Determining the Best Classification to Predict a Company's Bankruptcy Potential

John Desiderio
Computer Science Student
University of Georgia
Athens, GA 30602
desiderio@uga.edu

Aaron Xiao
Computer Science Student
University of Georgia
Athens, GA 30602
ax86988@uga.edu

Charles Eidex
Business School Student
University of Georgia
Athens, GA 30602
charles.eidex@uga.edu

Abstract

There are many different factors that influence a company's success. These same factors could also explain why a company fails and goes bankrupt. We will analyze a data set containing a multitude of different attributes as well as samples. We will use data preprocessing methods to trim the data for more efficient use while still maintaining its integrity. From here, we will find the best classification model to predict whether a company will go bankrupt. In addition, we will incorporate feature selection into our research to uncover the leading best leading causes of the company's failure. In the end, we can determine a company's potential for bankruptcy, as well as the cause for its failure.

1 Defining the Problem

For this project, we want to cover all the different ways that a company could fail. In our original project proposal, we wanted to determine whether or not a company would fail. As the semester progressed, we realized this was too vague a question to ask. Instead, we started asking ourselves about all the different ways a company could fail. We decided there were two questions we could use to help us solve our problem:

- Can we build a predictive model to calculate whether a company will go bankrupt given certain attributes?
- What feature(s) can we identify as the greatest contributing reasons for a company failure?

The first question we asked relates to classification, so our subsequent questions also related to classification.

1.1 Finding a Data Set

The first part of project was to find an acceptable data set. After searching the web, we arrived on an acceptable data set from Kaggle at

https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction

Preprint. Under review.

We chose to use this data set for multiple reasons. Kaggle gave the data set a rating of 10 for its usability, and the ease of use was vital for us. There are 6819 samples in the data set, and there are 96 samples for each sample. Our problem is a classification problem, and samples in the data set contain a binary attribute indicating whether the company went bankrupt. In the beginning, we found only 250 companies out of the 6819 went bankrupt, which provided an excellent start to the project.

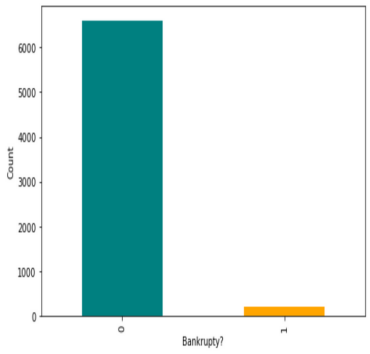

Figure 1: Comparison of the binary distribution. The 1 category indicates a company went bankrupt.

1.2 Data Preprocessing

Based on the previous figure, our team was working with an enormous number of samples. We realized we were also working with many different features for each sample, but we still had a small number of bankrupt companies. To first account for the data skewness, we utilized oversampling to balance the data by generating rows from the existing data set. We then took advantage of Principal Component Analysis to reduce the data set while still maintaining its integrity. Once we completed these tasks, we could use and understand the data more easily.

1.3 Data Visualization

To better understand our data set, we used a data visualization technique to reduce the size of the data into 3 features so we could represent it on a 3D surface. We implemented t-SNE, t-distributed Stochastic Neighbor Embedding, to visualize the data [9]. It should be noted that t-SNE serves as a visualization tool, so we did not utilize the function later in the project [9].

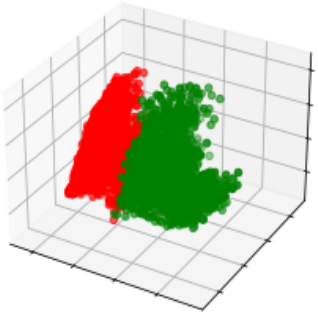

Figure 2: Density figure of the data set. Red indicates a company that failed.

There are two things to point out here. The first is the similarities between the companies that failed. The figure indicates that there is a correlation between the companies that went bankrupt, which means we will be able to uncover it. The second point is the density of the green dots. The sizes of the data in the figure look to be about the same, but there are far many more green dots. On the edges,

some of the green dots appear to be transparent, indicating that the dots in the middle of the cluster is much denser based on the fuller green color.

