STAT 471 FINAL PROJECT

Predicting Imminent Bankruptcies with Financial Statements

By Harrison Beard, Mingxian Gao, and Walter Kobasa

Implementing Modern Statistical Techniques on a Sparse Data Set to Build an Effective Prediction Model

December 16, 2018

Contents

Executive Summary

In the world of finance, managers are trying to integrate big data to make better decisions. Our analysis on predicting whether a company is at risk of going bankrupt over the next five years is of serious interest for a wide variety of groups. To outline some of the groups who would find this analysis beneficial to their daily work is as followed:

1. **Top executives at a firm.** CEOs and CFOs have the best insight into the firms that they run because they manage the day-to-day operations as well as hold information that is not publicly available. We believe their interest in a model that accurately predicts bankruptcy is small because out of all the people familiar with a particular company, the CEO and CFO should know the best if their company faces bankruptcy in the next five years.
2. **Credit Rating Agencies.** Credit rating agencies such as S&P and Moody's are tasked with determining how likely every company is to pay back their debt to bondholders. These agencies are constantly trying to provide better insights about the "credit-worthiness" of a company to investors. However, as recent history has shown (2008/2009) they are not perfect. People often critique that they are slow to react to fundamental changes. Any credit agency should be interested in any tools that they could use to better predict credit-worthiness.
3. **Asset Managers/Investors.** This group should be most interested in the results of this analysis. Large hedge funds and asset managers have really started to embrace quantitative finance in recent years to provide whatever edge they can get over the other firms. As such, firms like Point72, AQR, and Citadel have all risen in prominence. There are many strategies investors can execute on if they were able to better predict a company at risk of bankruptcy in the near future than their competitors. Large mutual funds who hold debt would want to sell their bonds if a company was at risk of bankruptcy. Hedge funds could short the company's stock as equity is typically wiped out when a company files for bankruptcy. A hedge fund could also buy a credit default swap, or CDS, which is a derivative that pays the investor if a company defaults on their debt, like an insurance policy.

Variable of Interest: Bankruptcy We will use fundamental financial data from 2010 to predict whether each company will file for bankruptcy protection within the next five years (2011 through 2015, inclusive). In our "`Bankrupt`" response variable, we include Chapter 7 (liquidation) and Chapter 11 (reorganization) bankruptcies because we are not concerned about the results of the company after the bankruptcy process, just whether it declares bankruptcy.

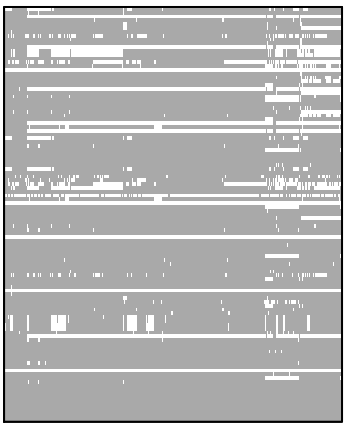
A bankruptcy is when a company cannot repay their obligations. In most cases, these obligations are the repayment of debt. More broadly defined, a company is forced to file for bankruptcy if their total liabilities are greater than their assets.

Cover Art: "New York City, Manhattan, Lower Manhattan, Financial District, Alexander Hamilton U.S. Custom House, 1901-1907. 1 Bowling Green." by (vincent desjardins). Licensed under CC BY 2.0 via Flickr Creative Commons.

Exploratory Data Analysis

Cleaning the Data

Raw Data **Before Imputation**



The plots above highlight the missing values from the raw data set on the left and the semi-cleaned data set on the right (after removing many highly correlated and unnecessary features, as well as other more detailed cleaning procedures outlined below). The y-axis on each plot are the individual features (not labeled here because it would be too cluttered), and the x-axis are the indices. White represents a missing data point. Thus, an entire row of white indicates a feature whose every observation is missing. Note that the data set on the right is exactly as our data appears before we begin the multiple imputation process.

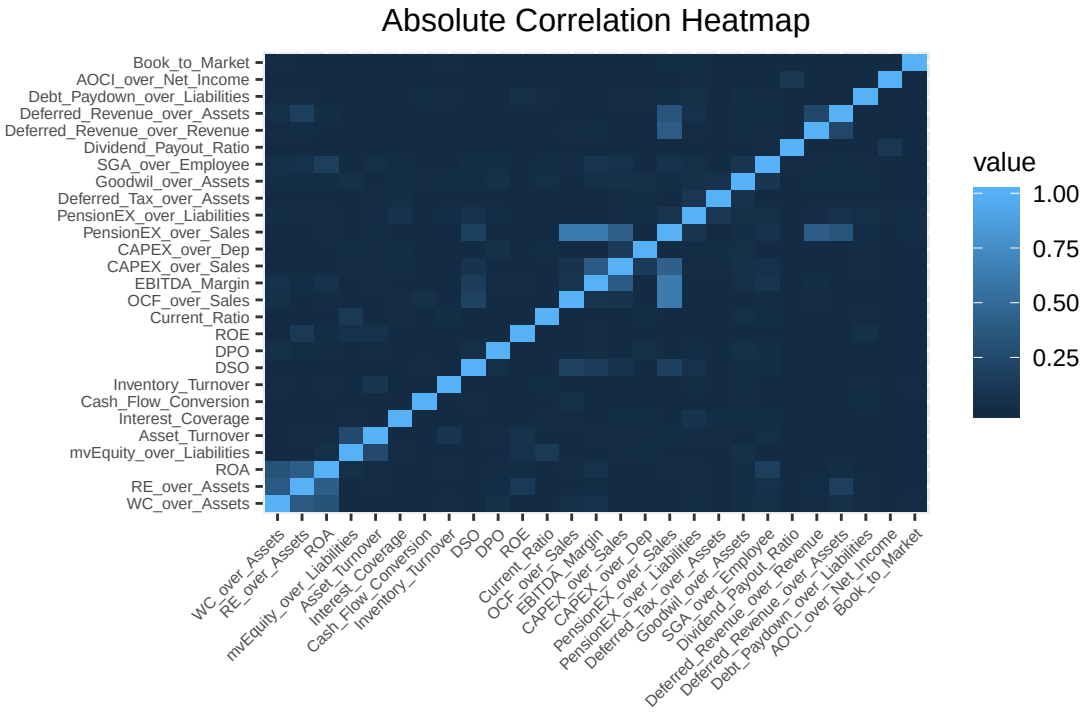
Our data set had an uncanny amount of missing values. In fact, every single row had at least one missing value, and some of the columns were missing values for almost all of the rows. And, since the data were mostly composed of startup companies without much of an established history (and some entirely without debt, for example), much of the filled-in fields were zero. This presented issues in our ratio analysis, since we ended up with an enormous amount of "`NA`", "`NaN`", or "`Inf`" entries. To resolve this enormous issue with our data set, we conducted the following procedures:

1. First, we removed all entries with "`NA`" entries for debt and net sales, and then replaced all the entries with zero debt and net sales to instead have "`NA`" values for debt and net sales.
2. Then, we converted all the existing (positive or negative) infinite values to "`NA`" values.
3. Next, we completely omitted a few ratios from the data set that had a vast majority of "`NA`" values.
4. We omitted four ratios that were highly correlated to one another to avoid collinearity issues in our regressions.
5. Next, we used the `mice` package to perform multiple imputation on the remaining "`NA`" values.
6. Finally, we removed a total of 74 observations that were extremely severe outliers (identified visually, usually in a very obvious way — some of the outliers lay almost 100,000 times the inter-quartile range away from the median) across at least one feature. Of these 74 observations, 25 were bankruptcies. In other words, about 34% of these severe outliers went bankrupt within the next five years, versus roughly 3% of the remaining data points that went bankrupt within the next five years. This may be cause for concern, since we may be removing valuable and telling outliers that could result in more helpful regressions. However, we think that this is a good thing, since:
    - many of these outlier observations were outliers just for a one or a few features, and there were extreme outliers across nearly all the features, suggesting that the companies that went bankrupt

among these outliers were mostly bad companies due to one or two very outlandish financial ratios that doomed the company, which is idiosyncratic to each company and not representative of systematic effects across the entire pool; and

- in the real world, investors can easily identify "outlier" companies without the aid of a computer; the value-add of "quantamental" statistical models is that they can help investors make inferences that are not at all obvious from face value.

The following plot illustrates the pairwise cross-correlations across all our numerical features after removing highly collinear features, but before the imputation of missing values.



Explanation of Variables

We downloaded a large set of fundamental financial data from Capital IQ to do our analysis. Because companies vary greatly by size, we want to use financial ratios to build are model. This standardizes the data and removes the need for us to use a log scale on dollar-denominated numbers. Often, these ratios are limited to a range of numbers. For example, ratios that are represented as a percentage of sales should be limited from 0 to 1. In total, we were able to calculate 38 sensible ratios provided by the financial metrics available to us. We believe each ratio we included makes intuitive sense in how a higher or lower ratio could be an indicator of a company that has financial trouble and may enter bankruptcy in the near future.

While an explanation of each variable and an explanation of the intuition on how it could help paint the picture of a company's financial health would be extensive, and ultimately outside of the scope of this statistics class, information about these metrics are readily available online. Therefore, we choose to highlight a select few.

Dividend Payout Ratio: is defined as total dividends/net income. A company paying dividends is a sign of financial health, therefore we believe a company with a high dividend payout ratio is less likely to go bankrupt. If the company where to run into financial trouble they could cut the dividend to pay of debt, so the fact that they are paying a high dividend should be seen as a positive for a company that isn't likely to go bankrupt.

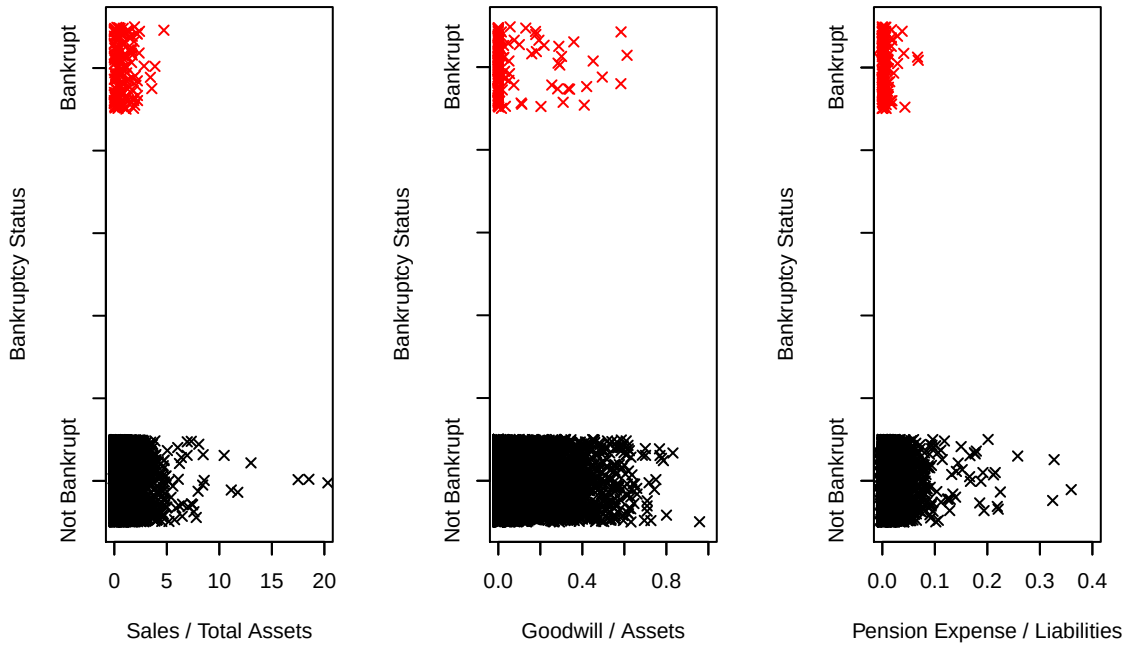
Pension Expense as a percentage of total liabilities is defined as pension expense/total liabilities. In the U.S.,

pensions are often times underfunded leading to huge pension liabilities on the balance sheet. To counteract this, companies are forced to have higher pension expenses as they have to put money into their employee's retirement accounts. A small pension expense as a percentage of total liabilities could highlight a company that has high liabilities (underfunded pensions!) or a company that has to pay a lot of money to fund their pension, both not good things for a company that doesn't want to go bankrupt.

