Predicting Bankruptcy Using Financial Attributes: A Machine Learning Approach

Master Of Quantitative Economics

The University Of California, Los Angeles

Author : Anshika Sharma

Advisor : Dr. Patrick Convery

Date Completed: 11th May, 2022

Contents

1. Abstract

Bankruptcy of companies and enterprises affects the financial market at multiple fronts, and hence the need to predict bankruptcy among companies by monitoring multiple variables takes on an added significance. Financial analysis is a critical aspect of any firm's or company's operations, and how the company projects its future financial health is a key input for all of its stakeholders. The aim of this paper is to calculate the financial projections of Polish companies (data set collected from the UCI repository) in terms of going bankrupt or remaining in operation. The intention of this study is to illustrate and review how machine learning can be exploited in the field of economics. More specifically, the aim is to study how machine learning methods can be used to predict corporate bankruptcies. For this, multiple machine learning algorithms were used and compared, including Logistic Regression, Quadratic Discriminant Analysis, K- nearest neighbor (K-NN), Random Forest and Decision Trees. Random Forests proved to be the most accurate method for predicting the likelihood of a company's bankruptcy.

2. Introduction

Prediction of a corporate bankruptcy is of great importance in economic decision making. A business condition of both small or large firms concerns the local community, industry participants and investors, and also influences policy makers and the global economy. Estimating the risk of corporate bankruptcies is of vital importance to creditors and investors. Corporate bankruptcies can have serious effects both locally and globally (De Haas and Van Horen, 2012), employees, investors, customers, suppliers and their financiers are all affected when a company disappears (Engström, 2002). The consequence of the high social and economic costs attached to corporate bankruptcies has attracted a lot of attention from researchers for better understanding and developing performant models which predict the financial distress with high accuracy.

According to the American Collectors Association, bankruptcy filings in the United States have been on the rise since 2015. More specifically, since September 2015, there has been

about a 65 percent increase in the number of filings (American Bankruptcy Institute). Many economists are concerned that the rise in bankruptcy filings could be a precursor to future economic downturns. Economic crisis of 2007 highlighted the need for market sustainability and the need for predictive tools to better forecast such events and prevent bankruptcies and any such economic crisis from happening again.

Until recently the dominating methods for predicting corporate bankruptcies have been based on statistical modeling, however, lately models based on machine learning have been proposed (Linden et al., 2015). Machine Learning models have successfully been used for many classification and regression problems and these models have often outperformed traditional classification methods (Krizhevsky et al., 2012). The purpose of bankruptcy prediction is to assess the financial health status and future perspectives of a company. For a given period of time this problem can be modeled as a two-class classification problem (Zie ba et al., 2016). Companies either survive the given time period, or go bankrupt during it. The problem is to predict which of these two possible outcomes is the more probable one.

This paper uses the data for Polish companies, freely available on the internet and identifies which of these companies declared bankruptcy or did not go non-bankrupt using several attributes like net profit, total liabilities, sales and many others. There has been research about this problem, so the aim of this study is to build classification models and see which one of those gives better predicted values. We believe analytics can help us understand if the values of several features can help us categorize the future of Polish companies in terms of bankruptcy. The paper uses financial data from 10,503 Polish enterprises to predict bankruptcy using a variety of machine learning techniques. This paper also identifies the model with the best predictive power based on a given set of standard financial metrics.

3. Data Set and Features

3.1 Data Gathering

The dataset for this paper for addressing the bankruptcy prediction problem is the Polish bankruptcy data, hosted by the University of California Irvine (UCI) Machine Learning Repository. The bankrupt companies were analyzed for the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. The dataset is very appropriate for bankruptcy prediction because it has highly useful econometric indicators as attributes (features) and comes with a large number of samples of Polish companies that were analyzed in 5 different timeframes. The data is classified in time frames depending on the forecasting period. First year forecasting period data is for bankruptcy status after 5 years, Second year forecasting period indicates bankruptcy status after 4 years and so on. This paper evaluates the third year forecasting period data that is for bankruptcy status after 3 years. It consists of 10,503 different Polish companies in the year 2010. The reason for choosing Polish companies is because Poland is the sixth largest economy in the European Union and as per McKinsey's report in 2015, Poland will become Europe's new growth engine by 2025. The motivation to choose this repository was that since the year 2004, Poland saw many manufacturing sectors going bankrupt.

Each observation in the data set is an individual company, and there are 64 numerical variables, representing different financial statement values like net income and long-term liabilities. These attributes will be used to predict bankruptcy, given by the class variable. The class variable is a dummy variable, where 1 represents a company that went bankrupt after three years, while a 0 represents a company that did not file for bankruptcy after 3 years. It is also important to note that the data only has 495 (4.7%) companies that went bankrupt after three years.

3.2 Data Preparation

3.2.1 Correcting Imbalanced Data Using SMOTE

While carrying out Exploratory Data Analysis, it is observed that there is an unbalanced dataset, with only 5% of the observations representing companies that went bankrupt and 95% of the observations being companies that did not go bankrupt. This presents an issue when trying to evaluate the performance of each model. For example, a model that does not predict any bankruptcies will still produce an accuracy of 95%. Data Imbalance can be treated with Oversampling and/or Undersampling. In data analysis, Oversampling and Undersampling are opposite and roughly equivalent techniques of dealing with Data Imbalance, where they adjust the class distribution of a data set (i.e. the ratio between the different classes/categories represented). In order to address this issue, I used the Synthetic Minority Oversampling Technique (SMOTE), which simultaneously over-samples from the class that is underrepresented and undersamples from the class that is overrepresented. After applying this technique, it was seen that 32% of the observations in the new dataset represent firms that filed for bankruptcy.

