Predicting Company Bankruptcy

Springboard DSC - Capstone 3 Final Report
Grace Tang
April 2021

Problem Statement:

Predicting company bankruptcy is critical in any financial institution, especially those involved in lending. For such companies, being able to predict whether a business will succeed or fail can result in successful business loans that grow companies, create jobs, and bolster the economy; or result in millions of dollars in losses.

In this project, I developed several binary classification models to predict whether a business is at risk for bankruptcy. The data used is from the Taiwan Economic Journal, ranging from years 1999-2009. The dataset contains approximately 3.2% bankruptcies and 96.8% non-bankruptcies, a highly imbalanced dataset. **The model with the best recall (fewest false negatives) was Logistic Regression with a recall of 83.33%. The model with the best F1-Score (a more balanced precision/recall) was a XGBoost Classifier, with a F1-Score of 0.3465.**

Background:

While seemingly far off from the United States, there are many similarities between the US and Taiwanese economies. Taiwan's economy is a developed capitalist economy with most government firms being privatized. Furthermore, despite Taiwan's population ranking 57th largest (equivalent to 0.31% of the total world population), it is the 7th-largest in Asia and 20th-largest in the world by purchasing power parity. Taiwan is also the most technologically advanced computer microchip maker in the world. It is definitely an economy worth studying, and may have many insights that carry over to our US economy.

More broadly, the results of this project can be applied to not only finance, but any kind of classification problem with imbalanced data.

Data Wrangling:

The dataset was donated by Deron Liang and Chih-Fong Tsai of National Central University, Taiwan. The data is historical data, spanning 10 years from 1999-2009, collected from the Taiwan Economic Journal. There were 6819 total companies, 95 features, and 1 class label (1 for bankrupt, 0 for non-bankrupt). Figure 1 below shows the first few rows of data, and Figure 2 lists the 95 features.

While looking through the dataset, I found that only **220 (3.23%)** of the 6819 total companies were bankrupt instances, while **6599 (96.77%)** of the 6819 total companies were non-bankrupt instances. Data imbalances often hinder the performance of a model; oftentimes the model may altogether ignore the minority class instances and misclassify them, especially if the minority instances were given the same weight as the majority class. There are several ways to handle class imbalance. The most naive method is to generate new samples by resampling the minority class with sample replacement, but this can lead to overfitting. An alternative method is to synthesize new data using Synthetic Minority Oversampling TEchnique (SMOTE), which we explore in the data preprocessing stage.

The raw data was very clean, with no missing values and no need to impute data. One of the features, called “Net Income Flag”, was removed since it had the same value (1) for every row, and thus did not add any information to the bankruptcy dataset. This reduced our number of features from 95 to 94.

When inspecting for outliers, I found that almost 40% of the dataset had outliers in at least 1 or more features. Rather than removing data points that were considered outliers, I decided to keep all the data.

Figure 1. First 5 rows of the dataset. Only the first 7 columns are shown, including the class label “Bankrupt?”

Index	**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**	**...**
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	...
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	...
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	...
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	...
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	...

Figure 2. List of Dataset Features. Column 94 “Net Income Flag” was later removed.

0 Bankrupt?
1 ROA(C) before interest and depreciation before interest

48 Inventory Turnover Rate (times)
49 Fixed Assets Turnover Frequency
50 Net Worth Turnover Rate (times)

2 ROA(A) before interest and % after tax
3 ROA(B) before interest and depreciation after tax
4 Operating Gross Margin
5 Realized Sales Gross Margin
6 Operating Profit Rate
7 Pre-tax net Interest Rate
8 After-tax net Interest Rate
9 Non-industry income and expenditure/revenue
10 Continuous interest rate (after tax)
11 Operating Expense Rate
12 Research and development expense rate
13 Cash flow rate
14 Interest-bearing debt interest rate
15 Tax rate (A)
16 Net Value Per Share (B)
17 Net Value Per Share (A)
18 Net Value Per Share (C)
19 Persistent EPS in the Last Four Seasons
20 Cash Flow Per Share
21 Revenue Per Share (Yuan ¥)
22 Operating Profit Per Share (Yuan ¥)
23 Per Share Net profit before tax (Yuan ¥)
24 Realized Sales Gross Profit Growth Rate
25 Operating Profit Growth Rate
26 After-tax Net Profit Growth Rate
27 Regular Net Profit Growth Rate
28 Continuous Net Profit Growth Rate
29 Total Asset Growth Rate
30 Net Value Growth Rate
31 Total Asset Return Growth Rate Ratio
32 Cash Reinvestment %
33 Current Ratio
34 Quick Ratio
35 Interest Expense Ratio
36 Total debt/Total net worth
37 Debt ratio %
38 Net worth/Assets
39 Long-term fund suitability ratio (A)
40 Borrowing dependency
41 Contingent liabilities/Net worth
42 Operating profit/Paid-in capital
43 Net profit before tax/Paid-in capital
44 Inventory and accounts receivable/Net value
45 Total Asset Turnover
46 Accounts Receivable Turnover
47 Average Collection Days
51 Revenue per person
52 Operating profit per person
53 Allocation rate per person
54 Working Capital to Total Assets
55 Quick Assets/Total Assets
56 Current Assets/Total Asset
57 Cash/Total Assets
58 Quick Assets/Current Liability
59 Cash/Current Liability
60 Current Liability to Assets
61 Operating Funds to Liability
62 Inventory/Working Capital
63 Inventory/Current Liability
64 Current Liabilities/Liability
65 Working Capital/Equity
66 Current Liabilities/Equity
67 Long-term Liability to Current Assets
68 Retained Earnings to Total Assets
69 Total income/Total expense
70 Total expense/Assets
71 Current Asset Turnover Rate
72 Quick Asset Turnover Rate
73 Working capital Turnover Rate
74 Cash Turnover Rate
75 Cash Flow to Sales
76 Fixed Assets to Assets
77 Current Liability to Liability
78 Current Liability to Equity
79 Equity to Long-term Liability
80 Cash Flow to Total Assets
81 Cash Flow to Liability
82 CFO to Assets
83 Cash Flow to Equity
84 Current Liability to Current Assets
85 Liability-Assets Flag
86 Net Income to Total Assets
87 Total assets to GNP price
88 No-credit Interval
89 Gross Profit to Sales
90 Net Income to Stockholders Equity
91 Liability to Equity
92 Degree of Financial Leverage (DFL)
93 Interest Coverage Ratio (Interest expense to EBIT)
94 Net Income Flag
95 Equity to Liability

Exploratory Data Analysis:

Feature Relationships and Statistical Significance:

Given the many features of the dataset, I wanted to understand how the features related to each other and whether those relations made sense in a real-world context. I began by plotting several heatmaps (Figures 3A, 3B, 3C), histograms, and boxplots. Then I conducted a t-test on all the features between the two classes.