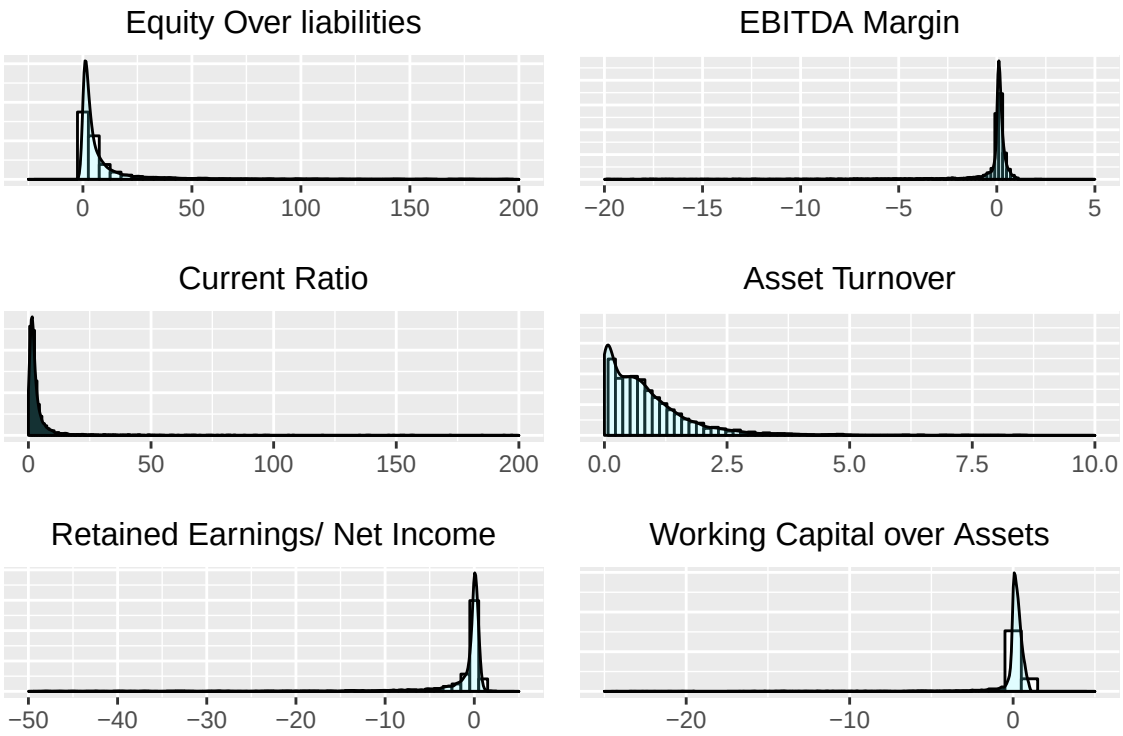
We also include the five ratios that are used in Altman's Z-score, the famous NYU professor's methodology to predict the probability that a firm will go bankrupt within two years. These five ratios are Working Capital/Total Assets, Retained Earnings/Total Assets, EBIT/Total Assets, MV of Equity/Total Liabilities, and Sales/Total Assets. Altman Z-score uses profitability, leverage, liquidity, and solvency ratios to determine the probability of a firm going bankrupt.

Summary of Cleaned Data

The jitter plots below highlight some of the distinguishing features of the bankrupt companies.



To demonstrate key variables, we used both jitter plots and histograms. We first selected three variables and used jitter plots to show the differences of fundamental variables between bankrupt and non-bankrupt companies. From the plots, we can see that bankrupt companies have a significantly lower asset turnover (sales over total assets) ratio, which means that they were not able to sell their products efficiently to generate income. While bankrupt companies seem to have a wide range of values for the goodwill over assets ratio, a large proportion of these bankrupt companies still concentrate on the lower end. An interesting observation is that bankrupt companies have extremely low pension expenses over liabilities ratios. We are not sure why this is, but this phenomenon seems to persist across most of our analysis for the rest of the paper.



We then selected six fundamental variables and plotted out histograms for each to visualize the outliers and resultant skew remaining in our cleaned data set. Indeed, we discovered that there are still many outliers: some companies have extremely negative EBITDA margins and retained earnings over net income, yet there are also some very healthy companies that are safe from bankruptcy because the distribution of fundamental metrics such as the equity over liabilities, current ratio, and asset turnover ratio all indicate outliers on the right side of the distribution. Although we will not present them here, boxplots and statistical summarize confirm that many of our remaining features contain a considerable amount of outliers. However, we elect to keep these outliers in the data set, since they are not nearly as severe as the ones we omitted earlier in our data cleaning, and the existence of the remaining outliers is helpful for conducting inference on the bankrupt companies. Any more removal of outliers would be irresponsible data mining, we figured.

Statistical Modeling

Logistic Model Selection via BIC

First, we selected a logit-link logistic regression model using the Bayesian Information Criterion. Note that we are not interested in Mallow's C_p because our model is a *logistic* regression instead of a traditional OLS linear regression (and its generalization AIC is not easily available using the R packages at our disposal). We will also not consider adjusted R^2 because of the lack of theoretical motivation. Ultimately, we are interested in parsimony, and BIC will help us best achieve that. BIC is a criterion for model selection derived through Bayesian methods (with a prior that all models are equally likely of being true). It is based on the likelihood function but penalizes very harshly for the number of predictors. Generally, adding predictors reduces RSS, but results in overfitting. BIC therefore adds a penalty for each predictor that is added to the model and usually ends up producing smaller models. Precisely, the BIC for our purposes is defined as

$$\text{BIC} \triangleq N \log\left(\frac{\text{RSS}}{N}\right) + d\log(N),$$

where N is the sample size, RSS is the residual sum of squares, d is the number of predictors in the model, and the logs are natural logarithms (this will be the case for the remainder of this article). A more general

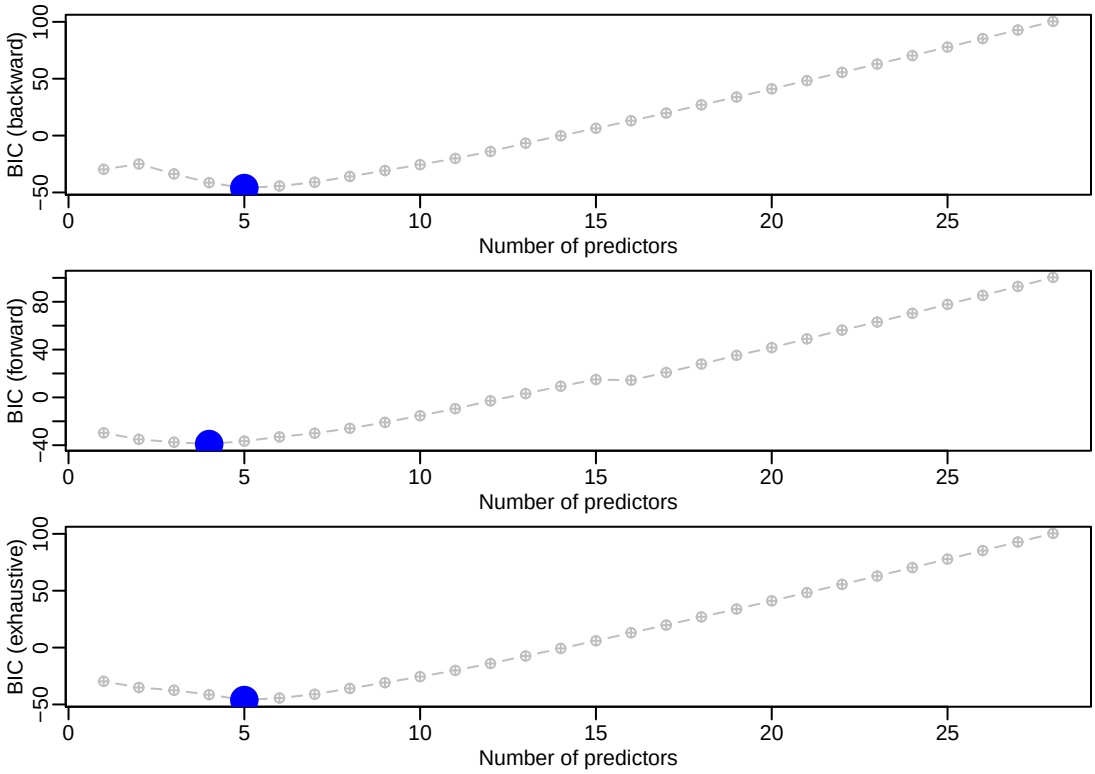
formulation is

$$\text{BIC} = -2\log\left(\widehat{\mathcal{L}}\right) + d\log(N),$$

where $\widehat{\mathcal{L}}$ is the maximized likelihood function.

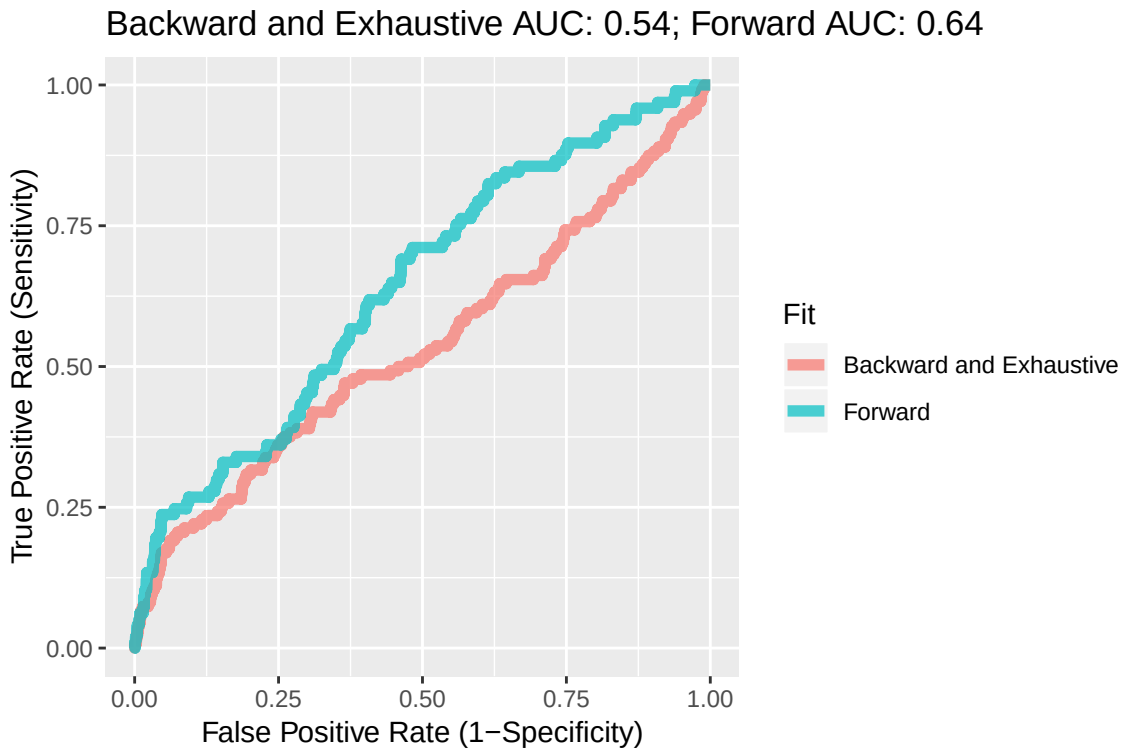
One method of finding the best model is to minimize BIC across all values of d, where each model with $d \in \{1, \ldots, K\}$ predictors is determined via minimizing RSS. We will minimize BIC across all values of d using backward, forward and exhaustive selection and then compare the ROC curves of the three resultant classification models below.