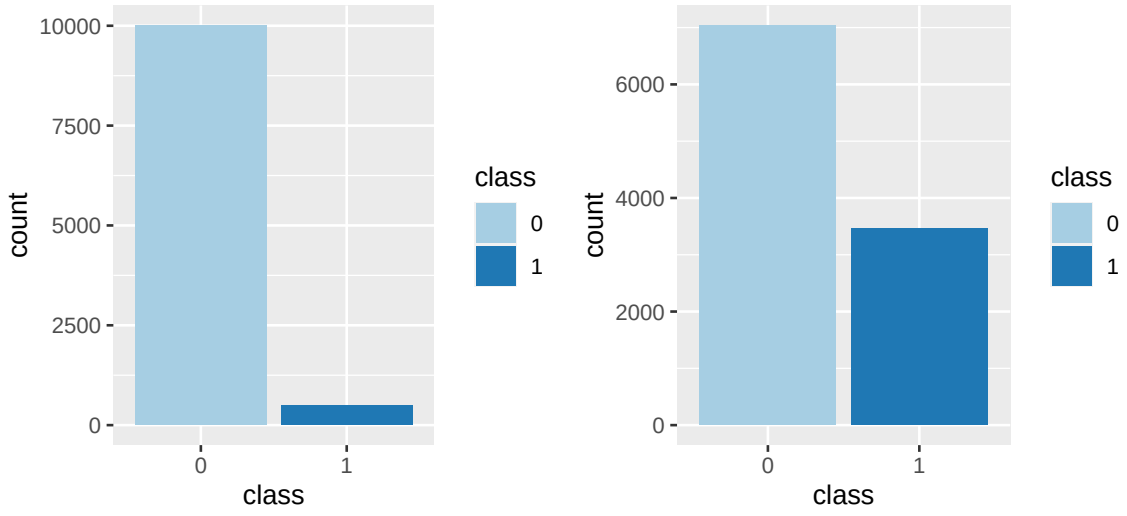

Figure 1: a)Unbalanced Dependent Variables. b) Balanced Dependent Variable After Using SMOTE Analysis

3.2.2 Dealing with Missing Data

For machine learning (ML) applications, high data quality standards are crucial to ensure robust predictive performance and responsible usage of automated decision making. One of the most frequent data quality problems is missing values. Upon inspection of the data, it was found that 44 of the 64 attributes had missing values. As can be seen from the figure below, the variable, X37, "(current assets - inventories) / long-term liabilities" has the maximum number of missing values followed by the ratio of sales in time n over sales in time n-1. For these attributes, the missing values were imputed using the mean of the attribute belonging to each missing value. Mean imputation attenuates any correlations involving the variable(s) that are imputed. However, this method can lead to biased estimates, so for robustness check, MissForest imputation technique was also used. It initially imputes all missing data using the mean/mode, then for each variable with missing values, MissForest fits a random forest on the observed part and then predicts the missing part. This process of training and predicting repeats in an iterative process until a stopping criterion is met, or a maximum number of user-specified iterations is reached. Both the techniques gave the same result, so I decided to go ahead with mean imputation.

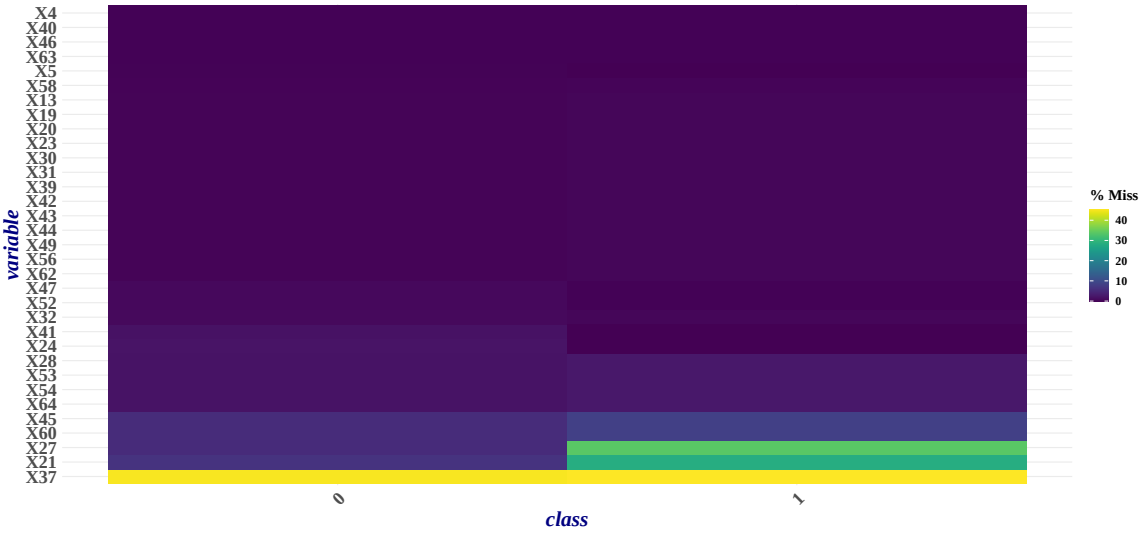

Figure 2: Visual Representation of Missing Observations

3.2.3 Feature Selection: Running Boruta on Balanced data

After correcting the imbalance dataset, we now focus on determining which features to use in the models. Since the data set has 64 features, there were concerns about overfitting and also including features that did not provide any relevant explanatory power to the models. To assist in the determination of which features to include in the models, Boruta algorithm was carried out. Essentially, the algorithm performs a cross-validation on each variable to observe and report features that are important and features that are rejected.. The figure below shows the attributes that Boruta algorithm deemed important. The importance of the features with respect to the outcome variable is measured on the y axis. The green box plots are the features that are considered predictive, so they are indicative of acceptance. I opted to use the top 30 features from the Boruta Algorithm. The most important features are X 27" and "X 15". They measure "profit on operating activities/financial expenses" and "(total liabilities * 365)/(gross profit + depreciation)" respectively. A detailed list of the corresponding attribute description can be found in the appendix. This matches economic intuition since lower the interest coverage ratio, the greater the company's debt and the possibility of bankruptcy. Intuitively, a higher ratio indicates that more operating profits are available to meet interest payments and stronger financial health. Also, this ratio helps to assess a company's ability to meet its long-term financial obligations.