1.4 Outliers

Considering how a single sample contained so many different features, our group combed through the data to identify potential outliers that could pose a negative impact on the results. We categorized samples as outliers if they contained standard deviations from the 25th and 75th percentiles. From there, we dropped them from our usable set of data. We identified a total of 170 samples and eliminated them from the data set. Running some initial testing with training a simple logistic regression model showed minimal variation in the overall performance [3]. Considering the small number of outliers as compared to the overall size of the data set, the presence of outliers in this particular data set did not seem too big of an issue for the scope of the project.

2 Classification

To answer our first question, we implemented multiple classification models to help us search for the best predictor of a company's bankruptcy. We chose to work with 6 different classification models. The classification models used are as follows: Logistic Regression [3], Support Vector Machines (SVM) [4], Random Forest [6], Decision Tree [5], k-Nearest Neighbor [7], and Gaussian Naive Bayes [2]. In particular, the Decision Tree classifier and Random Forest classifier interested us because they often exerted high performance for classification through feature selection, something we would need later. We also expected the Logistic Regression classifier since its best utilization is binary classification which is exactly what we are seeking to accomplish. We implement the classifiers provided by scikit-learn to help assist us with our research.

2.1 Test Accuracy

After training each model using the same train test split ratio, we first compared the performances by examining the resulting accuracy. We trained the data using a 30% testing ratio. We did try seeing if other testing ratio would dramatically affect performance, but the difference in performance among the testing ratios was marginal.

Table 1: Classifier Test Accuracy

Name	Test Accuracy
Logistic Regression	60.1322
Support Vector	81.4152
Random Forest	97.3844
Decision Tree	95.7324
k-Nearest Neighbor	89.9504
Gaussian Naive Bayes	51.4868

There are a number of surprising results we gathered from the table. The most astonishing part of the results was Logistic Regression's poor performance.[1] Based on what we understood about logistic regression, we predicted it would preform a lot higher than we realized. Gaussian Naive Bayes score also surprised us. GNB was by far the worst performer out of all the classifiers. We ran multiple tests to see what we could do to try to improve its score, but all the new score differences were marginal. Looking at the classifiers that did perform well, the Support Vector Classifier[2] trailed the k-Nearest Neighbor, which trailed the Random Forest Classifier and Decision Tree Classifier. The difference between the Random Forest Classifier and Decision Tree Classifier was marginal, and k-Nearest Neigbors is also performing at around 90% it would appear all three of the classifiers are reliable classifiers.

[1] It should also be noted before we decided to implement oversampling on the data, the Logistic Regression classifier received an accuracy score of 97%. This made it all the more confusing as to why its score dropped so far.

[2] We tried to use LinearSVC and fit the data, but the kernel repeatedly crashed even when we tried to adjust the function.

2.2 Dissecting the Accuracy Scores

The next part in understanding the data is to figure out what the classifiers correctly identified. We used confusion matrices to assess the true positives, true negatives, false positives, and false negatives. We compiled the results into a table [8].

Table 2: Confusion Matrix Results

Name	True Positive	True Negative	False Positive	False Negative
Logistic Regression	785	1384	439	1028
Support Vector	1540	1398	425	269
Random Forest	1801	1756	67	8
Decision Tree	1739	1719	104	70
k-Nearest Neighbor	1788	1506	317	21
Gaussian Naive Bayes	1783	86	1737	26

The first noticeable aspect of this table is the Gaussian Naive Bayes classifier's results. It seems like the prediction model believed it every sample it worked with was a positive save for a select few. Another interesting part to point out with the table is how the k-Nearest Neighbor classifier correctly identified more True Positives and less False Negatives than the Decision Tree classifier. The Decision Tree classifier still outperformed the k-Nearest Neighbor classifier, but it still lacked in some aspects. As the table suggests, the Random Forest test results continued to shows why that classifier is the superior prediction model.