One important note of consideration is that we are using the *pre-imputed* data in this section only to make inference. The reason is that the imputed data leads us to select trivial single-variable models according to BIC. Thankfully, the functions `regsubsets` and `glm` are built to be able to handle missing values.



As we can see, the number of predictors that minimize BIC for the backward-selected RSS values across all d is five, (identical to the number of variables for exhaustive selection) versus four for forward selection. Incidentally, the five variables in the exhaustive and backward selection are identical. ROE, capex, and goodwill over assets are the common variables among all three models.

Looking at the testing ROC curves and corresponding areas under the curves (AUC), we have the following:



The AUC values for the forward-selected model are more satisfactory, and the forward-selected model with fewer predictors edges out the backward-selected model. However, in the next few parts of the analysis, we will consider other models and compare their testing AUC to that of the forward-selected model above and see which one ends up winning out.

LASSO

Theoretical Framework

What we want is to estimate $\mathrm{P}(y = 1)$, where y is the binary variable indicating whether or not a company went bankrupt within the next five years. We can estimate this probability via maximum likelihood estimation. That is, we want to select the model parameters such that the likelihood function corresponding to our data is maximized. We make the assumption that the data-generating process takes the form of a logit-link model with independent observations, so that we can represent the likelihood simply as the product of the marginal densities corresponding to each of the observed data points. Put more precisely, for a vector of parameters $\boldsymbol{\beta}$, our fitted parameters $\hat{\boldsymbol{\beta}}^{\mathrm{MLE}}$ are defined as

$$\hat{\boldsymbol{\beta}}^{\mathrm{MLE}} := \arg\max_{\boldsymbol{\beta}} \left\{ \mathcal{L}\left(\boldsymbol{\beta} \mid (\mathbf{y}, \mathbf{X})\right) \right\},$$

where $(\mathbf{y}, \mathbf{X})$ are our observed data, and $\mathcal{L}\left(\boldsymbol{\beta} \mid (\mathbf{y}, \mathbf{X})\right)$ takes the form of a product of logit-link marginal density functions corresponding to $(\mathbf{y}, \mathbf{X})$, in terms of $\boldsymbol{\beta}$, as per our data-generating assumptions.

Note that our data source contains a lot of variables, and some of them may not be all that important, so we will start off by using model shrinkage techniques to select variables. Recall that the generally-used shrinkage technique called Elastic Net is represented as a weighted linear combination of LASSO and Ridge regressions, which each individually minimize deviance plus an added penalty, namely ℓ_1-loss for LASSO and ℓ_2-loss for Ridge. The LASSO-selected *coefficients* corresponding to underlying features can be solved via the definition

$$\hat{\boldsymbol{\beta}}^{\mathrm{LASSO}} := \arg\min_{\boldsymbol{\beta}} \left\{ -\left(\frac{\log\left(\mathcal{L}\right)}{N} \right) + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\},$$

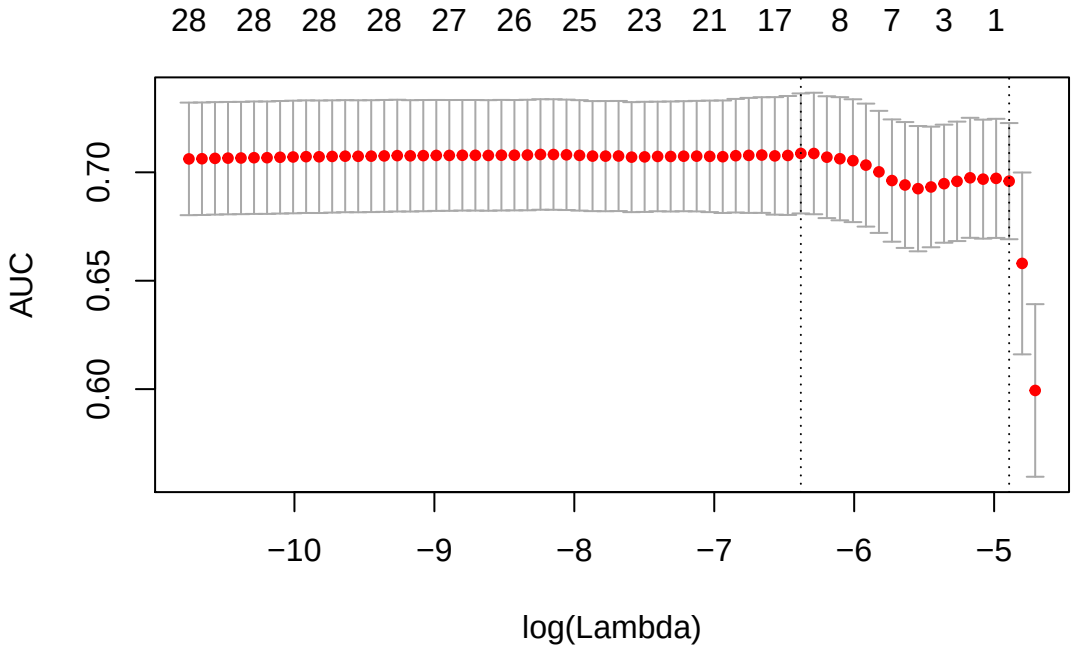
where the combined penalty for Elastic Net takes the form

$$\left(\frac{1-\alpha}{2}\right)\|\boldsymbol{\beta}\|_2 + \alpha\|\boldsymbol{\beta}\|_1,$$

where we will employ cross-validation across various levels of α to determine appropriate α and λ values for the Elastic Net regression.

Model Selection

We will now turn our attention to LASSO in our attempts to find the best model.



We performed a LASSO with 10-fold cross-validation with the intent to maximize AUC. The `lambda.min` value that corresponded with the highest AUC led to a model with 11 predictors. They are as followed: `DebtZero`, `mvEquity_over_Liabilities`, `Asset_Turnover`, `DSO`, `ROE`, `Current_Ratio`, `CAPEX_over_Dep`, `PensionEX_over_Liabilities`, `Goodwil_over_Assets`, `Deferred_Revenue_over_Assets`, `Book_to_Market`.

We will manually perform backward selection until all predictors are significant at the 0.05 level. This entails performing an Anova test and removing the predictor with the highest p-value. We subsequently removed `Asset_Turnover`, `DSO`, `Deferred_Revenue_over_Assets`, `mvEquity_over_Liabilities`, `ROE`, and `Book_to_Market` from our model. The final model included `DebtZero`, `Current_Ratio`, `CAPEX_over_Dep`, `PensionEX_over_Liabilities`, and `Goodwill_over_Assets`.

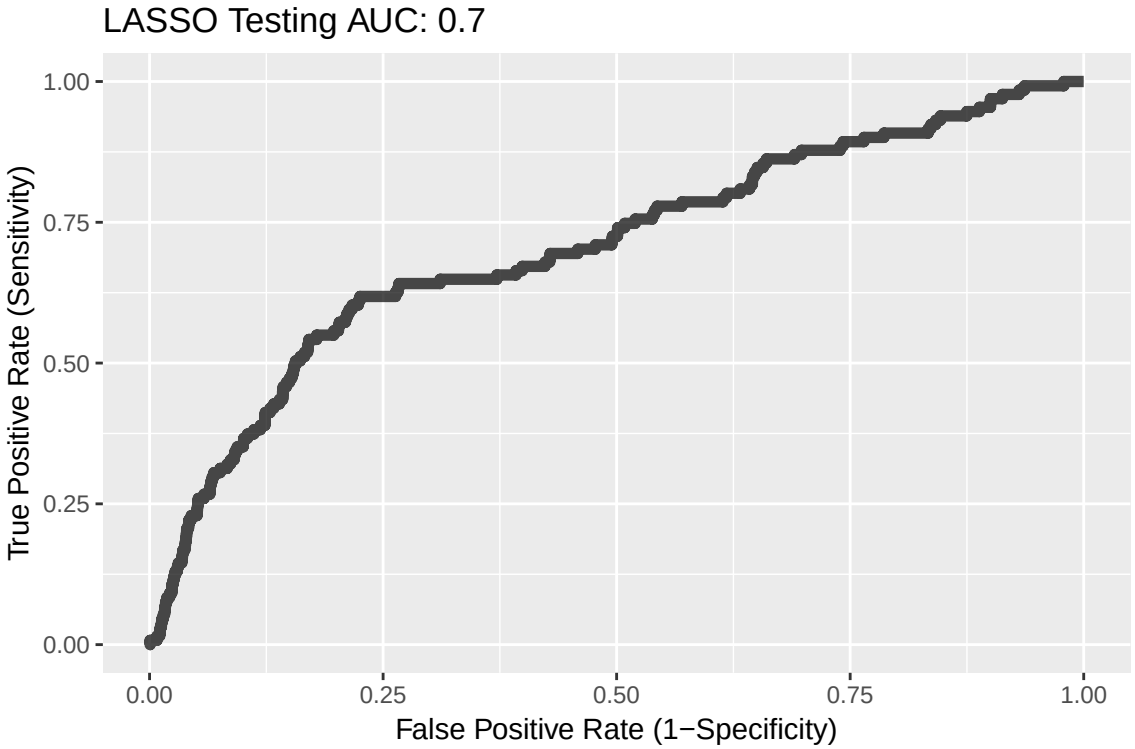
Increases to `Goodwill_over_Assets`, `PensionEX_over_liabilities`, `CAPEX_over_Dep`, and `Current_Ratio` all resulted in a lower probability of the company going bankrupt. These results also make intuitive sense.

1. `Goodwill_over_Assets`: the more goodwill a company has as a percentage of assets means the firm lacks significant collateral to secure that debt. This makes the company more at risk to default on their debt.

