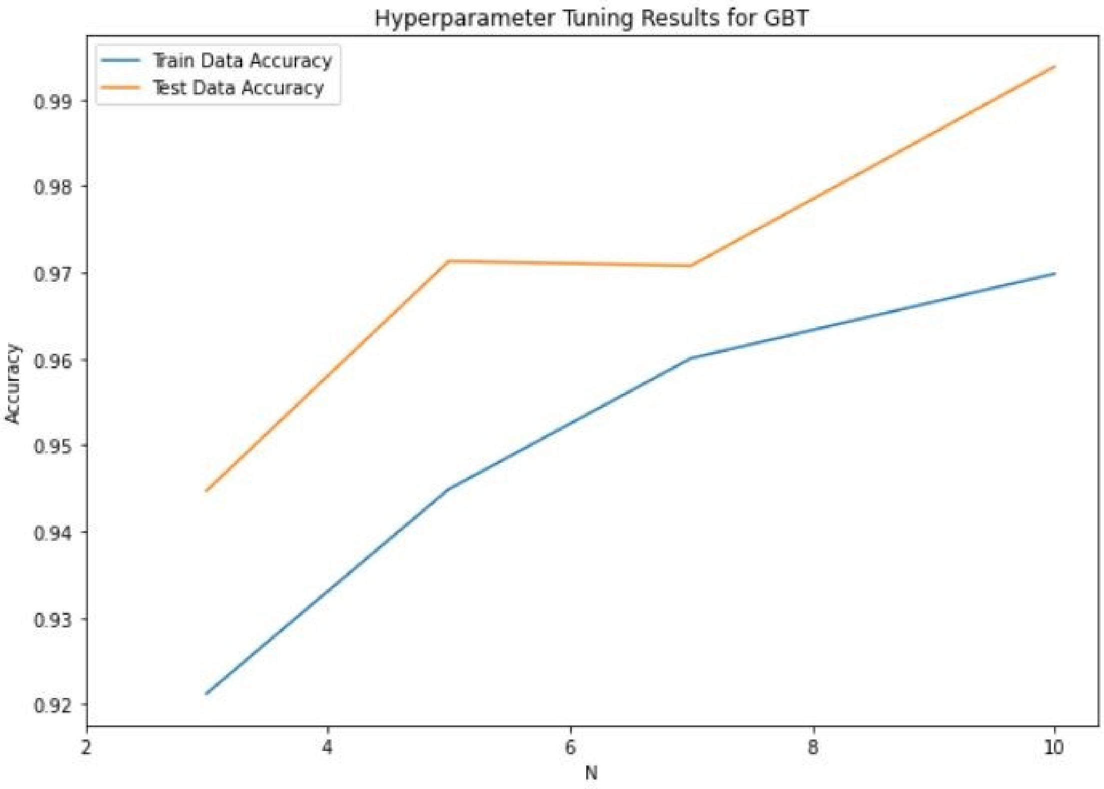


Figure 10

Each of these features constitute readily available financial data and, most importantly, make sense from a business perspective. For example, take "Net income to stockholder's equity." This metric essentially denotes how a company's pre-tax income (an asset, in accounting terms) compares to its total equity outstanding (a liability, in accounting terms). If this ratio is low or negative, it means the company is either losing money or, at the very least, has very little profits to show for all the equity held by its shareholders. Many institutional investors hold debt in addition to equity, and in case of continually low profits, combined with large amounts of equity, the shareholders may elect to cut their losses and force the company into bankruptcy. In such a scenario, shareholders may consider the company to be hopelessly over-leveraged, in which case their main recourse to recoup funds is by liquidating company assets to pay back the debt held by investors.

As a side note, once we realized we had an efficient and accurate GBT, we elected to forego publishing our Neural Net, as the Neural Net was much slower and had lower accuracy.

5. Conclusion and Future Projects

In conclusion, by using a GBT model we were able to achieve 99% prediction accuracy using only 10 of the 95 available features. Of these 10 features, all constituted readily available financial data, meeting our stakeholder's goal.

In the future, we would like to design and deploy an API that allows average people to input these 10 financial metrics from any company's public filings and determine that company's risk of bankruptcy. Furthermore, we plan to build a web scraper to mine these 10 important features from other data sources and expand our dataset to include companies from other countries, especially the United States. Finally, in a future project, we will dive deeper into the meaning of "readily available financial data” to determine if any other lower-ranked metrics can be added to the model to improve prediction accuracy.