2.3 Model Calibration

After assessing the confusion matrix results for each classifier, we wanted to measure each model's calibration over the time it took to train the model [1]. The results of the the calibration are in the following figure:

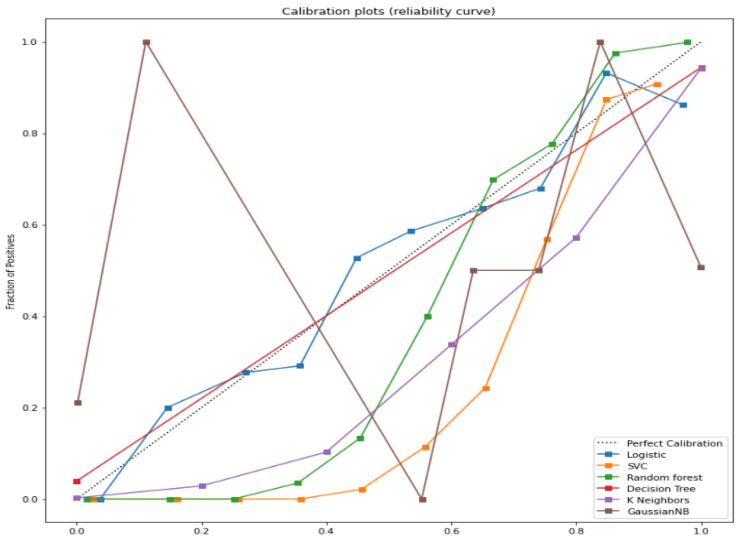

Figure 3: Model calibration. LR=Blue, SVC=Orange, RFC=Green, DTC=Red, KNN=Purple, GNB=Brown

One plot that stands out among the rest is the Gaussian Naive Bayes. The behavior of its calibration suggests it is highly erratic. The jump to the top, then down to the bottom, then back to the top again reflect its unreliability. It is clear this model is not suitable as a predictive model for the data set. The Decision Tree classifier's performance. It maintained the closed to slope to the perfect calibration. It

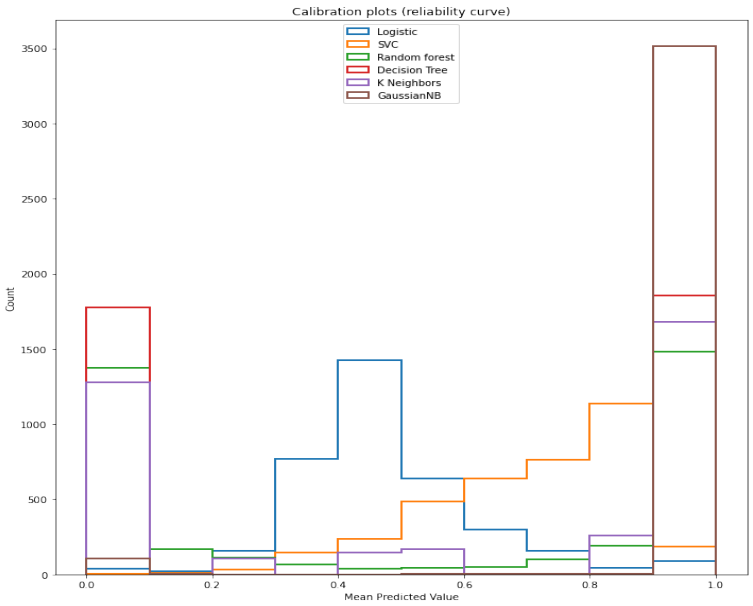

Figure 4: Model calibration. LR=Blue, SVC=Orange, RFC=Green, DTC=Red, KNN=Purple, GNB=Brown

performed much a better than Random Forest classifier when analyzing it through the calibration. The k-Nearest Neighbors classifier and the Support Vector Classifier performed adequately, but the Logistic Regression model performed the second best behind the Decision Tree classifier.

2.4 Conclusion

After running the different assessments on the classifiers, our group reached the conclusion that the Decision Tree classifier was the best classifier out the group of classifiers we selected. The Random Forest classifier performed better than the Decision Tree classifier, but the Decision Tree classifier had the undisputed best calibration out of all the classifiers. The difference in the test accuracy between the Random Forest classifier and the Decision Tree classifier was not as great as the difference between their calibrations. For those reasons, the best model to predict whether a company will go bankrupt is the Decision Tree Classifier.

3 Feature Selection

Our second question for this project sought to identify important features that may be indicative of bankruptcy.To perform this task we used the random forest classifier to try selecting different number of features and compare the accuracy scores [6]. The results showed very little change when testing different numbers of features from 5 to 30. As such, we chose to proceed with the top 10 most important features identified by the Random Forest classifier.