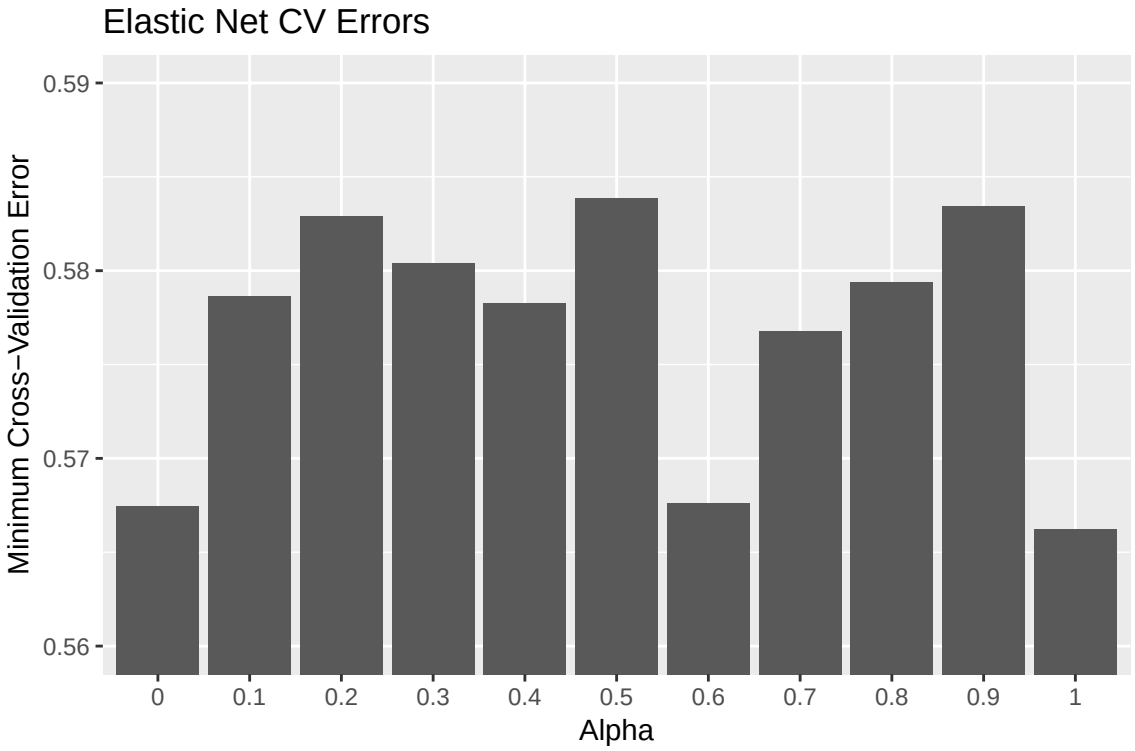
2. `PensionEX_over_liabilities`: Pensions are very similar to debt as they require annual commitments by the company. While many companies do not have large pensions, the companies that do often have underfunded pensions. This increases the company's obligation to their employees. The ratio scales the firm's pension expense by total liabilities, which is usually predominately debt. If the company has lower liabilities or debt the ratio is higher, meaning that the company is less likely to go bankrupt.

3. `CAPEX_over_Dep`: a higher productive asset investment ratio is a sign of strength for the company because they are increasing their investments today to reap the returns in the future. As a result, a higher ratio signals a healthier company; one that is less likely to default in the future, assuming that the company has a strong management team that is investing in capex wisely.
4. `Current_Ratio`: the current ratio is a liquidity ratio used to determine how quickly a company can convert its assets into cash. A higher ratio means that they have more cash-like assets that can be used to pay obligations that are due in the next 12 months.

The last predictor in the model is `DebtZero`: a categorical variable that is "0" if the company has debt and "1" if the company is debt-free. According to our five-variable model, on average, a company that has no debt has a higher probability of going into bankruptcy. This seems counterintuitive on the surface. However, the company could take on new debt within the next five years. If the company finances a bad acquisition with debt, it could result in them going bankrupt. Also, it's important to note that some small companies that might not have debt declare bankruptcy to close down their business to tie up all loose ends. Also, our data has nearly 10× more debt-free companies than ones with debt. In our data set, 154 out of the 5,750 debt-free companies went bankrupt. Meanwhile, eight of the 540 companies that had debt went bankrupt.



From our LASSO model we found a five-predictor model. From there we split the data into a training and testing set. The training set had 1,000 data points. We fit a logistic model of the five aforementioned variables on the training data set. Then we tested the versatility and rigor of our model on the testing data set. This led to an AUC of 0.7048.



We were also interested in seeing if our model would improve if we used elastic net and adjusted the α value. We varied the α value from 0 to 1 in increments of 0.1 and obtained the minimum cross-validation errors for 30 different train-test splits, and then averaged each of these minimum CV errors across all 30 splits for each α value. Interestingly, the minimum CV error occurred when $\alpha = 1$, the LASSO result. We also see low values for CV errors at $\alpha = 0$ (Ridge) and $\alpha = 0.6$ (Elastic Net). Of course, there could be considerable variability in these results, since 30 is not a very large number of iterations (we chose 30 due to time constraints), but what we do notice is that the range of the CV errors is constrained to a tight bound between 0.56 and 0.59 for all α increments, suggesting that LASSO, Ridge, or Elastic Net with $\alpha = 0.6$ (or any other α value, in that case) would produce models with nearly the same minimum cross-validation errors. Therefore, we did not move forward in creating a model using Elastic Net or Ridge.

Random Forest and Classification Trees

Finally, we considered a model in which we regress bankruptcy status *nonlinearly* on all the predictors through fitting a single classification tree and a random forest, subject to a few reasonable threshold levels. The assessment of random forest parameters and the resultant testing ROC curves for both random forest and the single classification tree can be found below.

Assessing Parameters

The relevant paramters for random forest are `mtry` and `ntree`, and the relevant parameters for a single tree (e.g., using `rpart`) are `minsplit` and `cp` (`cp` is a tolerance parameter and does not materially change the prediction power). We manually tested many different combinations of these parameters, and also ran a loop for all reasonable values of `minsplit` and `cp` for single trees, and arrived at the conclusion that the R default values produce the most desirable AUCs, except for the additional specification that `minsplit` should be 15.

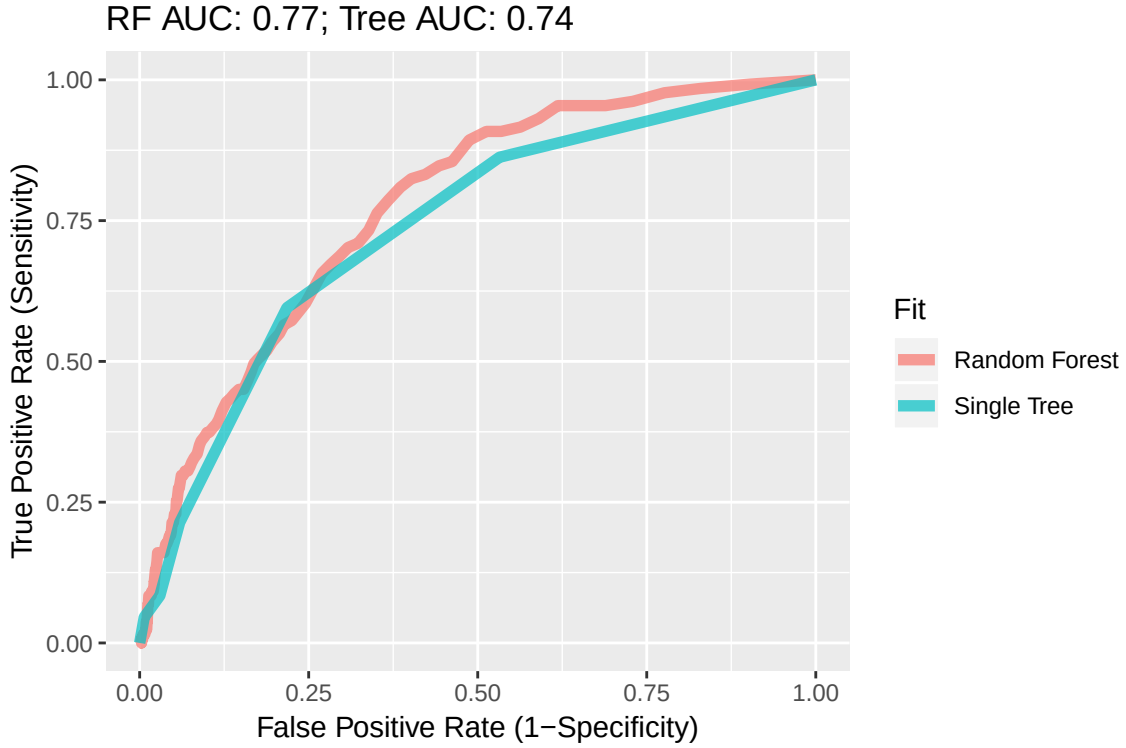
At a high level, random forest is a method of reducing the variance of tree models by introducing bagging, but it results in highly correlated trees. Thus, random forest builds deep random trees for each bootstrap sample by splitting `mtry` randomly-selected features at each splitting point, and then bagging the random

trees by taking averages. If `mtry` is too small, we miss out on important features, but if `mtry` is too large, we get more correlation between the trees. The default value of `mtry` is $\sqrt{p}$ for classification trees.

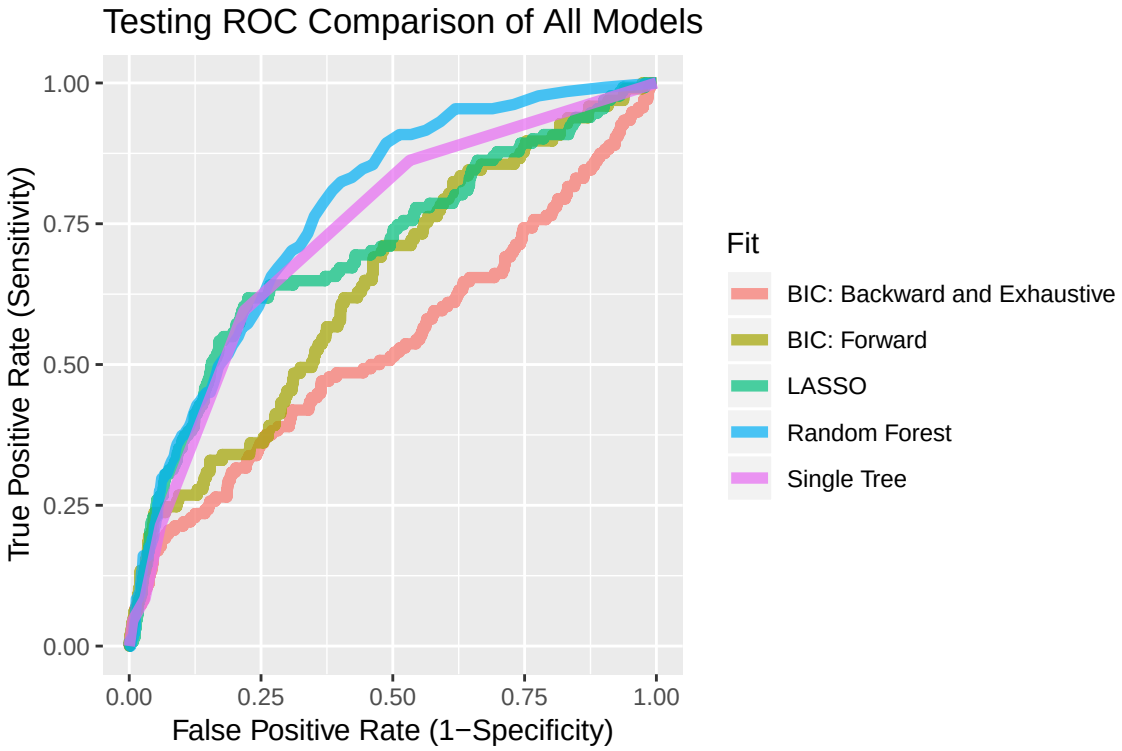
The out-of-bag mean squared errors for different values of `ntree` using random forest plateau very quickly, so more trees do not result in materially better predictive models. We did notice through running a loop that the AUC is maximized at around `ntree` $= 290$, but this is a very marginal maximum, and we think it is due to random variability and not due to the fact that `ntree` *should* be set to 290. Thus, we opt for the default setting of `ntree` $= 500$.