While I saw in the correlation heatmaps that there was no feature with an especially strong positive correlation to bankruptcy, “Debt ratio %” and “Current Liability to Assets”' both had a *slight positive correlation* with bankruptcy. This made real world sense, as companies with larger debts and more liabilities than assets are also more likely to go bankrupt.

There were also many features with *slight negative correlation* with bankruptcy. These included the following:

- In the first graph, Fig 3A.
    - the first three ROA (Return on Assets) features,
    - Persistent EPS (earnings per share) in the Last Four Seasons
    - Per Share Net profit before tax (Yuan ¥)
- In the second graph, Fig 3B.
    - Net worth/Assets
    - Net profit before tax/Paid in capital
    - Working Capital to Total Assets
- In the third graph, Fig 3C.
    - Working Capital/Equity
    - Retained Earnings to Total Assets
    - Net Income to Total Assets
    - Net Income to Stockholders Equity

This again made sense, as these features are signs of a healthy company (i.e. less likely to go bankrupt). If a company has good return on assets, that means they are growing as a company. Persistent earnings per share indicates that their stocks are stable. Large assets and profit again indicates a healthy company.

Setting aside the bankruptcy status, I saw that there were some strong correlations between the other features. In Figure 3A, there is a strong positive correlation between the first 3 ROA features. “Operating Profit Rate”, “Pre-tax net Interest Rate”, “After-tax net Interest Rate”, and “Continuous interest rate (after tax)” all have positive correlations as well. “Liability to Equity” and “Working Capital/Equity” have a strong negative correlation. In short, not all of these features are independent. Many of these features are redundant, and the data could benefit from some dimensionality reduction to reduce noise from these additional features.

In the histograms (not shown), many features appeared to have a normal distribution, there were others which were highly skewed. I adjusted for this skew during the modeling step by doing a log transformation on all features.

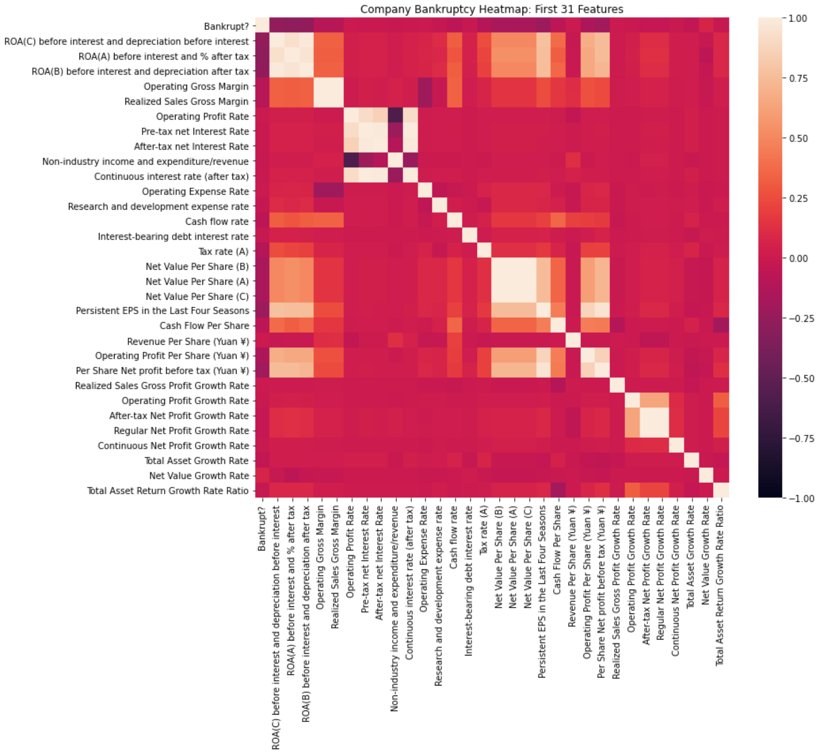


Figure 3A. Correlation Heatmap of the class label and features 1 through 31 of the Company Bankruptcy Dataset. The first column is the class label, "Bankrupt?"

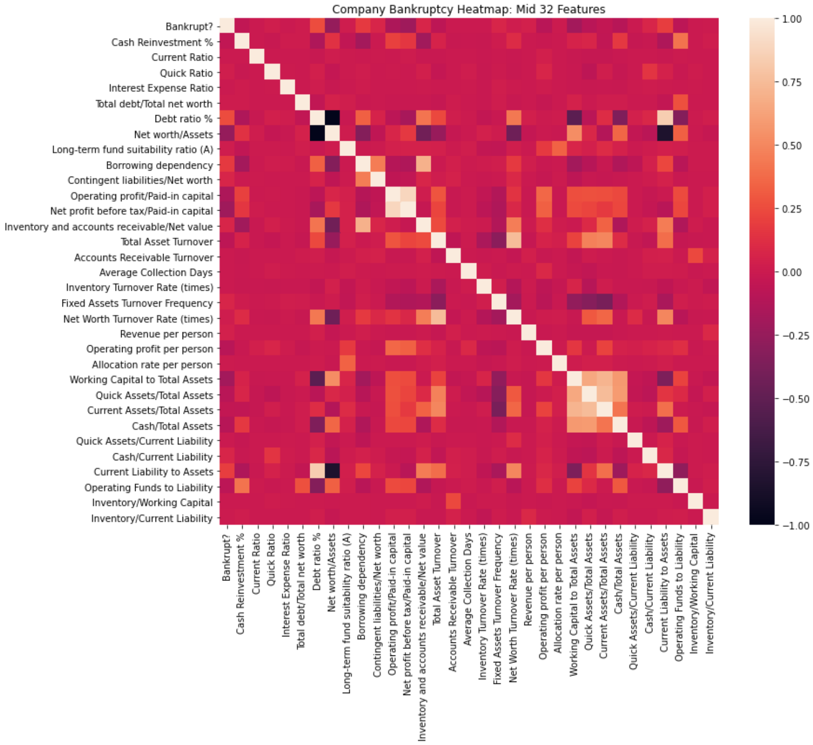


Figure 3B. Correlation Heatmap of the class label and features 32 through 63 of the Company Bankruptcy Dataset. The first column is the class label, "Bankrupt?"