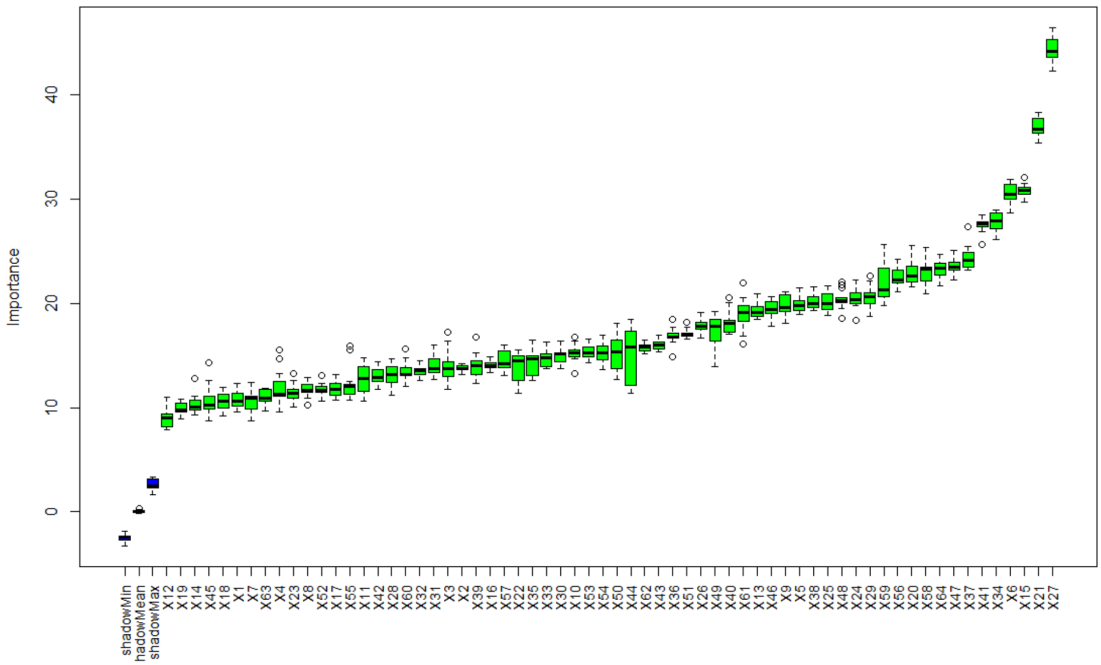

Figure 3: Feature selection using Boruta Algorithm

3.2.4 Multicollinearity

After feature selection, I did correction for multi collinearity and further removing variables that would cause problems in the analysis. Since most of the attributes are constructed using similar terms, it was expected that there would be strong correlations between the attributes, and as a result, strong multicollinearity. It is important to address the problem of multicollinearity, which may affect the accuracy of the model and the reliability in determining the effects of individual independent features on the dependent feature in the model can be undermined, which can be a problem while interpreting the model. Figure 4, below confirms this by showing the existence of moderate to strong positive correlations between different attributes. To remove collinear attributes, I decided to run the Variance Inflation Factor algorithm to identify the attributes that primarily contribute to multicollinearity within the dataset. In doing so, 12 attributes were removed and only the remaining 19 (18 + 1) attributes were considered for model fitting.

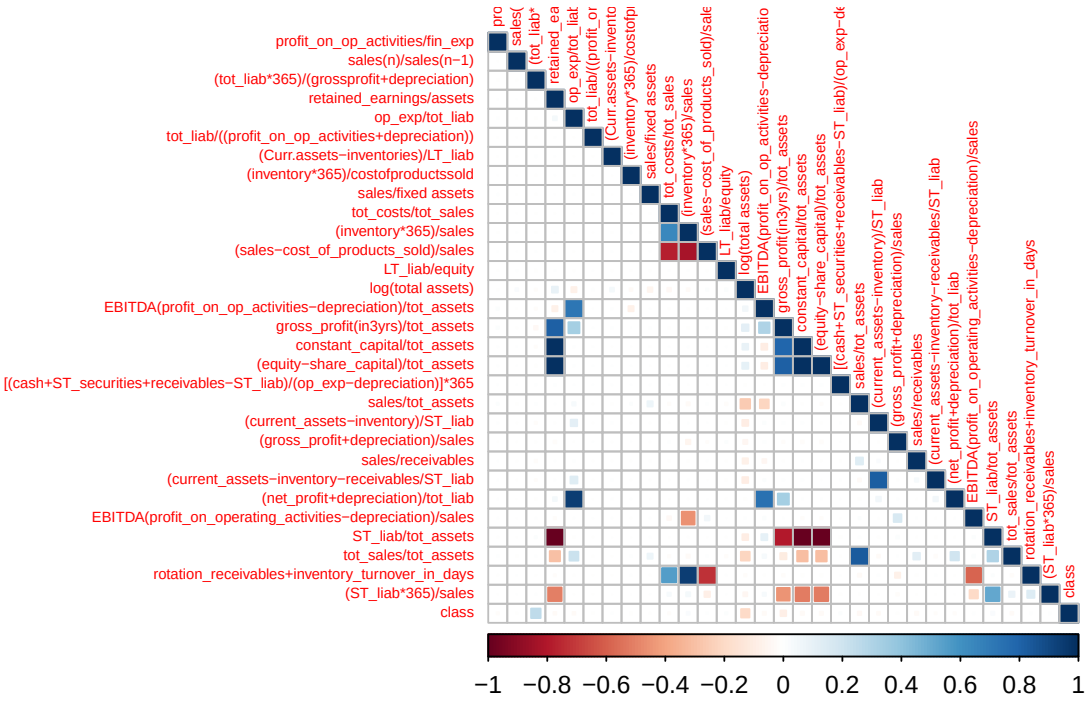

Figure 4: Correlation Matrix

3.2.5 Data Splitting (Train and Test Split)

I randomly split the data into training and testing sets consisting of 60% of the data, 40% of the data, respectively. In training data, we have a total of 6,300 instances(firms), out of which 2082 (33.04%) represents bankrupt companies, and 4218 (66.95%) firms that did not declare bankruptcy in the forecasting period. In test data, we have 4201 instances(firms), 1382 (32.89%) representing bankrupt companies, 2819 (67.10%) firms that did not go bankrupt in the forecasting period. This is depicted in Figure 5.