Results

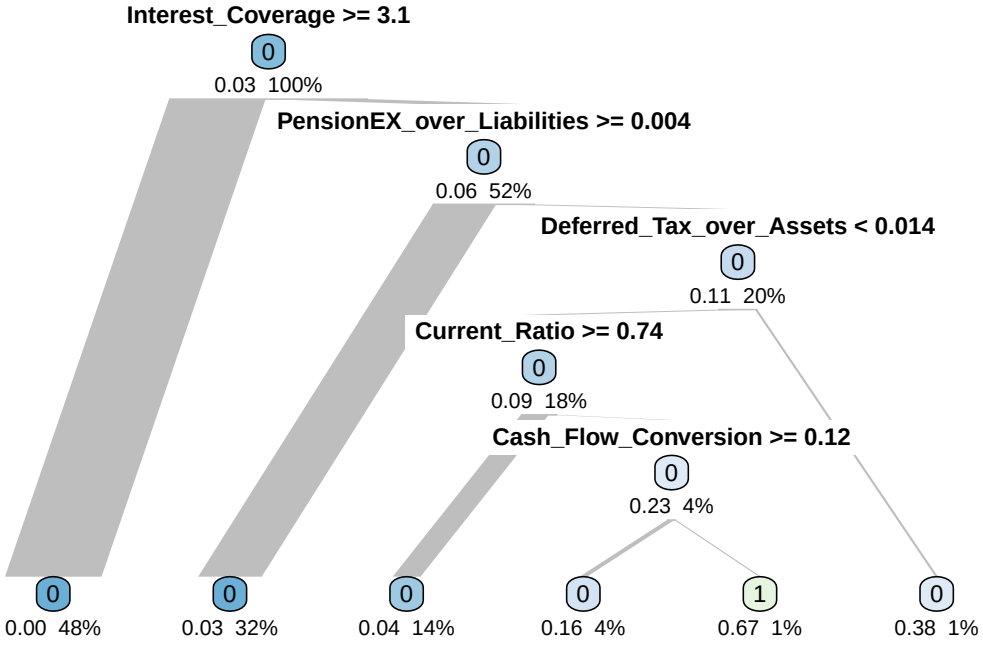
Below, we present the testing ROCs and corresponding AUCs for a random forest (all the default settings of `randomForest`) and a single tree model (using package `rpart`, with `minsplit` $= 15$).



How do these ROCs compare to all of our previous models? We plot them all below. Just by using AUC as a selection criterion, random forest clearly looks to be the best.



One note of consideration is that the single tree fit is still very good, and almost as good as the random forest. We plot the single tree below. Each branch width represents the number of observations in our data set that have that property. We notice that only at the bottom of the tree in one of the subsets are any observations expected to be bankruptcies (with a 66% fitted probability). In all other subsets generated by the tree, we still expect that the company will not go bankrupt.



Conclusion

The Final Model

Since the random forest model maximizes AUC, we will select this prediction model for helping quantamental investors narrow their search for identifying imminent bankruptcies.

Averaged across the random forest and the single tree, the six variables that are considered to be the most important for predicting bankruptcy status (in order of descending importance as determined by permuting OOB data and measuring node impurity via the Gini index) are:

1. Cash Flow Conversion Rate (operating cash flows over EBITDA)
2. Current Ratio (current assets over current liabilities)
3. Pension Expense over Liabilities
4. Interest Coverage Ratio (EBITDA over interest expense)
5. Return on Equity (net income over equity)
6. Return on Assets (EBITDA over assets)

Ultimately, our goal is *prediction*, not interpretability. Therefore, we believe a reasonably complicated model is justified expressly for this purpose. For this reason, we will stick to random forest rather than the single tree.

Classification

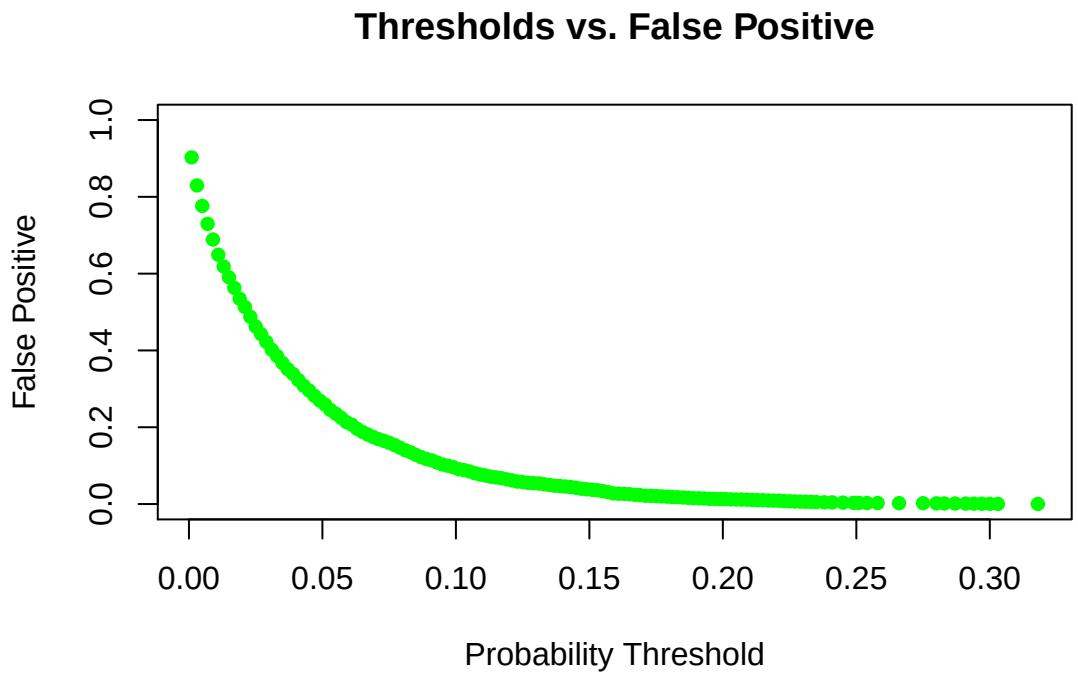


Table 1: Confusion Matrix

	0	1
0	5093	122
1	66	9

As we determined above, the random forest model was the best at predicting whether a company would go bankrupt in the next five years. The Probability Threshold vs. False Positive graph depicts how the false positive rate varies depending on the classification threshold that we choose. Using the elbow rule, we determined a threshold at 0.2 would lead to the best result. This threshold means that we classify a company as going bankrupt in the next five years if its probability is greater than 20%.

Our group felt that we could not justifiably use Bayes' unequal losses to determine a threshold because the cost of misclassifying a company that goes bankrupt as one that is financially strong and the cost of misclassifying a company that is financially strong as one that will go bankrupt depends on the person's objective. For example, if you decide to short the equity of the company that we predict will go bankrupt in the next five years a false positive is extremely costly as the opportunity to profit was missed. Meanwhile, an investment analyst that wants to buy stock in strong companies that will not go bankrupt would be severely punished if our model predicts the company will not go bankrupt and it actually does.

One of the problems that an investment analyst faces is the extremely large universe of companies that are available for investment (remember our 2010 analysis has over 7,000 companies). We believe our bankruptcy prediction model could be a very useful tool to dramatically cut down the investment universe. With a 0.2 classification threshold, our model had a positive prediction rate of 12%. While this might seem low, it's important to realize just how hard of a task it is to predict bankruptcies. If it was easy, than distressed investing would not be considered one of the most inefficient markets on Wall Street. A 12% positive prediction rate is in fact pretty good if you were to rely solely on the model and added no traditional fundamental analysis. Just 2.58% of the companies in our entire data set went bankrupt in 5 years meaning that if we were to randomly guess which companies were to go bankrupt, we would be wrong 97.42% of the time. Therefore, our model predicts bankruptcy over 4.6× better on testing data than randomly guessing.

Another benefit of our model is that it drastically limits the amount of companies that it would classify as going bankrupt. With a 0.2 threshold, it classified 75 companies going bankrupt, nine of which actually did. If these 75 companies were assigned to an investment analyst, they could easily perform additional fundamental analysis and talk to industry experts, customers, suppliers, former employees, and competitors to improve on our 12% positive prediction rate. The analyst would be able to more effectively use his time instead of considering all 5,000+ companies.