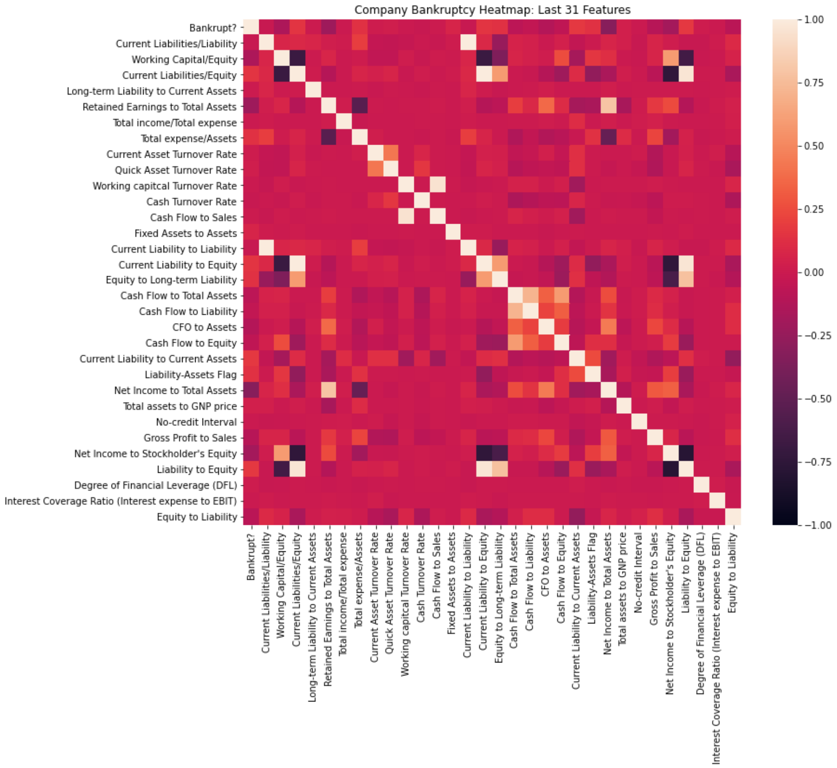


Figure 3C. Correlation Heatmap of the class label and features 64 through 94 of the Company Bankruptcy Dataset. The first column is the class label, “Bankrupt?”

Next, I used a series of t-tests to compare the features of the two classes. A t-test is a statistical test that is used to compare the means of two groups. In this case, the null hypothesis was that, for a given feature, there would be no difference in the means between the bankrupt and non-bankrupt classes. If I were to find that the results were significantly different (which I defined as p-value < 0.01), then that would indicate that the results were statistically significant and the means of the features are *different* between the two classes. The results are displayed in Figure 4.

56 of the 94 features were statistically significant between the bankrupt and non-bankrupt groups. This meant that for these features, their distributions/means were different enough (p-value < 0.01) that we reject the null hypothesis that says the two groups are the same. Parsing through Figure 4, the results made logical sense. ROA (return on assets) would intuitively be higher for non-bankrupt companies than for bankrupt companies. Operating Profit, cash flow, etc. many of these features likewise will intuitively be different between the two groups.

Among the list of non-significant features were features like non-industry income or research and development expense rate. These again made sense that we wouldn't see a significant difference between groups. How much money a company spends on R&D depends more on the market sector it is in, rather than whether it becomes bankrupt or not. These features can be considered noise, and dimensionality reduction would be able to consolidate it and remove it.

The list of significant features can have many practical uses. It can draw our attention to the characteristics of a company which do matter in assessing its likelihood to succeed or fail, whether it stays afloat or goes bankrupt.

Index	Feature	p_value	significance	Index	Feature	p_value	significance
1	ROA(C) before interest and depreciation before interest	1.950813e-106	TRUE	**48**	Inventory Turnover Rate (times)	9.095607e-01	FALSE
2	ROA(A) before interest and % after tax	1.033741e-125	TRUE	**49**	Fixed Assets Turnover Frequency	1.743940e-09	TRUE
3	ROA(B) before interest and depreciation after tax	7.094590e-117	TRUE	**50**	Net Worth Turnover Rate (times)	8.162027e-02	FALSE
4	Operating Gross Margin	1.225969e-16	TRUE	**51**	Revenue per person	1.036103e-03	TRUE
5	Realized Sales Gross Margin	1.859407e-16	TRUE	**52**	Operating profit per person	1.568225e-14	TRUE
6	Operating Profit Rate	9.848617e-01	FALSE	**53**	Allocation rate per person	8.153326e-01	FALSE
7	Pre-tax net Interest Rate	4.819580e-01	FALSE	**54**	Working Capital to Total Assets	2.855312e-58	TRUE
8	After-tax net Interest Rate	4.646049e-01	FALSE	**55**	Quick Assets/Total Assets	8.983430e-13	TRUE
9	Non-industry income and expenditure/revenue	1.706815e-01	FALSE	**56**	Current Assets/Total Assets	2.134267e-04	TRUE
10	Continuous interest rate (after tax)	4.882409e-01	FALSE	**57**	Cash/Total Assets	1.153688e-16	TRUE
11	Operating Expense Rate	6.154927e-01	FALSE	**58**	Quick Assets/Current Liability	7.522925e-01	FALSE
12	Research and development expense rate	4.539929e-02	FALSE	**59**	Cash/Current Liability	1.170368e-10	TRUE
13	Cash flow rate	2.208652e-09	TRUE	**60**	Current Liability to Assets	4.086586e-59	TRUE
14	Interest-bearing debt interest rate	5.686238e-02	FALSE	**61**	Operating Funds to Liability	1.847234e-10	TRUE
15	Tax rate (A)	1.037347e-19	TRUE	**62**	Inventory/Working Capital	8.749361e-01	FALSE
16	Net Value Per Share (B)	5.058196e-43	TRUE	**63**	Inventory/Current Liability	9.458861e-01	FALSE