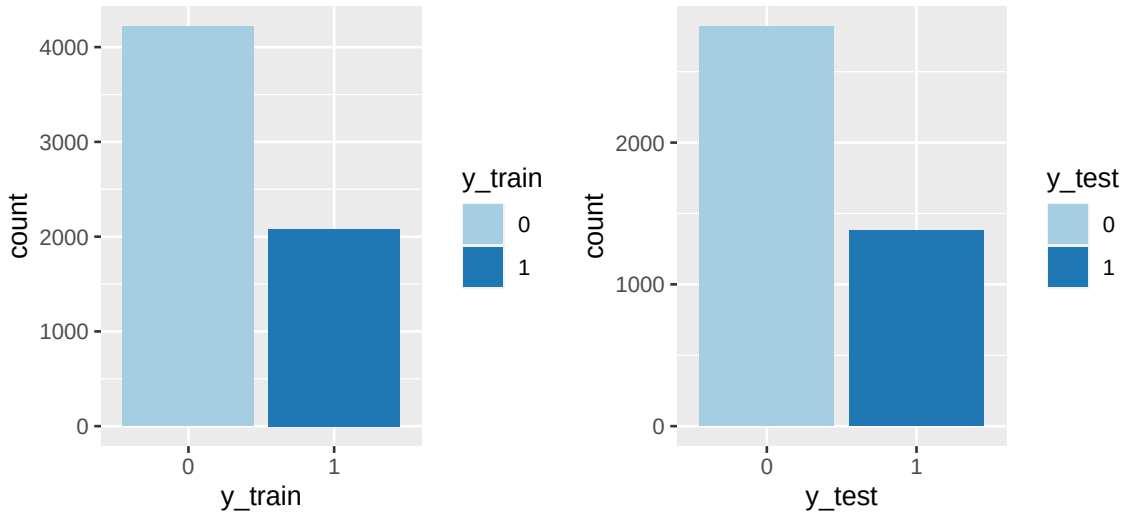

Figure 5: Training and Testing split

4. Machine Learning Models Fitting

The primary objective of this research is to predict whether or not a company will go bankrupt based on attributes taken from its financial statements. As a result of the dependent variable being binary, we are faced with a supervised classification problem. In order to obtain the best prediction, various machine learning models are applied to training and validation data. The optimal model will be chosen based on feedback from each model's performance on the validation data. This paper uses 5 separate Machine Learning Models: Decision trees, Random Forests, K-Nearest Neighbor (KNN), Logistic Regression, and Quadratic Discriminant Analysis (QDA). The confusion matrix (for accuracy rate) and the Receiver Operating Characteristics (ROC) curve (for measuring area under the curve) are used as statistical metrics to measure and compare the performance of each model. The performance of the classifier can be visualized using a confusion matrix where the number of true positives, false positives, true negatives and false negatives are listed.

The ROC curve indicates how well the probabilities from the positive classes are separated from the negative classes. The ROC curve tells us how good the model can distinguish between two classes, in our case declaring bankruptcy or not declaring bankruptcy. Thus,

we want the ROC value to be large and to assist us in determining how good the model is, the Area Under the Curve (AUC) is calculated. The larger the value for AUC, the better the model is at classifying.

4.1 Logistic regression

The first method applied is a simple logistic regression. The Logistic Regression model is a statistical model that uses a logistic function to model a binary dependent variable that in this case contains the values of "1" or "0". The coefficients are estimated using maximum likelihood estimation (MLE) and careful consideration is focused on this property. For obtaining the accuracy rate, the threshold for categorizing the bankruptcy status has to be decided. Figure 7 is a graph of Cost of Training data vs. threshold value and the value with lowest cost as the threshold for classifying the bankruptcy status is chosen. The ratio of weightage for False Positive and False Negative is 1:1, which implies that wrongly predicting the companies which will go bankrupt as "Not-Bankrupt" involves the same risk as predicting the vice versa. The optimal cutoff is 0.41.

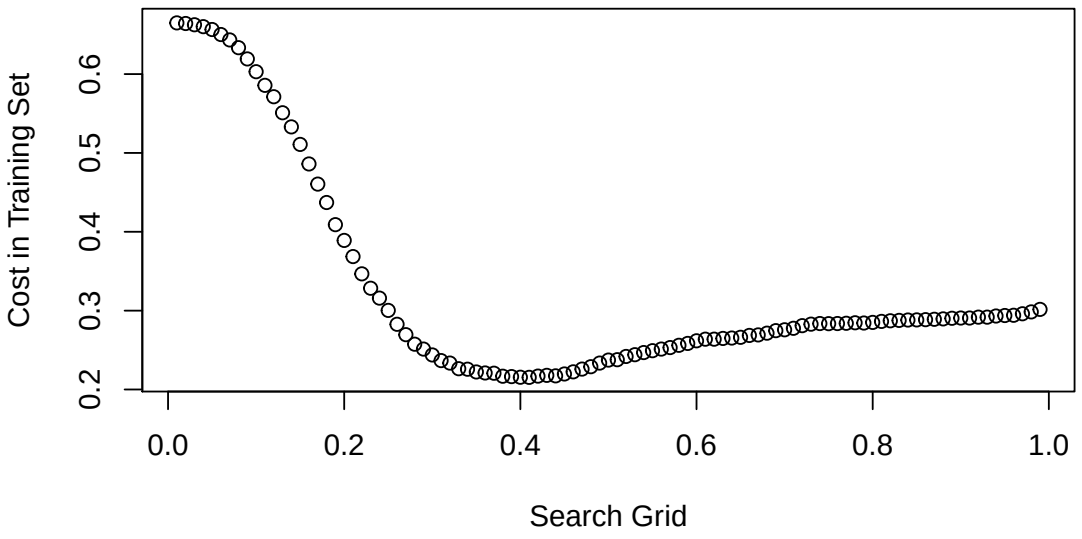

Figure 6: Assymetric cost Function

To gauge the model performance on the testing data, the bankruptcy status for the companies present in the test dataset using the final Logistic Regression Model will be predicted. We check for the accuracy rate and the Area under the Curve using the testing dataset. The testing dataset contains 4201 observations. We obtain a confusion matrix with an accuracy of 78.41%.

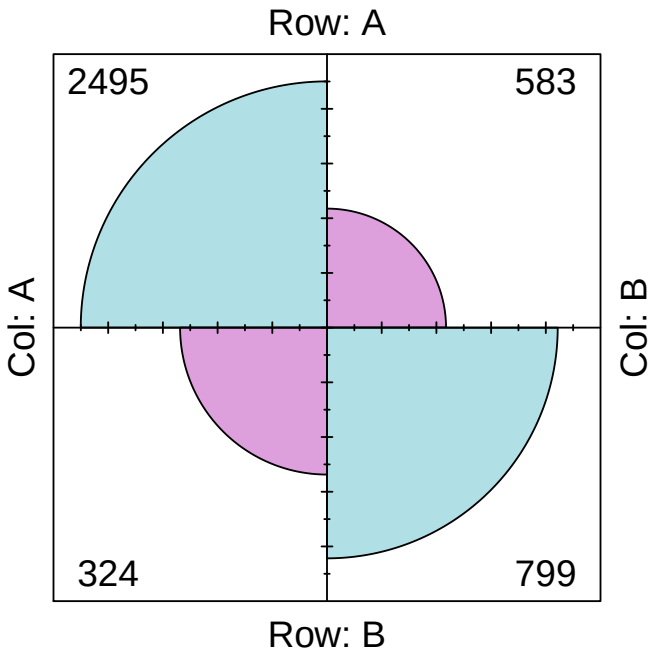

Figure 7: Confusion Matrix: Logistic Regression

The larger the value for AUC, the better the model is at classifying. Based on the results derived above, the model gives an AUC of 83%.