Code Appendix

```
# ============================================================================
# ADAPTED version of missing pattern plot by YuSung Su
# ============================================================================
mp.plot <- missing.pattern.plot <- function(data, y.order = FALSE, x.order = FALSE,
    clustered = TRUE, xlab = "Index", ylab = "Variable", main = NULL, gray.scale = FALSE,
    obs.col = "blue", mis.col = "red", ...) {

    if (is.null(main)) {
        main <- deparse(substitute(data))
    }
    index <- seq(nrow(data))
    x.at = 1:nrow(data)
    x.lab = index
    if (y.order) {
        data <- data[, order(colSums(is.na(data)), decreasing = TRUE)]
        ylab = "Ordered by number of missing items"
    }
    if (x.order) {
        data <- data[order(rowSums(is.na(data)), decreasing = FALSE), ]
        index <- row.names(data)
        xlab = "Ordered by number of missing items"
        x.at = NULL
        x.lab = FALSE
    }
    missingIndex <- as.matrix(is.na(data)) * 1
    if (clustered) {
        orderIndex <- order.dendrogram(as.dendrogram(hclust(dist(missingIndex),
            method = "mcquitty")))
        missingIndex <- missingIndex[orderIndex, ]
    }
    col <- if (gray.scale) {
        gray(c(0, 1))
    } else {
        c(obs.col, mis.col)
    }
    # par( mar = c( 4.5, 11, 3, 1 ) ) par( mgp = c( 1, .3, 0 ) ) par( cex.lab =
    # 0.7 )
    empty.labels <- as.vector(matrix(" ", length(names(data))))
    image(x = 1:nrow(data), y = 1:ncol(data), z = missingIndex, ylab = "", xlab = xlab,
        main = main, col = col, yaxt = "n", tck = -0.05, xaxt = "n", ...)
    box("plot")
    axis(side = 2, at = 1:ncol(data), labels = empty.labels, las = 1, tick = FALSE,
        yaxs = "r", tcl = 0.3, xaxs = "i", yaxs = "i")
    mtext(ylab, side = 3, line = 10, cex = 0.7)
    if (x.order) {
        axis(side = 1, at = x.at, labels = x.lab, tick = FALSE, xaxs = "i",
            las = 1)
    }
}
```

```
bankrupt <- read.csv("Bankrupt.csv")
fundamental <- read.csv("CapIQFundamentalData.csv")

fundamental_2010 <- fundamental[fundamental$fyear == 2010, ]
bankrupt_2011through2015 <- bankrupt[substring(bankrupt$BANK_BEGIN_DATE, first = 1,
    last = 4) %in% c("2011", "2012", "2013", "2014", "2015"), ]
fundamental_2010$bankrupt <- fundamental_2010$cik %in% bankrupt_2011through2015$COMPANY_FKEY
fundamental_2010$bankrupt <- as.factor(as.numeric(fundamental_2010$bankrupt))

colnames(fundamental_2010) <- c("Global_Key", "Data_Date", "Fiscal_Year", "Industry_Format",
    "Level_Of_Consolidation", "Population_Score", "Data_Format", "Ticker", "Company_Name",
    "ISO_Currency_Code", "AOCI", "Current_Assets_Total", "Accounts_Payable",
    "Accounts_Receivable", "Assets_Total", "Book_Value_Per_Share", "Capex",
    "Common_Equity_Total", "Cash", "Cash_ST_Investments", "COGS", "Common_Shares_Issued",
    "Debt", "Debt_D1", "Debt_D2", "Debt_D3", "Debt_D4", "Debt_D5", "Long_Term_Debt_Issuance",
    "LT_Debt_Reduction", "Dep_Amort", "Deferred_Revenue_Current", "Deferred_Revenue_LT",
    "Dividends", "Cash_Dividends_Paid", "Dividends_Total", "EBIT", "EBITDA",
    "Employees", "Goodwill", "Gross_Profit", "Intangible_Assets", "Inventory_Total",
    "Current_Liabilities", "Liabilities_Total", "Net_Income", "Notes_Payable",
    "Operating_CF", "PPE_Net", "R_and_D_In_Process_Exp", "Retained_Earnings",
    "Receivables_Total", "Revenue", "Net_Sales", "Stockholders_Equity_Total",
    "Net_Deferred_Tax_A_L", "Working_Capital", "Change_Working_Cap", "Interest_Expense",
    "Staff_Expense", "Operating_Expense", "Pension_Retirement_Exp", "R_And_D_Exp",
    "SG_and_A", "CIK", "Active_Inactive_Status", "Market_Value_Total", "gind",
    "Bankrupt")

fundamental_2010 <- fundamental_2010[!is.na(fundamental_2010$Debt), ]
fundamental_2010[fundamental_2010$Debt == 0, "Debt"] <- NA
fundamental_2010$DebtZero <- is.na(fundamental_2010$Debt)

fundamental_2010 <- fundamental_2010[!is.na(fundamental_2010$Net_Sales), ]
fundamental_2010[fundamental_2010$Net_Sales == 0, "Net_Sales"] <- NA
fundamental_2010$NetSalesZero <- is.na(fundamental_2010$Net_Sales)

# calculating ratios
fundamental_2010$WC_over_Assets <- fundamental_2010$Working_Capital/fundamental_2010$Assets_Total
fundamental_2010$RE_over_Assets <- fundamental_2010$Retained_Earnings/fundamental_2010$Assets_Total
fundamental_2010$ROA <- fundamental_2010$EBIT/fundamental_2010$Assets_Total
fundamental_2010$mvEquity_over_Liabilities <- fundamental_2010$Market_Value_Total/fundamental_2010$Liabi
fundamental_2010$Asset_Turnover <- fundamental_2010$Revenue/fundamental_2010$Assets_Total
fundamental_2010$Debt_to_Equity <- fundamental_2010$Debt/fundamental_2010$Stockholders_Equity_Total
fundamental_2010$Interest_Coverage <- fundamental_2010$EBIT/fundamental_2010$Interest_Expense
fundamental_2010$Leverage_Ratio <- fundamental_2010$Debt/fundamental_2010$EBITDA
fundamental_2010$Cash_Flow_Conversion <- fundamental_2010$Operating_CF/fundamental_2010$EBITDA
fundamental_2010$Inventory_Turnover <- fundamental_2010$COGS/fundamental_2010$Inventory_Total
fundamental_2010$DSO <- fundamental_2010$Receivables_Total/fundamental_2010$Net_Sales *
    365
fundamental_2010$DPO <- (fundamental_2010$Accounts_Payable/fundamental_2010$COGS) *
    365
fundamental_2010$ROE <- fundamental_2010$Net_Income/fundamental_2010$Stockholders_Equity_Total
fundamental_2010$Current_Ratio <- fundamental_2010$Current_Assets_Total/fundamental_2010$Current_Liabili
fundamental_2010$Quick_Ratio <- (fundamental_2010$Cash_ST_Investments + fundamental_2010$Receivables_To
fundamental_2010$OCF_over_Sales <- fundamental_2010$Operating_CF/fundamental_2010$Net_Sales
```

```
fundamental_2010$stDebt_over_Debt <- fundamental_2010$Debt_D1/fundamental_2010$Debt
fundamental_2010$EBITDA_Margin <- fundamental_2010$EBITDA/fundamental_2010$Revenue
fundamental_2010$Gross_Margin <- fundamental_2010$Gross_Profit/fundamental_2010$Revenue
fundamental_2010$CAPEX_over_Sales <- fundamental_2010$Capex/fundamental_2010$Revenue
fundamental_2010$CAPEX_over_Assets <- fundamental_2010$Capex/fundamental_2010$Assets_Total
fundamental_2010$CAPEX_over_PPE <- fundamental_2010$Capex/fundamental_2010$PPE_Net
fundamental_2010$CAPEX_over_Dep <- fundamental_2010$Capex/fundamental_2010$Dep_Amort
fundamental_2010$RandD_over_Sales <- fundamental_2010$R_And_D_Exp/fundamental_2010$Net_Sales
fundamental_2010$OPEX_over_Sales <- fundamental_2010$Operating_Expense/fundamental_2010$Revenue
fundamental_2010$PensionEX_over_Sales <- fundamental_2010$Pension_Retirement_Exp/fundamental_2010$Revenu
fundamental_2010$PensionEX_over_Liabilities <- fundamental_2010$Pension_Retirement_Exp/fundamental_2010$
fundamental_2010$Interest_over_Debt <- fundamental_2010$Interest_Expense/fundamental_2010$Debt
fundamental_2010$Deferred_Tax_over_Assets <- fundamental_2010$Net_Deferred_Tax_A_L/fundamental_2010$Asse
fundamental_2010$Goodwil_over_Assets <- fundamental_2010$Goodwill/fundamental_2010$Assets_Total
fundamental_2010$SGA_over_Employee <- fundamental_2010$SG_and_A/fundamental_2010$Employees
fundamental_2010$Dividend_Payout_Ratio <- fundamental_2010$Dividends_Total/fundamental_2010$Net_Income
fundamental_2010$Deferred_Revenue_over_Revenue <- fundamental_2010$Deferred_Revenue_LT/fundamental_2010$
fundamental_2010$Deferred_Revenue_over_Assets <- fundamental_2010$Deferred_Revenue_LT/fundamental_2010$A
fundamental_2010$Debt_Paydown_over_Liabilities <- fundamental_2010$LT_Debt_Reduction/fundamental_2010$Li
fundamental_2010$Debt_Paydown_over_Debt <- fundamental_2010$LT_Debt_Reduction/fundamental_2010$Debt
fundamental_2010$AOCI_over_Net_Income <- fundamental_2010$AOCI/fundamental_2010$Net_Income
fundamental_2010$Book_to_Market <- fundamental_2010$Book_Value_Per_Share/(fundamental_2010$Market_Value_

data_labels <- fundamental_2010
data_labels <- data_labels[, -c(1:64, 67, 68)]  # with CIK labels
```

```
data <- data_labels[, -1]  # without CIK labels

summary(data)

inf2NA <- function(x) {
    # This function is inspired from 'ricardo' on Stack Exchange:
    # https://stackoverflow.com/questions/12188509/cleaning-inf-values-from-an-r-dataframe
    for (i in 1:ncol(x)) {
        x[, i][is.infinite(x[, i])] <- NA
    }
    return(x)
}

data <- inf2NA(data)

data <- data[, -26]  # all NA/NANs
data <- data[, -c(10, 12, 21, 27, 31, 39)]  # lots of NAs
data <- data[, -1]  # Throws off analysis

summary(data)

data <- data[, -c(16, 19, 21, 23)]  # highly correlated

data <- data[, -3]  # Omit the NetSalesZero variable, since including it in a logit model creates singu


# The most naive model: out <- glm(Bankrupt ~. , data, family='binomial',
# na.action = na.exclude) summary(out)
```

```
par(mfrow = c(1, 2))
mp.plot(fundamental_2010, main = "Raw Data", xlab = "", ylab = "", obs.col = "darkgray",
    mis.col = "white")
mp.plot(data, main = "Before Imputation", xlab = "", ylab = "", obs.col = "darkgray",
    mis.col = "white")
```

```
data_num <- select_if(data, is.numeric)
cormat <- cor(data_num, use = "pairwise.complete.obs")
cormat <- abs(cormat)
melted_cormat <- melt(cormat)
colnames(melted_cormat)[3] <- "value"
ggplot(data = melted_cormat, aes(x = Var1, y = Var2, fill = value)) + geom_tile() +
    xlab("") + ylab("") + ggtitle("Correlation Heatmap") + theme(title = element_text(size = 10),
    axis.text.x = element_text(size = 6, angle = 45, hjust = 1), axis.text.y = element_text(size = 6),
    plot.title = element_text(hjust = 0.5))
```

```
data_imp <- data
data_imp$Bankrupt <- as.numeric(data_imp$Bankrupt) - 1
data_imp$DebtZero <- as.numeric(data_imp$DebtZero)

set.seed(1)
imputation <- mice(data_imp)

###

imputed_data <- imputation$imp
filled_data <- data_imp

for (i in 3:length(imputed_data)) {
    imputed_NAs <- as.vector(rowSums(imputed_data[[i]]))/ncol(imputed_data[[i]])
    for (j in 1:nrow(filled_data)) {
        if (is.na(filled_data[j, i])) {
            filled_data[j, i] <- imputed_NAs[1]
            imputed_NAs <- imputed_NAs[-1]
        }
    }
}

summary(filled_data)
filled_data$Bankrupt <- as.factor(filled_data$Bankrupt)
filled_data$DebtZero <- as.factor(filled_data$DebtZero)
summary(filled_data)

filled_data_temp <- filled_data  # for backups
filled_data <- filled_data_temp  # for backups
```

```
# outliers
# source('https://raw.githubusercontent.com/talgalili/R-code-snippets/master/boxplot.with.outlier.label
# boxplot.with.outlier.label(filled_data[,3], 1:nrow(filled_data),
# spread_text = F)

outlier_rows <- c(4081)
filled_data <- filled_data[-outlier_rows, ]
```

```