17	Net Value Per Share (A)	4.685137e-43	TRUE	**64**	Current Liabilities/Liability	8.575144e-02	FALSE
18	Net Value Per Share (C)	1.034332e-42	TRUE	**65**	Working Capital/Equity	2.386080e-34	TRUE
19	Persistent EPS in the Last Four Seasons	3.201175e-75	TRUE	**66**	Current Liabilities/Equity	2.218160e-37	TRUE
20	Cash Flow Per Share	1.459335e-10	TRUE	**67**	Long-term Liability to Current Assets	9.487602e-01	FALSE
21	Revenue Per Share (Yuan ¥)	6.984700e-01	FALSE	**68**	Retained Earnings to Total Assets	5.243960e-74	TRUE
22	Operating Profit Per Share (Yuan ¥)	4.511015e-32	TRUE	**69**	Total income/Total expense	5.556956e-01	FALSE
23	Per Share Net profit before tax (Yuan ¥)	2.440694e-63	TRUE	**70**	Total expense/Assets	8.659356e-31	TRUE
24	Realized Sales Gross Profit Growth Rate	9.698112e-01	FALSE	**71**	Current Asset Turnover Rate	3.246716e-01	FALSE
25	Operating Profit Growth Rate	2.104353e-01	FALSE	**72**	Quick Asset Turnover Rate	3.304165e-02	FALSE
26	After-tax Net Profit Growth Rate	1.805102e-03	TRUE	**73**	Working capital Turnover Rate	8.111256e-01	FALSE
27	Regular Net Profit Growth Rate	2.358402e-03	TRUE	**74**	Cash Turnover Rate	1.364466e-01	FALSE
28	Continuous Net Profit Growth Rate	4.376455e-01	FALSE	**75**	Cash Flow to Sales	9.684704e-01	FALSE
29	Total Asset Growth Rate	2.424265e-04	TRUE	**76**	Fixed Assets to Assets	4.199605e-08	TRUE
30	Net Value Growth Rate	6.682585e-08	TRUE	**77**	Current Liability to Liability	8.575144e-02	FALSE
31	Total Asset Return Growth Rate Ratio	1.639514e-01	FALSE	**78**	Current Liability to Equity	2.218160e-37	TRUE
32	Cash Reinvestment %	2.215705e-05	TRUE	**79**	Equity to Long-term Liability	8.965114e-31	TRUE
33	Current Ratio	8.551369e-01	FALSE	**80**	Cash Flow to Total Assets	5.738535e-09	TRUE
34	Quick Ratio	3.852828e-02	FALSE	**81**	Cash Flow to Liability	3.678279e-04	TRUE
35	Interest Expense Ratio	8.248247e-01	FALSE	**82**	CFO to Assets	1.198120e-21	TRUE
36	Total debt/Total net worth	3.093085e-01	FALSE	**83**	Cash Flow to Equity	1.303233e-06	TRUE

37	Debt ratio %	8.373953e-98	TRUE	**84**	Current Liability to Current Assets	4.558388e-46	TRUE
38	Net worth/Assets	8.373953e-98	TRUE	**85**	Liability-Assets Flag	7.392576e-31	TRUE
39	Long-term fund suitability ratio (A)	1.624007e-01	FALSE	**86**	Net Income to Total Assets	2.098102e-157	TRUE
40	Borrowing dependency	7.330779e-49	TRUE	**87**	Total assets to GNP price	3.741542e-03	TRUE
41	Contingent liabilities/Net worth	5.741725e-09	TRUE	**88**	No-credit Interval	6.469788e-01	FALSE
42	Operating profit/Paid-in capital	1.145170e-31	TRUE	**89**	Gross Profit to Sales	1.225452e-16	TRUE
43	Net profit before tax/Paid-in capital	1.941814e-67	TRUE	**90**	Net Income to Stockholders Equity	2.659396e-51	TRUE
44	Inventory and accounts receivable/Net value	4.847961e-10	TRUE	**91**	Liability to Equity	9.676390e-44	TRUE
45	Total Asset Turnover	1.980446e-08	TRUE	**92**	Degree of Financial Leverage (DFL)	3.855990e-01	FALSE
46	Accounts Receivable Turnover	6.946951e-01	FALSE	**93**	Interest Coverage Ratio (Interest expense to EBIT)	6.492108e-01	FALSE
47	Average Collection Days	5.883388e-01	FALSE	**94**	Equity to Liability	6.487088e-12	TRUE

Figure 4. Table of results of t-tests of the 94 features of the Company Bankruptcy Dataset. The dataset was split between bankrupt and non-bankrupt labels, and each feature between the two labels compared with a t-test. “True” indicates that the feature mean value is significantly different between the two classes, while “False” indicates that they are not significantly different. There are 56 (60%) features that are statistically significant between the two groups. There are 38 (40%) features that are NOT statistically significant between the two groups.

Dimensionality Reduction and Visualization:

From the correlation heatmap and t-test results, many of the features were found to be dependent on each other and 40% (38 of 94) of the features were shown to be non-significant. This indicated that the dataset could benefit from dimensionality reduction. Reducing a dataset from a high-dimensional space to a lower one could reduce noise and improve model performance.

There are multiple methods to approach dimensionality reduction. One common method is Principal Component Analysis (PCA) which captures the variance of the data in a smaller number of features, or principal components. The plots of the first 2 and 3 principal components are shown in Figure 5. The first 2 principal components only explain up to 21% of the variance in the data, indicating that 2 dimensions is insufficient in capturing the full internal structure of the data.