The Binomial Logistic model is tested using a 10-Fold Cross Validation approach because there are concerns about introducing bias due to the data partition that was carried out. It is found that the accuracy derived was 76.93%. This is a small dip compared to 78.41% accuracy that was derived from our prior analysis which seems to indicate a decent accuracy rate for the model.

4.2 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) estimates separate variances/covariances for each class in our dataset. Since QDA allows for different variances among the classes in the dataset, the resulting boundaries become quadratic. In other words, the predictor variables are not assumed to have common variance across each of the k levels in the variable, Y. QDA works best for datasets that have different variances for each class and it is assumed there are enough observations to accurately estimate the results. QDA is particularly useful if there is prior knowledge that individual classes exhibit distinct covariances.

After fitting a QDA model to the training data, I wanted to observe the accuracy that the model would derive based on using the testing dataset that was created earlier in the analysis.The accuracy derived was 37.25%. This statement is supported by observing the Kappa statistic of 0.0387. Kappa statistic that is closer to 1 implies there is a perfect agreement in terms of predictability. Another metric we can check is the AUC for the ROC. The AUC for QDA is 52.9%.

A 10-Fold Cross Validation was performed in order to address any bias that was introduced when the data was partitioned. After performing the 10-Fold Cross Validation, an accuracy of 37.38% was derived, which was marginally better than the accuracy derived from the training data. Again, the Kappa statistic is 0.03, which means that the model's predictability is close to being pure randomness.

4.3 K- Nearest Neighbors (K-NN)

The K-Nearest Neighbors (KNN) is a non-parametric method that is used for classification. The K-NN uses nearby observations to classify if a new observation would be identified near similar observed observations. This is done by designating one parameter, K, which represents the number of nearby observations or "neighbors" that will be used to classify a new record or in other words observation. Typically, the measured distance when using the KNN is the Euclidean distance. In order to select the parameter, "k" in KNN, cross-validation is carried out using the training data. Before doing the same, I chose k = "1" to

see what results does the model give. We get an accuracy of 86.71 %.

In order to select the parameter, "k" in KNN, cross-validation is carried out using the training data. After running the 10 fold cross validation KNN algorithm based on accuracy measure, we have obtained the optimal K to be 9 as seen in Figure 8. For the model with the optimal k, we acquired an accuracy of 86.24%. There is a great improvement in the prediction accuracy from the KNN model as compared to the previous models.

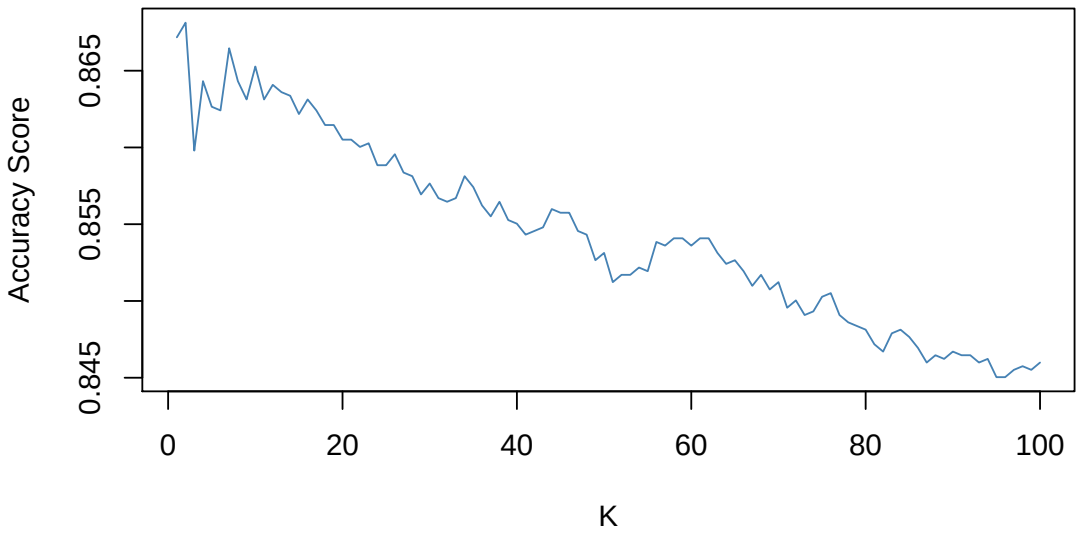

Figure 8: Deriving the Optimal K Value

4.4 The Decision Trees

Decision Trees is a widely used supervised learning algorithm for solving classification and regression problems (Landgrebe; 1991). While shrinkage methods reduce the number of predictors by applying a tuning parameter, classification trees will create optimal splits on predictors that allow for simple and transparent interpretation. It has an individual tree representation wherein each leaf node represents a class label and attributes correspond to the internal node of the tree. It starts with the training data as the root and then breaks

down into smaller subsets having decision nodes and leaf nodes. Identifying the attribute for root node in each and every level is the challenging part and the two attribute selection measures that are commonly used are Information gain and Gini index. Based on the learnings from previous literature, this algorithm has performed well on the bankruptcy prediction problem hence we will evaluate the performance of this model on our set of variables.

Standard Decision Trees

In order to visualize these splits, we begin by constructing a default tree with low complexity (complexity parameter = 0) as illustrated in the figure below. This default tree has a total of 14 splits that we use to predict bankruptcy classes within the testing set.

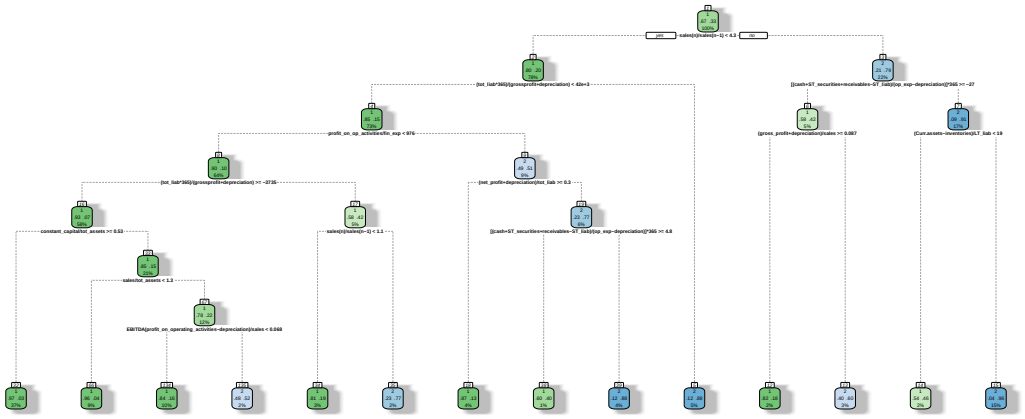

Figure 9: Standard Decision Tree

By doing this, we get an accuracy of 88.53%, which outperforms all three methods used to this point. To confirm the strength of the model on predicting bankruptcy, we will again look at precision and recall rates. The model yields a precision rate of 84.72% and a recall rate of 79.45%, making this the best model thus far. We obtain an AUC of 86.2%.