# boxplot.with.outlier.label(filled_data[,3], 1:nrow(filled_data),
# spread_text = F)

outlier_rows <- c(outlier_rows, 1237)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,3], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,4], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 2314, 5249, 1510)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,4], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 4042, 2959)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,4], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,5], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6310, 6353, 6582)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,5], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,6], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6591)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,6], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 1614, 6545)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,6], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,7], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 3786, 7074)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,7], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 5565)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,7], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 574)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,7], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,8], 1:nrow(filled_data),
```

```
# spread_text = F) reasonable

# boxplot.with.outlier.label(filled_data[,9], 1:nrow(filled_data),
# spread_text = F) reasonable

# boxplot.with.outlier.label(filled_data[,10], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 2708, 6570, 5260)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,10], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,11], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6253, 6309, 6325, 6810, 6623)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,11], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,12], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 1776)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,12], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,13], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 3853, 1405, 2998, 762, 1025)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,13], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,14], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 3007)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,14], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 4177, 1660)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,14], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,15], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 4859, 3393)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,15], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6676, 2303, 3626)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,15], 1:nrow(filled_data),
```

```
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,16], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 3723)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,16], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 4525, 4971, 5378, 1727)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,16], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,17], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 2200, 5800, 6305)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,17], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,18], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6488, 4629, 3534)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,18], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,19], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 1549, 3081)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,19], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,20], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 260)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,20], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,21], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 5459, 5556)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,21], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,22], 1:nrow(filled_data),
# spread_text = F) reasonable

# boxplot.with.outlier.label(filled_data[,23], 1:nrow(filled_data),
# spread_text = F)
```

```
outlier_rows <- c(outlier_rows, 4966, 1219, 4843)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,23], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,24], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 6322, 4484)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,24], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 4194, 4258)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,24], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,25], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 30)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,25], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 5315)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,25], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,26], 1:nrow(filled_data),
# spread_text = F) reasonable

# boxplot.with.outlier.label(filled_data[,27], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 5575, 4099)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,27], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 183)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,27], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,28], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 2104, 1350, 3394, 202)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,28], 1:nrow(filled_data),
# spread_text = F)

# boxplot.with.outlier.label(filled_data[,29], 1:nrow(filled_data),
# spread_text = F)
outlier_rows <- c(outlier_rows, 3306, 2760, 991)
filled_data <- filled_data[-outlier_rows, ]
# boxplot.with.outlier.label(filled_data[,29], 1:nrow(filled_data),
```

```
# spread_text = F)

summary(filled_data)
filled_data_no_outliers <- filled_data
# filled_data <- filled_data_temp

data <- filled_data_no_outliers
```

```
par(mfrow = c(1, 3))
plot(jitter(as.numeric(data$Bankrupt) - 1, factor = 0.5) ~ data$Asset_Turnover,
    pch = 4, col = data$Bankrupt, ylab = "Bankruptcy Status", xlab = "Sales / Total Assets",
    xlim = c(0, 20), labels = F)
# legend('topright', legend=c('Not bankrupt', 'Bankrupt'), lty=c(1,1),
# lwd=c(2,2), col=unique(data$Bankrupt))
axis(at = c(0, 1), labels = c("Not Bankrupt", "Bankrupt"), side = 2)
axis(at = c(0, 5, 10, 15, 20), side = 1)
plot(jitter(as.numeric(data$Bankrupt) - 1, factor = 0.5) ~ data$Goodwil_over_Assets,
    pch = 4, col = data$Bankrupt, ylab = "Bankruptcy Status", xlab = "Goodwill / Assets",
    xlim = c(0, 1), labels = F)
# legend('topright', legend=c('Not bankrupt', 'Bankrupt'), lty=c(1,1),
# lwd=c(2,2), col=unique(data$Bankrupt))
axis(at = c(0, 1), labels = c("Not Bankrupt", "Bankrupt"), side = 2)
axis(at = c(0, 0.2, 0.4, 0.6, 0.8, 1), side = 1)
plot(jitter(as.numeric(data$Bankrupt) - 1, factor = 0.5) ~ data$PensionEX_over_Liabilities,
    pch = 4, col = data$Bankrupt, ylab = "Bankruptcy Status", xlab = "Pension Expense / Liabilities",
    xlim = c(0, 0.4), labels = F)
# legend('topright', legend=c('Not bankrupt', 'Bankrupt'), lty=c(1,1),
# lwd=c(2,2), col=unique(data$Bankrupt)) axis(2, labels = FALSE)
axis(at = c(0, 1), labels = c("Not Bankrupt", "Bankrupt"), side = 2)
axis(at = c(0, 0.1, 0.2, 0.3, 0.4), side = 1)
```

```
p1 <- ggplot(data, aes(x = mvEquity_over_Liabilities)) + geom_histogram(aes(y = ..density..),
    binwidth = 5, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust =
    ggtitle("Equity Over liabilities") + xlab(element_blank()) + xlim(c(-25,
    200))

p2 <- ggplot(data, aes(x = Current_Ratio)) + geom_histogram(aes(y = ..density..),
    binwidth = 1, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust =
    ggtitle("Current Ratio") + xlab(element_blank()) + xlim(c(0, 200))

p3 <- ggplot(data, aes(x = RE_over_Assets)) + geom_histogram(aes(y = ..density..),
    binwidth = 1, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust =
    ggtitle("Retained Earnings/ Net Income") + xlab(element_blank()) + xlim(c(-50,
    5))

p4 <- ggplot(data, aes(x = EBITDA_Margin)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.2, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
```

```
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust = 
    ggtitle("EBITDA Margin") + xlab(element_blank()) + xlim(c(-20, 5))

p5 <- ggplot(data, aes(x = Asset_Turnover)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.15, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust = 
    ggtitle("Asset Turnover") + xlab(element_blank()) + xlim(c(0, 10))

p6 <- ggplot(data, aes(x = WC_over_Assets)) + geom_histogram(aes(y = ..density..),
    binwidth = 1, colour = "black", fill = "white") + geom_density(alpha = 0.2,
    fill = "#66F6FF") + theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
    axis.ticks.y = element_blank(), title = element_text(size = 10), plot.title = element_text(hjust = 
    ggtitle("Working Capital over Assets") + xlab(element_blank()) + xlim(c(-25,
    5))


multiplot(p1, p2, p3, p4, p5, p6, cols = 2)
```

```
data1 <- data_imp
set.seed(1)
fit.backward <- regsubsets(Bankrupt ~ ., data1, nvmax = ncol(data1), method = "backward")
f.b <- summary(fit.backward)

fit.forward <- regsubsets(Bankrupt ~ ., data1, nvmax = ncol(data1), method = "forward")
f.f <- summary(fit.forward)

fit.exhaustive <- regsubsets(Bankrupt ~ ., data1, nvmax = ncol(data1), method = "exhaustive")
f.e <- summary(fit.exhaustive)

par(mar = c(1, 1, 1, 1))
par(mfrow = c(3, 1), mar = c(2.5, 4, 0.5, 1), mgp = c(1.5, 0.5, 0))
# BIC, backward
min_f.b.bic <- which.min(f.b$bic)
plot(f.b$bic, xlab = "Number of predictors", ylab = "BIC (backward)", col = "gray",
    type = "b", pch = 10)
points(min_f.b.bic, f.b$bic[min_f.b.bic], col = "blue", cex = 3, pch = 16)
# BIC, forward
min_f.f.bic <- which.min(f.f$bic)
plot(f.f$bic, xlab = "Number of predictors", ylab = "BIC (forward)", col = "gray",
    type = "b", pch = 10)
points(min_f.f.bic, f.f$bic[min_f.f.bic], col = "blue", cex = 3, pch = 16)
# BIC, exhaustive
min_f.e.bic <- which.min(f.e$bic)
plot(f.e$bic, xlab = "Number of predictors", ylab = "BIC (exhaustive)", col = "gray",
    type = "b", pch = 10)
points(min_f.e.bic, f.e$bic[min_f.e.bic], col = "blue", cex = 3, pch = 16)
```

```
min_f.b.bic
rownames(data.frame(f.b$outmat[min_f.b.bic, ]))[which(data.frame(f.b$outmat[min_f.b.bic,
    ])[, 1] == "*")]

min_f.f.bic
rownames(data.frame(f.f$outmat[min_f.f.bic, ]))[which(data.frame(f.f$outmat[min_f.f.bic,
```

```
    ])[, 1] == "*")]

min_f.e.bic
rownames(data.frame(f.b$outmat[min_f.e.bic, ]))[which(data.frame(f.b$outmat[min_f.e.bic,
    ])[, 1] == "*")]
```

```
var_b.bic <- c("WC_over_Assets", "DPO", "ROE", "CAPEX_over_Sales", "Goodwil_over_Assets")
data_b.bic <- cbind(Bankrupt = data1$Bankrupt, data1[, var_b.bic])
glm_formula_b.bic <- as.formula(paste("Bankrupt", "~", paste(paste(var_b.bic,