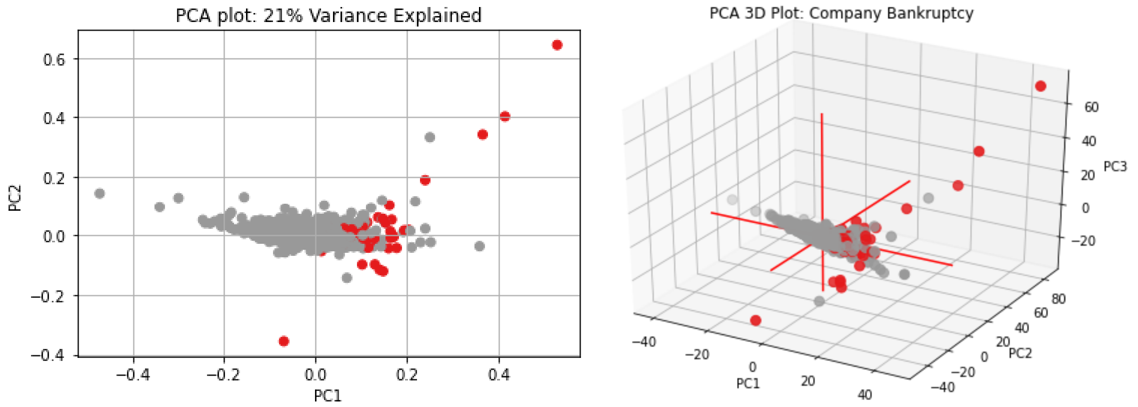


Figure 5. (Left) The dataset plotted along the first and second principal components. (Right) the dataset plotted along the first, second, and third principal components. Bankrupt companies are plotted in red, while non-bankrupt companies are in gray. There is not a clear separation between classes in the PCA plots.

Since there was not a clear separation between the bankrupt/non-bankrupt classes, I then took the reduced dataset and further visualized using t-distributed Stochastic Neighbor Embedding (t-SNE). t-SNE is a tool to visualize high-dimensional data and is particularly sensitive to local structure. Depending on how it is initialized, t-SNE can result in widely different visualizations. Each parameter must also be carefully tuned, and each plot individually inspected as, unlike PCA, there is no scree plot that can be used to evaluate the efficacy of the plot. The various t-SNE plots are shown in Figure 6. The poor separation in t-SNE may indicate that 2 dimensions may be insufficient in representing the internal structure of the data.

Instead of relying on t-SNE to visualize separation, I turned to K-Means Clustering to label and visualize the clusters within the data. K-Means Clustering is an unsupervised learning method that partitions the data and tries to group them into *k* subgroups/clusters, such that data in each subgroup are as similar to each other as possible while data between subgroups are very different, based on similarity measures such as euclidean distance. By running K-Means Clustering on the dataset, I hoped to identify whether there were underlying similarities in the types of companies. Using K=8 clusters, I used clustered against the PCA-reduced data to generate labels. Then, using K=6 clusters, I used K-Means Clustering against the t-SNE reduced data to generate labels. The results are shown in Figure 7. While the visual for the K=6 clusters appears aesthetically pleasing, t-SNE does not preserve global data structure, meaning cluster similarities are not guaranteed, therefore conducting K-Means on the PCA is preferred over the t-SNE.

Figure 6A. t-SNE plots, comparing perplexities of 5, 30, 50, 80, 100.

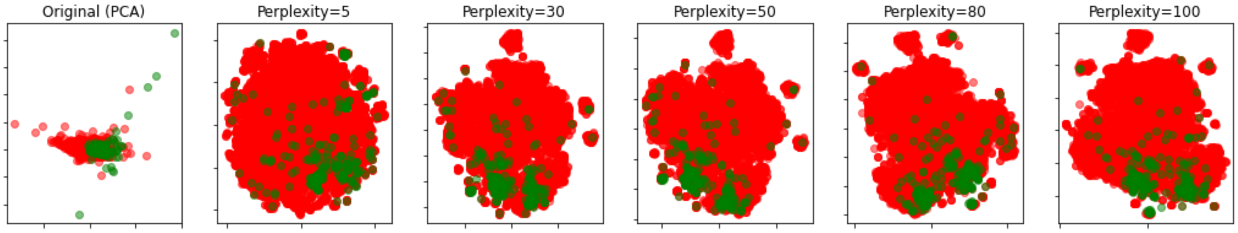


Figure 6B. t-SNE plots, comparing learning rates of 10, 100, 500, 1000. Perplexity is kept at 80.

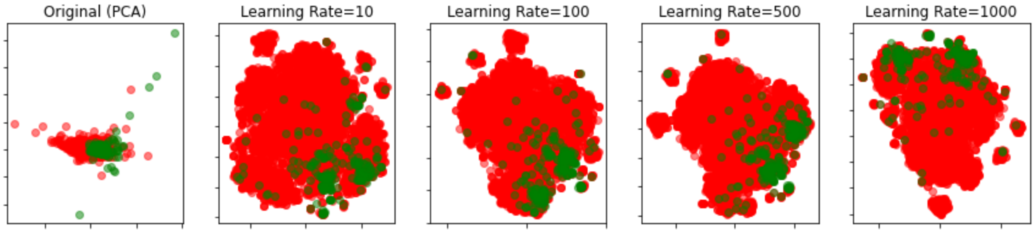


Figure 6C. t-SNE plots, comparing angles of 0.1, 0.2, 0.8, 1.0 (note: the default angle is 0.5, and can be seen in the 4th image under 6B). Perplexity is kept at 80, learning rate is 500.

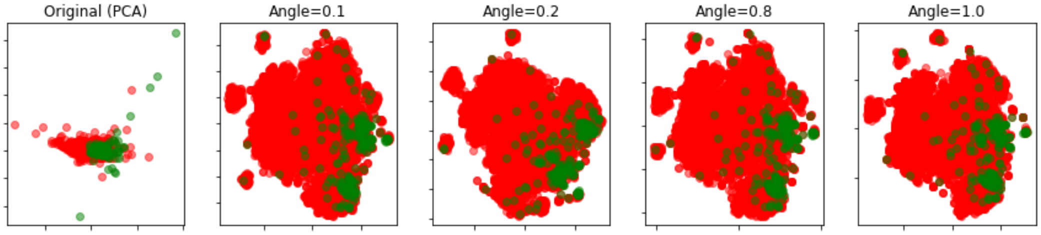


Figure 6D. t-SNE plots, comparison between PCA and t-SNE plots. The final tuned parameters were perplexity = 80, init = 'random', learning_rate = 500, angle = 0.1.