Pruned Decision Trees

Given that we have a low complexity parameter, we could be concerned with our model overfitting the data. This could result in a poor fit to the testing data due to a high level of variance within the model. For this reason, we 'prune the tree' by iterating through the depth of the tree grown using the 10-Fold Cross Validation method. In the table below, we can see complexity parameters associated with their cross-validated errors. Given the minimum tree error (0.37080), we use the associated complexity parameter of 0.010000 to prune the tree.

	CP	nsplit	rel error	xerror	xstd
1	0.387608	0	1.00000	1.00000	0.017933
2	0.111912	1	0.61239	0.61335	0.015326
3	0.044428	2	0.50048	0.50432	0.014208
4	0.025937	4	0.41162	0.42123	0.013197
5	0.018252	5	0.38569	0.40346	0.012959
6	0.016811	6	0.36744	0.38905	0.012761
7	0.010000	8	0.33381	0.36984	0.012487

The new tree is displayed in Figure 10.

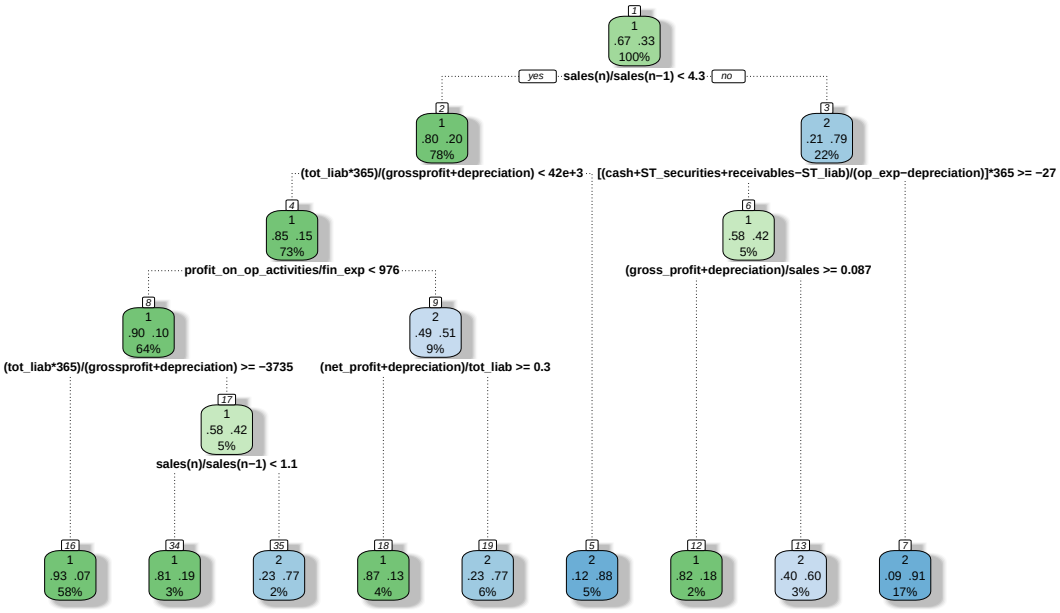

Figure 10: Pruned Decision Tree

As it can be seen, the tree is significantly smaller than the original default tree. We get an accuracy of 87.91%, one percentage point lesser than the standard decision tree. The AUC is 85.8%.

Best Pruned Decision Trees

One way to simplify the tree further is by using the 'best pruned tree' method. This method requires choosing the complexity parameter that falls within one standard deviation of the minimum tree error. From the table above, I chose the complexity parameter that is less than but closest to 0.383301 (0.37080+0.012501). Therefore, I chose the complexity parameter of 0.387608 to further prune the tree and get the Figure below.

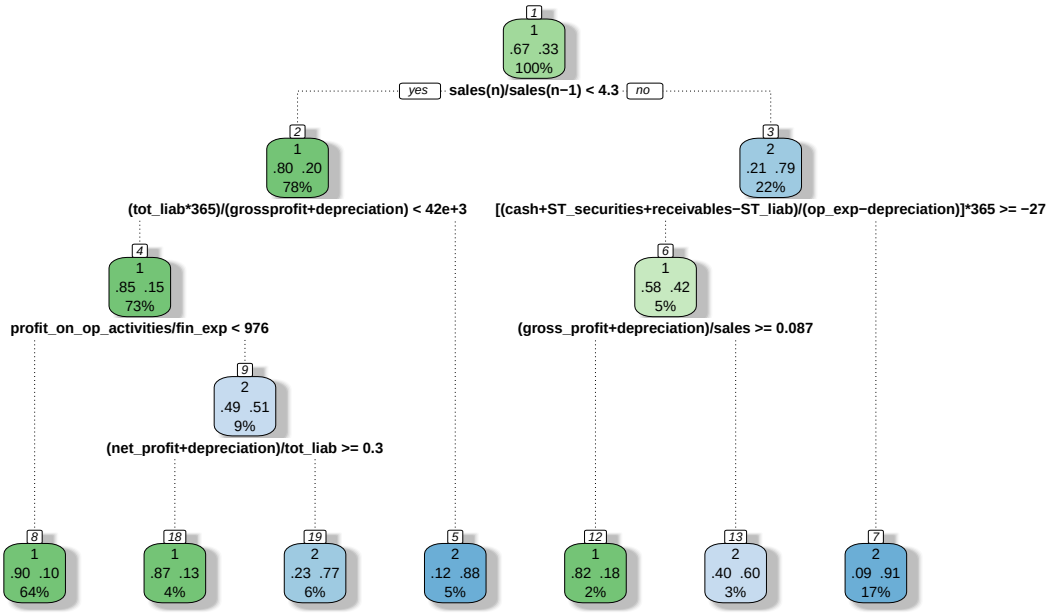

Figure 11: Best Pruned Decision Tree

Although this tree solves the overfitting issue of our original tree, we get lower accuracy rates. Our accuracy rate is 86.96%.