    sep = ""), collapse = "+")))

var_f.bic <- c("PensionEX_over_Liabilities", "ROE", "CAPEX_over_Sales", "Goodwil_over_Assets")
data_f.bic <- cbind(Bankrupt = data1$Bankrupt, data1[, var_f.bic])
glm_formula_f.bic <- as.formula(paste("Bankrupt", "~", paste(paste(var_f.bic,
    sep = ""), collapse = "+")))

set.seed(1)
index.train <- sample(nrow(data), 1000)

data.train.b <- data_b.bic[index.train, ]
data.test.b <- data_b.bic[-index.train, ]

data.train.f <- data_f.bic[index.train, ]
data.test.f <- data_f.bic[-index.train, ]

fit.train.b <- glm(glm_formula_b.bic, data.train.b, family = "binomial")
fit.fitted.test.b <- predict(fit.train.b, data.test.b, type = "response")

fit.train.f <- glm(glm_formula_f.bic, data.train.f, family = "binomial")
fit.fitted.test.f <- predict(fit.train.f, data.test.f, type = "response")


# par(mfrow=c(1,2))
fit.test.roc.b <- roc(data.test.b$Bankrupt, fit.fitted.test.b, plot = F)
auc.b <- auc(fit.test.roc.b)
fit.test.roc.b <- roc(data.test.b$Bankrupt, fit.fitted.test.b, plot = F, main = paste("Backward AUC: ",
    round(auc.b, 4), sep = ""))

fit.test.roc.f <- roc(data.test.f$Bankrupt, fit.fitted.test.f, plot = F)
auc.f <- auc(fit.test.roc.f)
fit.test.roc.f <- roc(data.test.f$Bankrupt, fit.fitted.test.f, plot = F, main = paste("Forward AUC: ",
    round(auc.f, 4), sep = ""))
```

```
b_roc_ss <- data.frame(TPR = fit.test.roc.b$sensitivities, FPR = (1 - fit.test.roc.b$specificities),
    Fit = "Backward and Exhaustive")
f_roc_ss <- data.frame(TPR = fit.test.roc.f$sensitivities, FPR = (1 - fit.test.roc.f$specificities),
    Fit = "Forward")
roc_ss_BIC <- rbind(b_roc_ss, f_roc_ss)
ggplot(roc_ss_BIC, aes(FPR, TPR, color = Fit)) + geom_line(size = 2, alpha = 0.7) +
    labs(title = paste("Backward and Exhaustive AUC: ", round(fit.test.roc.b$auc,
        2), "; Forward AUC: ", round(fit.test.roc.f$auc, 2), sep = ""), x = "False Positive Rate (1-Spe
        y = "True Positive Rate (Sensitivity)")
```

```
X <- model.matrix(Bankrupt ~ ., data)[, -1]  #This is omitting the NAs
Y <- data$Bankrupt

set.seed(1)

fit.LASSO.cv <- cv.glmnet(X, Y, alpha = 1, family = "binomial", nfolds = 10,
    type.measure = "auc")

plot(fit.LASSO.cv)

# Using Lambda.min
coef.min <- coef(fit.LASSO.cv, s = "lambda.min")
coef.min <- coef.min[which(coef.min != 0), ]
# coef.min
LASSO_Variables <- rownames(as.matrix(coef.min))
```

```
fit1min.LASSO.Logit <- glm(Bankrupt ~ DebtZero + mvEquity_over_Liabilities +
    Asset_Turnover + DSO + ROE + Current_Ratio + CAPEX_over_Dep + PensionEX_over_Liabilities +
    Goodwil_over_Assets + Deferred_Revenue_over_Assets + Book_to_Market, data = data,
    family = binomial)

summary(fit1min.LASSO.Logit)
Anova(fit1min.LASSO.Logit)

fit2min.LASSO.Logit <- update(fit1min.LASSO.Logit, . ~ . - Asset_Turnover)
Anova(fit2min.LASSO.Logit)

fit3min.LASSO.Logit <- update(fit2min.LASSO.Logit, . ~ . - DSO)
Anova(fit3min.LASSO.Logit)

fit4min.LASSO.Logit <- update(fit3min.LASSO.Logit, . ~ . - Deferred_Revenue_over_Assets)
Anova(fit4min.LASSO.Logit)

fit5min.LASSO.Logit <- update(fit4min.LASSO.Logit, . ~ . - mvEquity_over_Liabilities)
Anova(fit5min.LASSO.Logit)

fit6min.LASSO.Logit <- update(fit5min.LASSO.Logit, . ~ . - ROE)
Anova(fit6min.LASSO.Logit)

fit7min.LASSO.Logit <- update(fit6min.LASSO.Logit, . ~ . - Book_to_Market)
Anova(fit7min.LASSO.Logit)
summary(fit7min.LASSO.Logit)  #Final Model: DebtZero, Current_Ratio, CAPEX_over_Dep PensionEX_over_Liab

# judging the fits using AUC fit7min.LASSO.roc <-
# roc(data$Bankrupt,fit7min.LASSO.Logit$fitted, plot = T, col = 'blue')


# Training Indices
set.seed(1)
index.train <- sample(nrow(data), 1000)

# Train-Test Split
data.train <- data[index.train, ]
data.test <- data[-index.train, ]
```

```
# Refitting LASSO fit #5 for training/testing
fit7min.LASSO.Logit.train <- glm(Bankrupt ~ DebtZero + Current_Ratio + CAPEX_over_Dep +
    PensionEX_over_Liabilities + Goodwil_over_Assets, data = data.train, family = binomial)
predict.LASSO <- predict.glm(fit7min.LASSO.Logit.train, newdata = data.test,
    type = "response")
fit7min.LASSO.testing.roc <- roc(data.test$Bankrupt, as.vector(predict.LASSO),
    plot = FALSE)

fit7min.LASSO.testing.roc$auc  #auc = .7048


# par(mfrow=c(1,2))
```

```
# plot(fit7min.LASSO.testing.roc, main=paste('LASSO Testing
# AUC:',round(fit7min.LASSO.testing.roc$auc,3)))
roc_ss_LASSO <- data.frame(TPR = fit7min.LASSO.testing.roc$sensitivities, FPR = (1 -
    fit7min.LASSO.testing.roc$specificities))
ggplot(roc_ss_LASSO, aes(FPR, TPR)) + geom_line(size = 2, alpha = 0.7) + labs(title = paste("LASSO Testi
    round(fit7min.LASSO.testing.roc$auc, 2)), x = "False Positive Rate (1-Specificity)",
    y = "True Positive Rate (Sensitivity)")
```

```
# Elastic Net Model

Alist <- seq(0, 1, by = 0.1)

################################################
set.seed(1)
Alist_vals_mat <- matrix(0, 30, length(Alist))
for (j in 1:30) {
    elasticnet <- lapply(Alist, function(a) {
        cv.glmnet(X, Y, alpha = a, family = "binomial", nfolds = 10, type.measure = "auc")
    })
    Alist_vals <- as.vector(matrix(0, length(Alist)))
    for (i in 1:11) {
        Alist_vals[i] <- min(elasticnet[[i]]$cvm)
    }  #minimum CV error when alpha=1 (LASSO result)
    Alist_vals_mat[j, ] <- Alist_vals
}
write.csv(Alist_vals_mat, "Alist_vals_mat.csv")
######### The above code takes very long to run, so I have saved the data frame to a
######### file and upload it for quick knitting: Alist_vals_mat <-
######### read.csv('Alist_vals_mat.csv') Alist_vals_mat <- as.matrix(Alist_vals_mat)
######### Alist_vals_mat <- Alist_vals_mat[,-1] colnames(Alist_vals_mat) <- NULL
colnames(Alist_vals_mat) <- Alist
Alist_vals_agg <- as.vector(matrix(0, length(Alist)))
for (i in 1:length(Alist_vals_agg)) {
    Alist_vals_agg[i] <- mean(Alist_vals_mat[, i])
}

Adf <- data.frame(alpha = Alist, minCVE = Alist_vals_agg)
Adf$alpha <- as.factor(Adf$alpha)

ggplot(data = Adf, aes(x = alpha, y = minCVE)) + geom_bar(stat = "identity") +
```

```
    ggtitle("Elastic Net CV Errors") + ylab("Minimum Cross-Validation Error") +
    xlab("Alpha") + coord_cartesian(ylim = c(0.56, 0.59))

# X <- model.matrix(Bankrupt~.,data.train)[,-1] #This is omitting the NAs Y
# <- data.train$Bankrupt fit.EN <- glmnet(X,Y, alpha = .8, family =
# 'binomial', nfolds = 10, type.measure = 'auc') predict.EN <-
# predict.glm(fit.EN, newdata=data.test, type='response') roc.EN <-
# roc(data.test$Bankrupt, as.vector(predict.EN), plot=F) roc.EN$auc
```

```
# Training Indices
set.seed(1)
index.train <- sample(nrow(data), 1000)

# Train-Test Split
data.train.tree <- data[index.train, ]
data.test.tree <- data[-index.train, ]

# Random Forest
set.seed(1)
fit.rf.train <- randomForest(Bankrupt ~ ., data.train.tree)
predict.rf <- predict(fit.rf.train, newdata = data.test.tree, type = "prob")

rf_roc <- roc(data.test.tree$Bankrupt, predict.rf[, 2], plot = FALSE)

# Tree
fit.tree.train <- rpart(Bankrupt ~ ., data.train.tree, minsplit = 15)
predict.tree <- predict(fit.tree.train, newdata = data.test.tree, type = "prob")
tree_roc <- roc(data.test.tree$Bankrupt, predict.tree[, 2], plot = FALSE)

# par(mfrow=c(1,2)) plot(rf_roc, main=paste('Random Forest Testing
# AUC:',round(rf_roc$auc,3))) plot(tree_roc, main=paste('Tree Testing
# AUC:',round(tree_roc$auc,3)))

rf_roc_ss <- data.frame(TPR = rf_roc$sensitivities, FPR = (1 - rf_roc$specificities),
    Fit = "Random Forest")
tree_roc_ss <- data.frame(TPR = tree_roc$sensitivities, FPR = (1 - tree_roc$specificities),
    Fit = "Single Tree")
roc_ss <- rbind(rf_roc_ss, tree_roc_ss)
ggplot(roc_ss, aes(FPR, TPR, color = Fit)) + geom_line(size = 2, alpha = 0.7) +
    labs(title = paste("RF AUC: ", round(rf_roc$auc, 2), "; Tree AUC: ", round(tree_roc$auc,
        2), sep = ""), x = "False Positive Rate (1-Specificity)", y = "True Positive Rate (Sensitivity)
```

```
b_roc_ss$Fit <- "BIC: Backward and Exhaustive"
f_roc_ss$Fit <- "BIC: Forward"
roc_ss_BIC <- rbind(b_roc_ss, f_roc_ss)
roc_ss_LASSO$Fit <- "LASSO"
roc_ss <- rbind(roc_ss_BIC, roc_ss_LASSO, roc_ss)
ggplot(roc_ss, aes(FPR, TPR, color = Fit)) + geom_line(size = 2, alpha = 0.7) +
    labs(title = "Testing ROC Comparison of All Models", x = "False Positive Rate (1-Specificity)",
        y = "True Positive Rate (Sensitivity)")
```

```
# fancyRpartPlot(fit.tree.train) prp(fit.tree.train, box.palette = 'RdYlGn')
# prp(fit.tree.train, extra = 6, box.palette = 'auto')
rpart.plot(fit.tree.train, type = 1, clip.right.labs = FALSE, branch = 0.3,
```

```
    under = TRUE, branch.type = 5, yesno = FALSE, faclen = 0)
```

```
rf_importance <- as.numeric(fit.rf.train$importance)
names(rf_importance) <- rownames(fit.rf.train$importance)

# barplot(rev(sort(rf_importance)))
# barplot(rev(sort(fit.tree.train$variable.importance)))
# data.frame(RF=names(head(rev(sort(rf_importance)),5)),
# Tree=names(head(rev(sort(fit.tree.train$variable.importance)),5)))

rf_importance <- rev(sort(rf_importance))
tree_importance <- rev(sort(fit.tree.train$variable.importance))
rf_importance_standardized <- (rf_importance - mean(rf_importance))/sd(rf_importance)
tree_importance_standardized <- (tree_importance - mean(tree_importance))/sd(tree_importance)

rf_importance_df <- data.frame(names(rf_importance_standardized), as.vector(rf_importance_standardized))
names(rf_importance_df) <- c("name", "importance")
rf_importance_df <- rf_importance_df[order(rf_importance_df$name), ]

tree_importance_df <- data.frame(names(tree_importance_standardized), as.vector(tree_importance_standard
names(tree_importance_df) <- c("name", "importance")
tree_importance_df <- tree_importance_df[order(tree_importance_df$name), ]

tree_importance_df <- tree_importance_df[(tree_importance_df$name %in% rf_importance_df$name),
    ]
rf_importance_df <- rf_importance_df[(rf_importance_df$name %in% tree_importance_df$name),
    ]

importance_df <- cbind(tree_importance_df[, 2], rf_importance_df[, 2])
rownames(importance_df) <- tree_importance_df$name
colnames(importance_df) <- c("tree", "rf")
importance_df <- data.frame(importance_df)
importance_df$sum <- importance_df$tree + importance_df$rf
importance_df <- importance_df[order(importance_df$sum), ]
importance_df <- importance_df[(importance_df$sum > summary(importance_df$sum)[[4]]),
    ]

head(rev(rownames(importance_df)), 6)
```

```
plot(rf_roc$thresholds, 1 - rf_roc$specificities, col = "green", pch = 16, xlab = "Probability Threshol
    ylab = "False Positive", main = "Threshold vs. False Postive")
```

```
# .1 using the elbow rule
fit.rf.pred.1 <- ifelse(predict.rf > 0.2, "1", "0")
cm.1 <- table(fit.rf.pred.1[, 2], data.test$Bankrupt)
cm.1

positive.pred <- cm.1[2, 2]/(cm.1[2, 1] + cm.1[2, 2])
positive.pred
```

```
kable(cm.1, align = "c", caption = "Confusion Matrix")
```