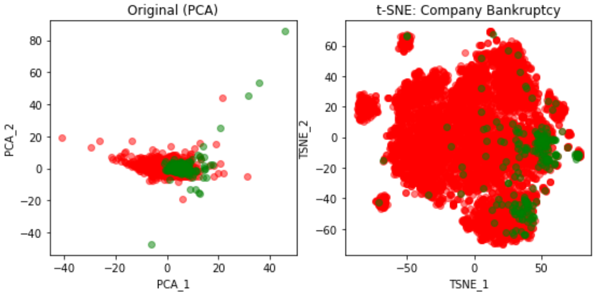
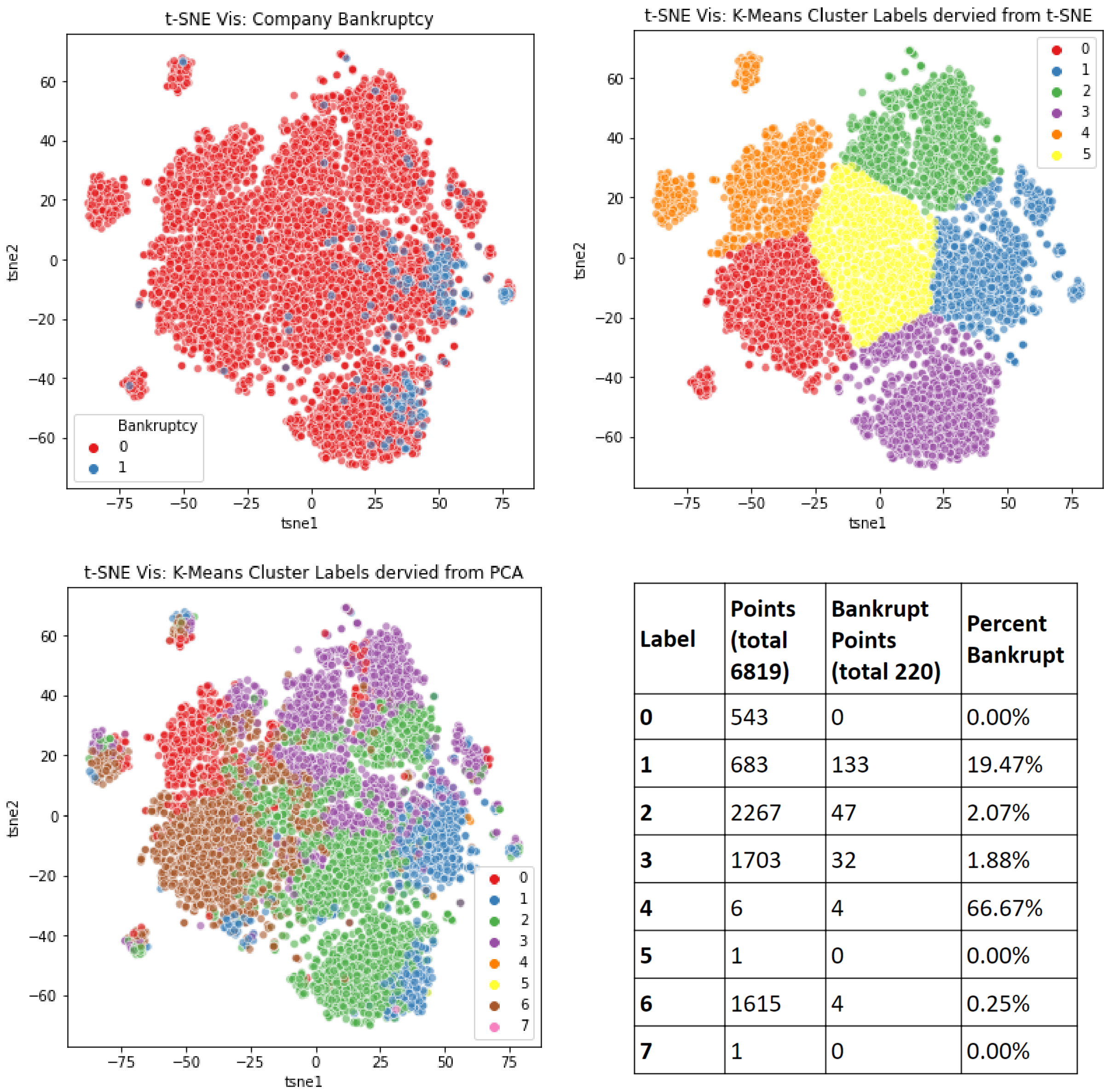


Figure 6. Iterative plotting of the t-SNE. t-SNE results will vary widely depending on how it is initialized (i.e. beginning with a different random_state will yield dramatically different results, and require a different set of tuned parameters). The green points are bankrupt instances, red points are non-bankrupt.



Label	Points (total 6819)	Bankrupt Points (total 220)	Percent Bankrupt
0	543	0	0.00%
1	683	133	19.47%
2	2267	47	2.07%
3	1703	32	1.88%
4	6	4	66.67%
5	1	0	0.00%
6	1615	4	0.25%
7	1	0	0.00%

Figure 7. t-SNE plots of the Company Bankruptcy Dataset with K-Means cluster labels. **Top-Left:** Class labels from raw data. Blue points indicate bankruptcies, while red indicate non-bankruptcies. **Top-Right:** K-Means cluster labels derived from t-SNE data. **Bottom-Left:** K-Means cluster labels derived from PCA data. **Bottom-Right:** Summary of results of the K-Means labels on PCA (94 features reduced to 50 features). Notice most of the bankrupt points are captured in clusters 1, 2, and 3.

Besides data visualization, PCA can also show how much each feature contributes to each principal component. Looking at the eigenvectors of the first 2 principal components, I found that the features with the biggest values and their sign (+/-) are:

- PC 1:
    - ROA (A, B and C) (-)
    - Persistent EPS in the Last Four Seasons (-)
    - Operating Profit Per Share (Yuan ¥) (-)
    - Per Share Net profit before tax (Yuan ¥) (-)
    - Operating profit/Paid-in capital (-)
    - Net profit before tax/Paid-in capital (-)
    - Net Income Total Assets (-)
- PC 2:
    - Debt ratio % (+)
    - Borrowing dependency (+)
    - Inventory and accounts receivable/Net value (+)
    - Net Worth Turnover Rate (times) (+)
    - Current Liability to Assets (+)
    - Current Liabilities/Equity (+)
    - Current Liability to Equity (+)
    - Liability to Equity (+)

Interestingly, the signage is (-) for the highest-contributing features in PC1 and (+) for those in PC2. In other words, the features listed under PC1 likely play a role in decreasing the likelihood of a company going bankrupt (since I defined bankruptcy as '1' and non-bankruptcy as '0'), while the features listed under PC2 likely play a role in increasing the likelihood of bankruptcy. In combination with the list of t-test results, this list of components in the “principal components” can tell us which features to look out for when evaluating the health of a business.