3.1 Data Analysis on Features

With the 10 features we performed data analysis to find out variations in the features between bankrupt vs non-bankrupt companies. By comparing the median values for each feature between bankrupt vs non-bankrupt companies, we could identify how the stats varied.

From the resulting graphs of the ten features, we found four that had a variation. The most telling signs of a bankrupt or likely at risk of bankruptcy include a high total debt to net worth ratio as well as a low cash to total assets ratio. Less telling signs includes value growth factor rate and Earnings Per Share (EPS). After performing the data analysis and identifying the best indicators of bankruptcy, we also identified the weakest indicators of bankruptcy. These features all shared a commonality of a high Gini index. Out of all the 96 different features, these particular features were the most useless in

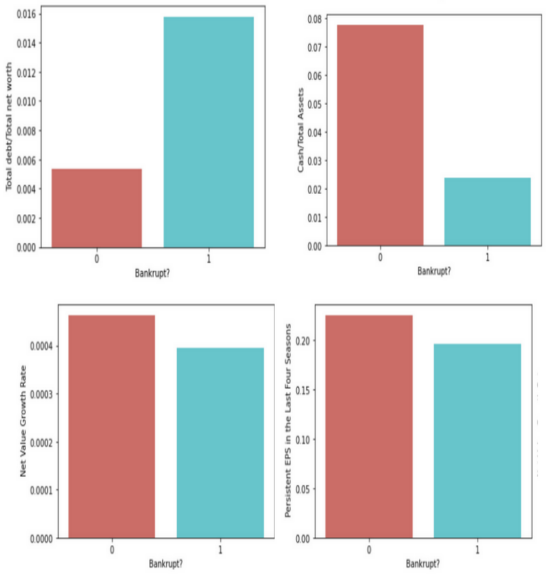

Figure 5: Variation among four of the ten features. Top Left: Total Debt/Total Net Worth, Top Right: Cash/Total Assets, Bottom Left: Net Value Growth Rate, Bottom Right: Persistent EPS in the Last Four Seasons

trying to assess a company's potential to go bankrupt. There are many more that also do not serve a useful purpose, we identified these 6 as the worst. The features are in the following table:

Table 3: Worst Feature Gini Index

Name	Gini Index
Cash Flow Rate	0.5
Cash Reinvestment	0.5
Long Term Fund Suitability Ratio	0.48
Quick Assets/Total Assets	0.444
Operating Profit Per Person	0.444
ROA Before Interest and Depreciation after Tax	0.444

4 Conclusion

In conclusion, the project's goal was to be able to predict company bankruptcy and identify useful features for identifying bankruptcy. This was achieved by taking a dataset, preprocessing the data, finding the best attributes in the dataset, and then building several models to find which would be the best predictor. We found that using the 10 best attributes would yield a relatively high test score, and the best of these attributes were a high total debt to net worth ratio, a low cash to total assets ratio, value growth factor rate, and EPS. The worse attributes included quick assets / total assets, operating profit per person, and long term fund suitability ratio. After preprocessing the data and selecting the best features, we tested the data on several models and found the decision tree classifier to have the highest accuracy and best confusion matrix with KNN and SVC returning decent results as well. In the end, we were able to narrow down the dataset to the most useful features, and find a classification model that gave fairly good results.

5 Broader Impact

We hope other students and researchers can use our approach to problem solving in their own endeavors. We hope they can pick apart this paper and use it in building their own idea. More importantly, we hope to get an A for our efforts.

6 References

[1] "1.16. Probability Calibration¶." Scikit, scikit-learn.org/stable/modules/calibration.html.

[2] "Sklearn.naive_bayes.GaussianNB¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[3] "Sklearn.linear_model.LogisticRegression¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[4] "Sklearn.svm.SVC¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[5] "Sklearn.tree.DecisionTreeClassifier¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.h

[6] "Sklearn.ensemble.RandomForestClassifier¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[7] "Sklearn.neighbors.KNeighborsClassifier¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[8] "Sklearn.metrics.confusion_matrix¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[9] "Sklearn.manifold.TSNE¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.