4.5 Random Forests

A random forest model is an ensemble of many decision trees and is often used in classification problems. It uses techniques such as feature randomness and bagging for building each tree such that an uncorrelated forest of trees is obtained (Khoshgoftaar; 2007). Each of the trees relies on an independent random sample. The prediction performance of this collection of trees is more accurate than the individual tree. Few of the factors that make it a suitable choice for the chosen data set include the quick training speed of the model, being robust to outliers and the ability to handle unbalanced data. To further improve the analysis, I used the Random Forest method. This method allows me to reduce the variance without sacrificing bias by randomly sampling a subset of predictors for each tree and then averaging the results. For this paper, I selected 500 trees and computed the relative influence plot seen in the figure below.

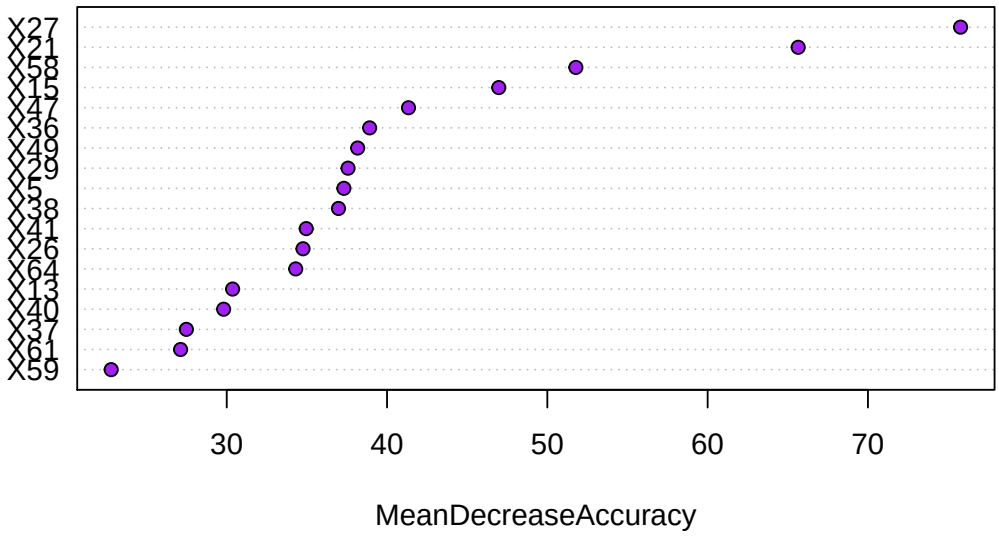

Figure 12: Relative Influence Plot

5. Conclusion

I have successfully modeled five classification models: Logistic Regression, Random Forests, k-Nearest Neighbors (k-NN), Quadrant Discriminant Analysis and Decision Trees. Below is a visual summary of how each model performs. Based on the accuracy and AUC as metrics, Random Forest algorithm performed the best with the highest accuracy of 95.26% as depicted in the figures below.

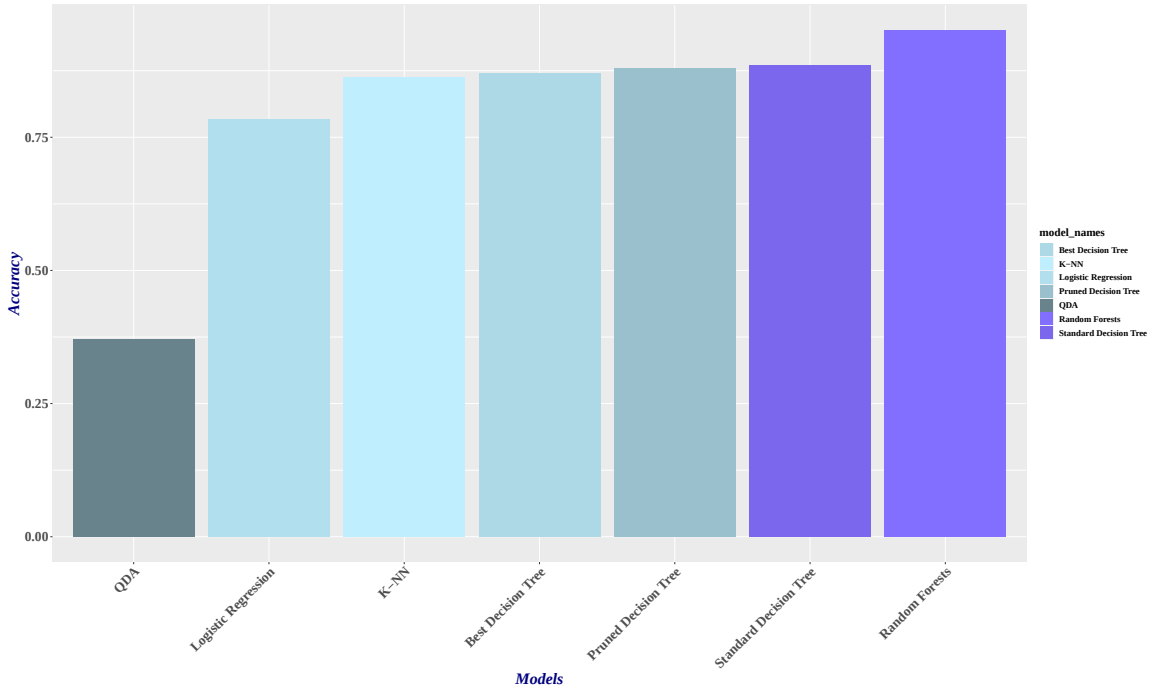

Figure 13: Model Comparison

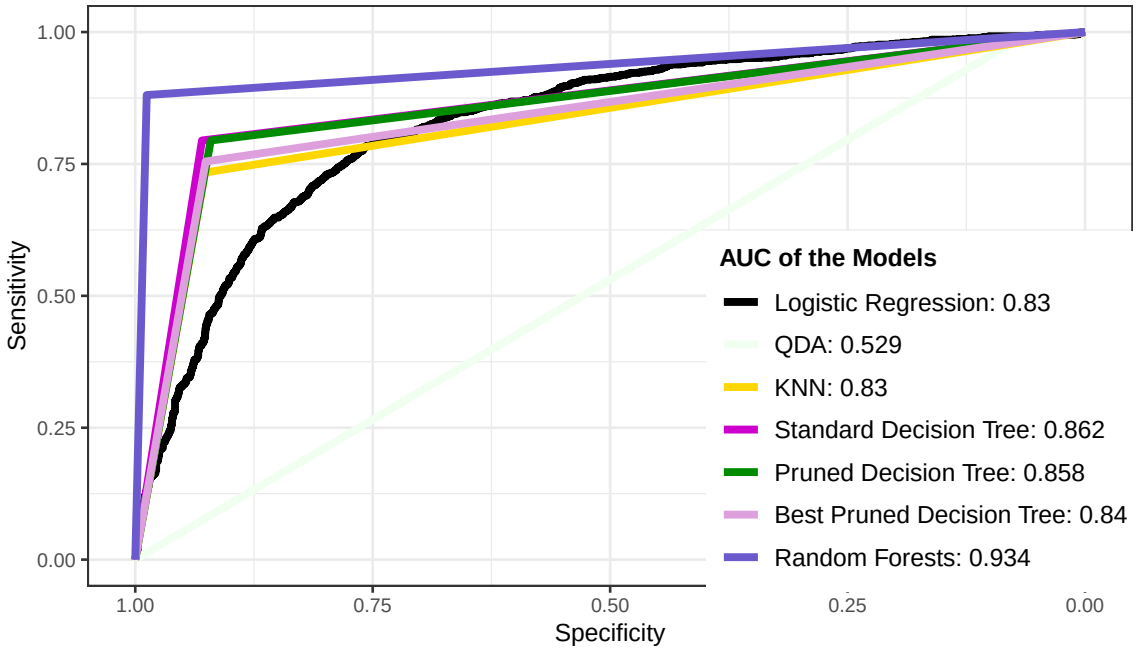

Figure 14: Model Comparison: ROC Curves of the models

I implemented not only machine learning techniques into the model creation, but also followed econometric techniques in the attempt to integrate both machine learning and econometrics into a field that has been growing ever since the introduction of using rigorous machine learning techniques to explain this phenomena.

The analysis produced strong results, but careful consideration needs to be used when selecting the features that should be included in the model because of concerns of overfitting and econometric issues like multicollinearity. A major challenge was data imbalance that dealt with using SMOTE analysis. I have also imputed the missing values in the data using two imputation techniques: Mean Imputation and Missforest. The features on which the bankruptcy prediction is based are not as straightforward as the financial ratios found on the balance sheets of the companies and need to thoroughly be studied and validated. The challenging part, as discussed, lies in selecting the best financial attributes that are responsible for a company declaring bankruptcy. With the vast number of features available, feature selection method was then performed in order to narrow down the features that

would create models with high accuracies and remove variables that were considered to be unimportant as well. After that, I went through the process of further eliminating features because of econometric issues like multicollinearity. I also checked for outliers in the data set.

My aim was not only to further understand the study of bankruptcy predictions, but perhaps contribute to the study itself. By having the ability to anticipate bankruptcies in advance, economic downturns could be mitigated or lessened in magnitude by lending institutions involving themselves in corporations before further problems arise.

References

Altman, E. I., & Hotchkiss, E. (2010). Corporate financial distress and bankruptcy: Predict and avoid bankruptcy, analyze and invest in distressed debt . John Wiley & Sons.

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. Expert Systems with Applications, 83, 405–417. https://doi.org/10.1016/j.eswa.2017.04.006

Krulick, A. (2018). Bankruptcy Statistics. Debt.org. https://www.debt.org/bankruptcy/statistics/

Laitinen, E. K. (1991). Financial ratios and different failure processes. Journal of Business Finance & Accounting, 18.

Mandel J, S. P. (2015). A Comparison of Six Methods for Missing Data Imputation. Journal of Biometrics & Biostatistics, 06(01). https://doi.org/10.4172/2155-6180.1000224

Tharwat, A. (2018). Classification assessment methods. Applied Computing and Informatics. https://doi.org/10.1016/j.aci.2018.08.003

Tomczak, S. (2016). Polish companies bankruptcy Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Polish companies bankruptcy data

Wang, N. (2017). Bankruptcy Prediction Using Machine Learning. Journal of Mathematical Finance, 07(04), 908–918. https://doi.org/10.4236/jmf.2017.74049

Zhang, W. (2017). Machine Learning Approaches to Predicting Company Bankruptcy. Journal of Financial Risk Management, 06(04), 364–374. https://doi.org/10.4236/jfrm.2017.64026

Appendix

Table 1: Data Description

Index	Attribute_Names
X1	net profit / total assets
X2	total liabilities / total assets
X3	working capital / total assets
X4	current assets / short-term liabilities
X5	[(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365
X6	retained earnings / total assets
X7	EBIT / total assets
X8	book value of equity / total liabilities
X9	sales / total assets
X10	equity / total assets
X11	(gross profit + extraordinary items + financial expenses) / total assets
X12	gross profit / short-term liabilities
X13	(gross profit + depreciation) / sales
X14	(gross profit + interest) / total assets
X15	(total liabilities * 365) / (gross profit + depreciation)
X16	(gross profit + depreciation) / total liabilities
X17	total assets / total liabilities
X18	gross profit / total assets
X19	gross profit / sales
X20	(inventory * 365) / sales
X21	sales (n) / sales (n-1)
X22	profit on operating activities / total assets
X23	net profit / sales
X24	gross profit (in 3 years) / total assets
X25	(equity - share capital) / total assets
X26	(net profit + depreciation) / total liabilities
X27	profit on operating activities / financial expenses
X28	working capital / fixed assets
X29	logarithm of total assets
X30	(total liabilities - cash) / sales
X31	(gross profit + interest) / sales
X32	(current liabilities * 365) / cost of products sold
X33	operating expenses / short-term liabilities
X34	operating expenses / total liabilities
X35	profit on sales / total assets
X36	total sales / total assets
X37	(current assets - inventories) / long-term liabilities

Index	Attribute_Names
X38	constant capital / total assets
X39	profit on sales / sales
X40	(current assets - inventory - receivables) / short-term liabilities
X41	total liabilities / ((profit on operating activities + depreciation) * (12/365))
X42	profit on operating activities / sales
X43	rotation receivables + inventory turnover in days
X44	(receivables * 365) / sales
X45	net profit / inventory
X46	(current assets - inventory) / short-term liabilities
X47	(inventory * 365) / cost of products sold
X48	EBITDA (profit on operating activities - depreciation) / total assets
X49	EBITDA (profit on operating activities - depreciation) / sales
X50	current assets / total liabilities
X51	short-term liabilities / total assets
X52	(short-term liabilities * 365) / cost of products sold)
X53	equity / fixed assets
X54	constant capital / fixed assets
X55	working capital
X56	(sales - cost of products sold) / sales
X57	(current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
X58	total costs /total sales
X59	long-term liabilities / equity
X60	sales / inventory
X61	sales / receivables
X62	(short-term liabilities *365) / sales
X63	sales / short-term liabilities
X64	sales / fixed assets)