Preprocessing and Training Data Development:

At this point, I proceeded to split the data and began data processing. Earlier I mentioned that the histograms showed a skew in several of the features in the data, indicating that the data would benefit from a log transform. I also discussed the need for SMOTE upsampling of the minority class.

I conducted a 70/30 train/test split, splitting evenly along both classes (i.e. the 220 bankrupt companies were split into 154 (train)/66 (test), likewise for the 6,599 non-bankrupt companies). Then I scaled and processed the data as before: log transform, followed by StandardScaler to scale the data, followed by PCA to reduce from 94 features down to 50, and finally K-Mean Clustering to generate cluster labels. The scaler, PCA, and K-Means were all fitted to the train data, and then used to transform the test data. Care was taken to ensure that nothing was able to “look at” or fit on the test data.

Since the K-Means cluster labels are categorical in nature, I one-hot-encoded the labels, and added them to the data as additional features to the dataset.

Then with the processed data, I upsampled the minority class with Synthetic Minority Oversampling TEchnique (SMOTE). As discussed earlier, our dataset has a class imbalance; only 3.23% of our dataset are bankrupt companies while the other 96.77% are non-bankrupt companies. Since resampling the minority class could lead to overfitting on the minority class instances, I instead used SMOTE. SMOTE works by introducing synthetic examples, rather than resampling. Based on a distance metric, several nearest neighbors of the same class are selected and their features interpolated to generate new data points. Figure 8 shows the results of adjusting the k_neighbors parameter of SMOTE (i.e. the number of points used to synthesize a new data point), the examples shown are k=5 and k=10 neighbors.

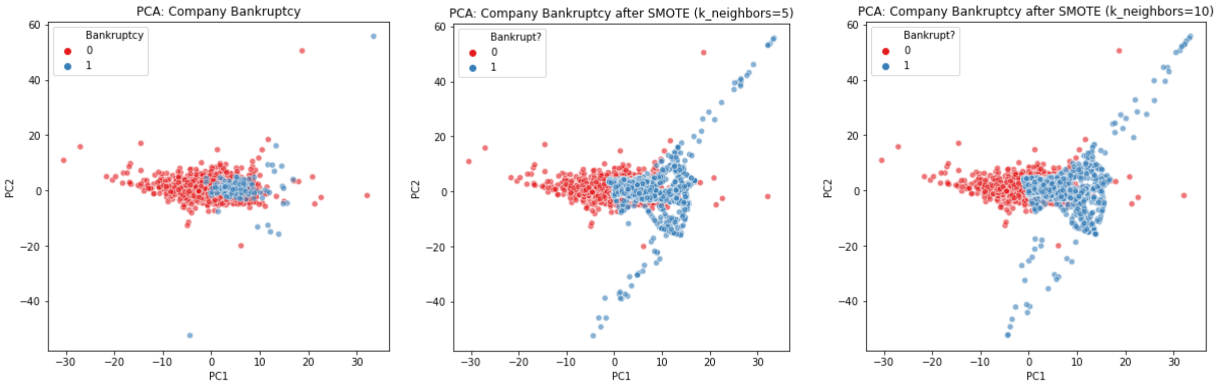


Figure 8. SMOTE upsampling. **Left:** No SMOTE upsampling. **Middle:** SMOTE upsampling with k_neighbors=5. **Right:** SMOTE upsampling with k_neighbors=10.

Baseline Model:

After the SMOTE upsampling, I now had a version of my training data with an even 50/50 split between each class. I then developed a “baseline model”, something to compare against with later models, and trained it on both the original training data as well as the upsampled training data in order to compare the effects of SMOTE on the model performance.

For my baseline model, I selected Logistic Regression, as it is one of the simplest methods used to solve classification problems. I conducted hyperparameter tuning with 10-fold cross-validation against the original training data as well as the upsampled data. The results are summarized below.

For the unprocessed data (i.e. the data from the initial train/test split):

- Best Params: LogisticRegression(C=0.04923882631706741, l1_ratio=0.8, multi_class='multinomial', n_jobs=-1, penalty='elasticnet', random_state=42, solver='saga')
- Accuracy: 0.9619
- F1-Score: 0
- Precision: 0
- Recall: 0

For the processed data (i.e. scaled, reduced with PCA, labeled with K-Means clustering, and upsampled with SMOTE):

- Best Params: LogisticRegression(C=0.5878016072274912, fit_intercept=False, l1_ratio=0.5, multi_class='multinomial', n_jobs=-1, penalty='elasticnet', random_state=42, solver='saga')
- Accuracy: 0.8773
- F1-Score: 0.2807
- Precision: 0.1731
- Recall: 0.7424

What is most striking were the evaluation metrics. The unprocessed data had Recall=0 and Precision=0. This means that not a single bankrupt company was correctly classified. What the regressor had done instead was to label everything as the majority class. This led to a deceivingly high accuracy rate, but inspecting the precision, recall, and F1-Score reveals the truth. At least against a simple model like LogisticRegression(), using SMOTE to upsample our minority class greatly improved the performance of the model.

Model Evaluation:

Moving beyond Logistic Regression, I then looked at the Random Forest Classifier and XGBoost Classifier as alternative models.

The Random Forest Classifier is an ensemble method where bagging is used with Decision Trees. In basic terms, a number of decision tree classifiers are fitted on various sub-samples of the dataset, and averaging between the trees is used to improve predicting accuracy and avoid overfitting. RF tends to have high accuracy, and comes with several advantages, such as being able to handle large datasets and being able to handle missing data (though the company bankruptcy dataset does not have missing data).

"Extreme Gradient Boosting", or XG Boost, is another ensemble method. Unlike Random Forest, XG Boost uses boosting rather than bagging. XG Boost is improved from regular Gradient Boosting in that it uses a more regularized model formalization to control for over-fitting, resulting in improved performance. XG Boost also has several important performance enhancements in its implementation, making it faster than Gradient Boost, and also allowing it to utilize less memory. In many ways, it is the "gold standard" of decision tree modeling.

With each of these models (Logistic Regression, RF Classifier, XGBoost Classifier), I created a pipeline (I used the pipeline from the imbalanced-learn library, as the scikit-learn pipeline is incompatible with samplers like SMOTE). Each pipeline recreated the same data preprocessing steps as described earlier (log transform, standard scaler, PCA, K-Means Clustering), and ending with one of the 3 models. I conducted hyperparameter tuning on each pipeline using Randomized Search with 10-fold cross-validation. The results are shown below.

Table of Results:

Model	Data Preprocessing	Accuracy	F1-Score	Precision	Recall	ROC AUC	P-R AUC	Runtime (s)
Logistic Regression	No	0.9619	0.0000	0.0000	0.0000	0.5887	0.0377	1.0776
Logistic Regression	Yes	0.8739	0.2989	0.1821	0.8333	0.9252	0.3020	2.0035
Random Forest	No	0.9707	0.2308	0.7500	0.1364	0.9497	0.4728	1.0542
Random Forest	Yes	0.9355	0.2979	0.2295	0.4242	0.9203	0.2473	6.6547
XGBoost	No	0.9697	0.3404	0.5714	0.2424	0.9472	0.4602	0.3819
XGBoost	Yes	0.9355	0.3465	0.2574	0.5303	0.9086	0.2524	6.2850

Model	Data Preprocessing	TP	FN	TN	FP
Logistic Regression	No	0	66	1968	12
Logistic Regression	Yes	55	11	1733	247
Random Forest	No	9	57	1977	3
Random Forest	Yes	28	38	1886	94
XGBoost	No	16	50	1968	12
XGBoost	Yes	35	31	1879	101

Model Selection:

From the table of results, we see that Logistic Regression had the best Recall, but at the expense of Precision. It correctly classified many of our positive (bankrupt) instances, but at the expense of misclassifying our negatives (non-bankrupt) instances. Random Forest and XG Boost are slightly more balanced, with XG Boost performing better than RF in both Precision and Recall when the data is preprocessed.

Looking at the F1-Score (which is the harmonic mean of Precision and Recall), we see that XG Boost has the highest F1-Score of the 3 models. F1-Score tends to place more weight on the smaller number, i.e. since Logistic Regression allowed Recall to improve at the expense of Precision, the lower Precision number drags down the total F1-Score. We see a similar effect on the Random Forest model without data preprocessing, in which the Precision is high but Recall is low, giving an overall low F1-Score.

We also see that the F1-Score is higher for the models that were fitted to the processed data. While the data preprocessing steps did not improve metrics like F1-Score or accuracy, they importantly improved the models' abilities to correctly identify bankrupt companies. We see this improvement reflected in the significantly increased Recall value.

With regards to what model should be implemented in predicting company bankruptcy, that would depend on the domain. From the perspective of an institution deciding whether to insure a small business, or a bank deciding whether to issue a loan, the cost of a False Negative (i.e. giving money to a company that will fail) may be greater than the cost of a False Positive (i.e. not giving money to a company that will actually succeed). In that case, we want to minimize FN, and the Logistic Regression Model with SMOTE preprocessing would be the best choice. Another case where Logistic Regression may be preferred is when the company wants to have a general understanding of what contributes to bankruptcy, as Logistic Regression has higher interpretability than does XG Boost.

If, however, we are looking from the perspective of an investor(s) deciding which companies to invest their money in, then (depending on the investment strategy) we may not want to be overly conservative and do actually take on some risk in order to not miss out on promising businesses that were misclassified. In this case, we want a more balanced Precision and Recall, and XG Boost would be the best choice. XG Boost also has the advantage of being more robust to outliers than Logistic Regression, and would provide more consistent results for ongoing classifications.

Future Directions:

If this model were to be implemented in a company, to evaluate which companies to lend to and which to deny, it would be beneficial to develop an understanding of the underlying factors that contribute to company bankruptcy. Since we conducted hyperparameter tuning on a pipeline rather than an individual model, it can be easy to treat the pipeline as a black box, not truly understanding which features in a company contribute to company bankruptcy. Future actions that can be taken could be to remove the features that failed the t-test or remove features with low correlation scores with bankruptcy status from the correlation heatmap (Figure 3), and then analyze how that would affect the model performance. The list of features under the first 2 principal components showed us that ROA, EPS, Net Profit, Debt ratio %, and Current Liabilities, among several other features, all seem to capture more information that contribute to determining company bankruptcy compared to the other features. In combination with the features that passed the t-test, we can further improve our model by removing noise and redundant features.

We also saw that the t-SNE data did not show clear separation between the classes. Finding ways to improve separation and be able to visualize it in 3 or fewer dimensions can be another future direction to explore.

Another avenue to explore is applying these pipelines to American business data. While this project used a Taiwanese dataset, there are similarities between the Taiwan and US economies--both are capitalist societies, and Taiwan's economy is the 7th largest in Asia and 20th largest in the world, lagging not too far behind from the US despite having only 0.31% of the world population. It is not unfeasible to see how these models transfer to our US business data.

Credit:

I would like to thank Chris Esposo for being an awesome Springboard mentor: for teaching me about SMOTE, helping me narrow down ideas, and giving me guidance throughout this project.

I would like to thank Deron Liang and Chih-Fong Tsai (National Central University, Taiwan) for donating this valuable dataset to the UCI Machine Learning Repository, and Federico Soriano Palacios for uploading the dataset to kaggle, where I was able to access it.

Sources:

- Dataset donated by: Deron Liang and Chih-Fong Tsai, deronliang '@' gmail.com; cftsai '@' mgt.ncu.edu.tw, National Central University, Taiwan
- Relevant Paper: Liang, D., Lu, C.-C., Tsai, C.-F., and Shih, G.-A. (2016) Financial Ratios and Corporate Governance Indicators in Bankruptcy Prediction: A Comprehensive Study. European Journal of Operational Research, vol. 252, no. 2, pp. 561-572.
- Dataset: https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction