Code

Predicting Bankruptcy

Evan Gao

2024-06-15

Introduction

Overview

Bankruptcy prediction is a critical task for financial institutions, investors, and regulators to assess the financial health and stability of companies. By leveraging historical financial data and other relevant features, machine learning models can be developed to predict the likelihood of a company going bankrupt within a certain time frame. This project aims to explore bankruptcy prediction using a dataset of companies with known bankruptcy outcomes and various financial attributes.

Objective

The primary objective of this project is to build a predictive model that can accurately predict the likelihood of bankruptcy for companies based on their financial characteristics. By analyzing historical financial data and other relevant features, the model will provide insights into the key factors that contribute to bankruptcy risk. The model's performance will be evaluated using appropriate metrics to assess its predictive power and generalizability.

The predictive model developed in this project will help stakeholders, including financial institutions, investors, and regulators, make informed decisions regarding risk assessment, investment strategies, and regulatory compliance. By identifying companies at high risk of bankruptcy, stakeholders can take proactive measures to mitigate financial losses and minimize risks.

Dataset Description

The dataset used in this project contains information on companies with known bankruptcy outcomes and various financial attributes. The dataset includes features such as liquidity ratios, profitability ratios, leverage ratios, and other financial indicators that are commonly used in bankruptcy prediction models. The dataset is sourced from Kaggle and provides a comprehensive set of features for building predictive models.

Definition of Key Terms

- **Bankruptcy Prediction**: The task of predicting the likelihood of a company going bankrupt within a certain time frame based on historical financial data and other relevant features.
- **Financial Ratios**: Quantitative indicators calculated from a company's financial statements that provide insights into its financial health and performance. (Further expansion on specific ratios are in the preliminary feature selection section)

Data Description and Preprocessing

Source

The dataset used in this project is sourced from Kaggle (https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction/data) which is obtained from UCI Machine Learning Repository. The dataset originates from the Taiwan Economic Journal for the years 1999 to 2009. It contains financial ratios and other attributes of companies, along with their bankruptcy status within a certain time frame. The dataset will be preprocessed and cleaned to ensure data quality and consistency for building predictive models.

Loading the Data

We will first load the data and evaluate the raw dataset to understand its structure and features.

As seen in the table below, the dataset contains 96 features including our response variable 'bankrupt' which indicates whether a company went bankrupt or not. Of these 96 features, we have 'bankrupt,' 'liability_assets_flag,' and 'net_income_flag' as indicators when the other variable are numerical ratios. As we move on with our analysis, we would need to transform and select our features accordingly to build a predictive model.

Show

Search:

	bankrupt	roa_c_before_interest_and_depreciation_before_interest	roa_a_b
1	1.000	0.371	
2	1.000	0.464	
3	1.000	0.426	
4	1.000	0.400	
5	1.000	0.465	
6	1.000	0.389	

Showing 1 to 6 of 6 entries Previous 1 Next

We will also need to factor our indicator variables to categorical variables for our analysis.

Show

Next, we will check for missing values in the dataset and handle them appropriately.

Show

```
## [1] 0
```

As we see, there are no missing values within our dataset. This is a good sign as we can proceed with our analysis without the need for imputation.

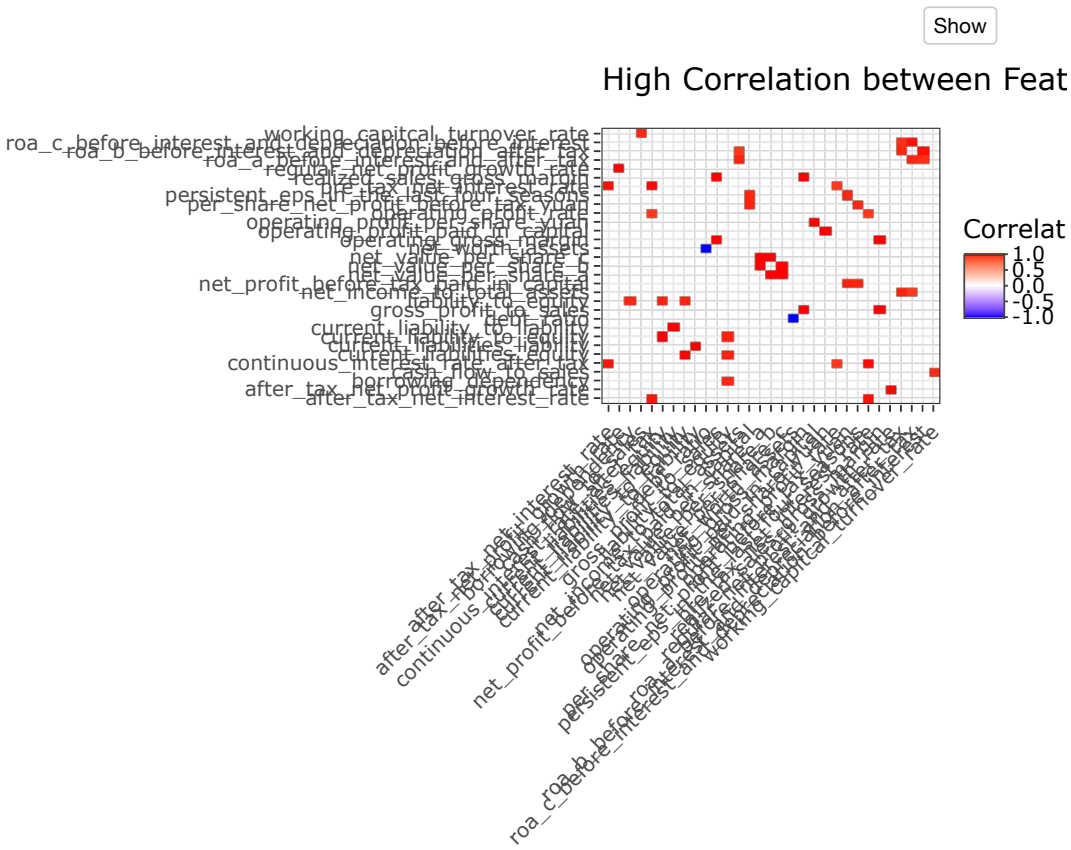
Exploratory Data Analysis

Preliminary Feature Selection

Due to the large number of features in the dataset, we will focus on exploring the correlation between each variable to help us in our feature selection process. To achieve this, we will emphasize the variables with high degrees of correlation (>0.9 and <-0.9). This will serve as our basis for feature selection in our predictive model by reducing certain highly correlated variables.

As seen in the heatmap below, there are several features that are highly correlated with each other. This justifies our combination of some of these highly correlated variables for example, we could select the debt_ratio variable and exclude the net_worth_assets as they have a correlation of -1.

Show



With the combination of the analysis from our correlation plot along with some domain knowledge, we can narrow the features that will be used in our predictive model to 16. The variables would fall under different categories, further explanation of the meaning of each variable is available in the codebook. The selected features are as follows:

- **Response Variable:** Bankrupt
- **Profitability Metrics:** These variables measure a company's ability to generate earnings relative to sales, assets, and equity
    - Return on Assets: roa_a_before_interest_and_after_tax
    - Operating Gross Margin: operating_gross_margin
    - Realized Sales Gross Profit Growth Ratio: realized_sales_gross_profit_growth_rate
    - Continuous Net Profit Growth Rate: continuous_net_profit_growth_rate

- **Liquidity Metrics:** These variables assess the company's ability to meet its short-term obligations
    - Quick Ratio: quick_ratio
    - Cash Flow to Liability: cash_flow_to_liability
    - Cash Flow to Total Assets: cash_flow_to_total_assets
- **Leverage and Solvency Metrics:** These ratios measure the degree to which a company is financing its operations through debt
    - Debt Ratio: debt_ratio
    - Interest Coverage Ratio: interest_coverage_ratio_expense_to_ebit
    - Liability to Equity Ratio: liability_to_equity
- **Efficiency Metrics:** These ratios measure how effectively a company is utilizing its assets
    - Revenue per Person: revenue_per_person
    - Total Income to Total Expense Ratio: total_income_total_expense
- **Growth Metrics:** These ratios measure the growth of a company's assets and net value
    - Total Asset Growth Rate: total_asset_growth_rate
    - Net Value Growth Rate: net_value_growth_rate
    - Long Term Liabilities to Current Assets: long_term_liability_to_current_assets

Show

We will also rename our chosen variables to make it easier to interpret.

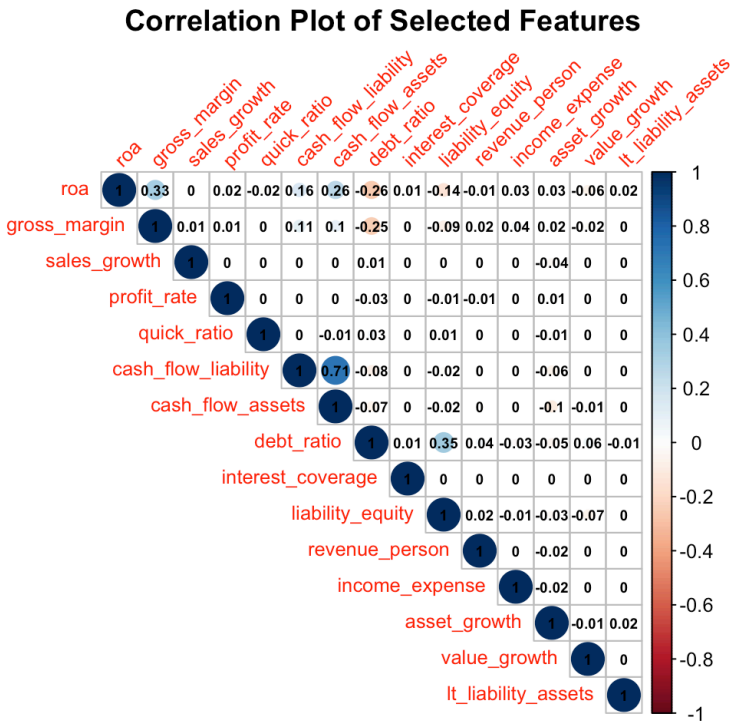
Show

Visual EDA

Correlation Plot

We will now visualize the correlation between the selected features to understand the relationship between the variables.

Show



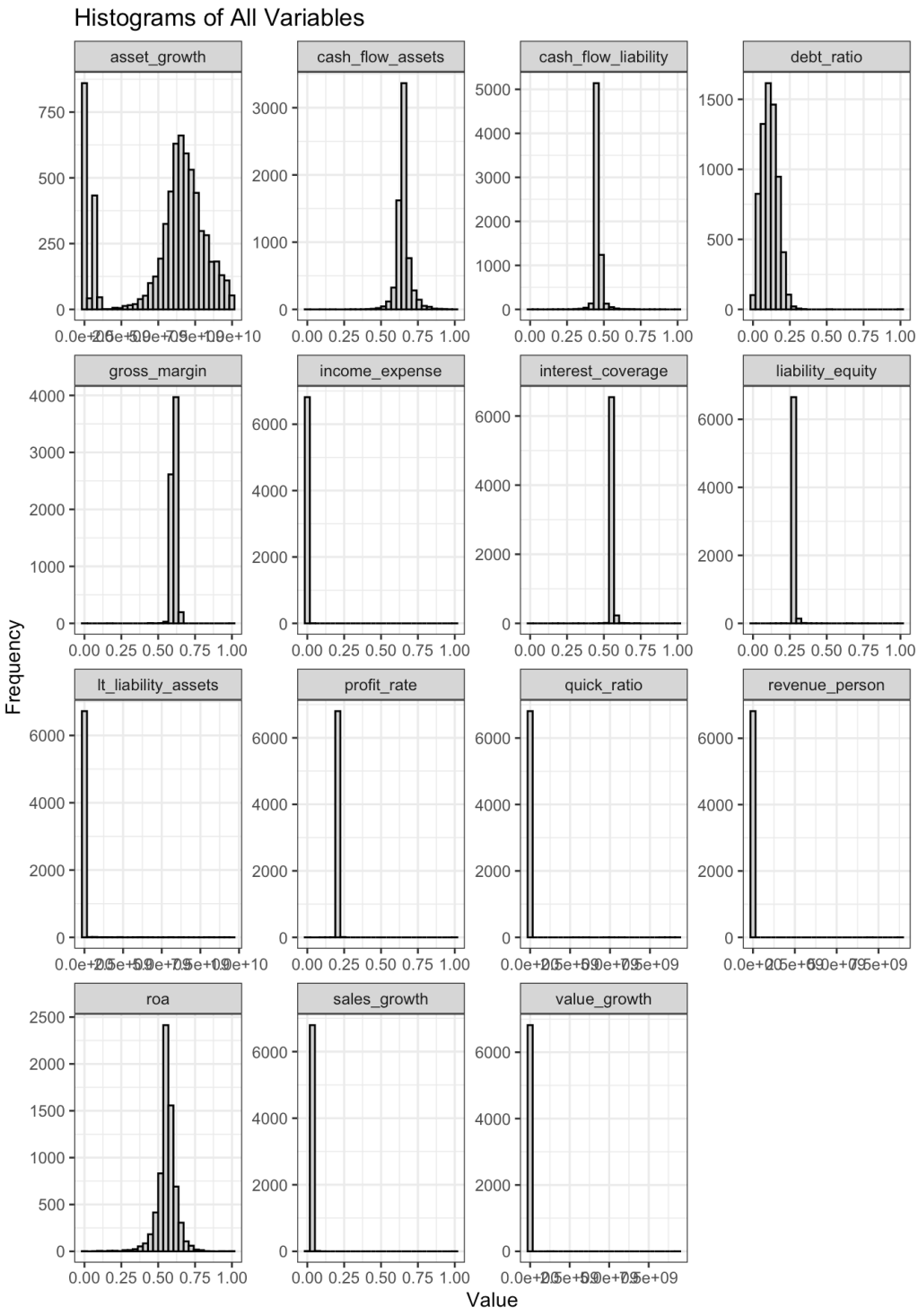
Show

From the correlation graph, we could see that most of the chosen variables have little to no correlation with each other. This is a good sign as it indicates that the features are not redundant and can provide unique information to the model. There are a few exceptions, most notably a strong correlation between cash flow to liability and cash flow to assets which is not surprising.

Distribution of Selected Features

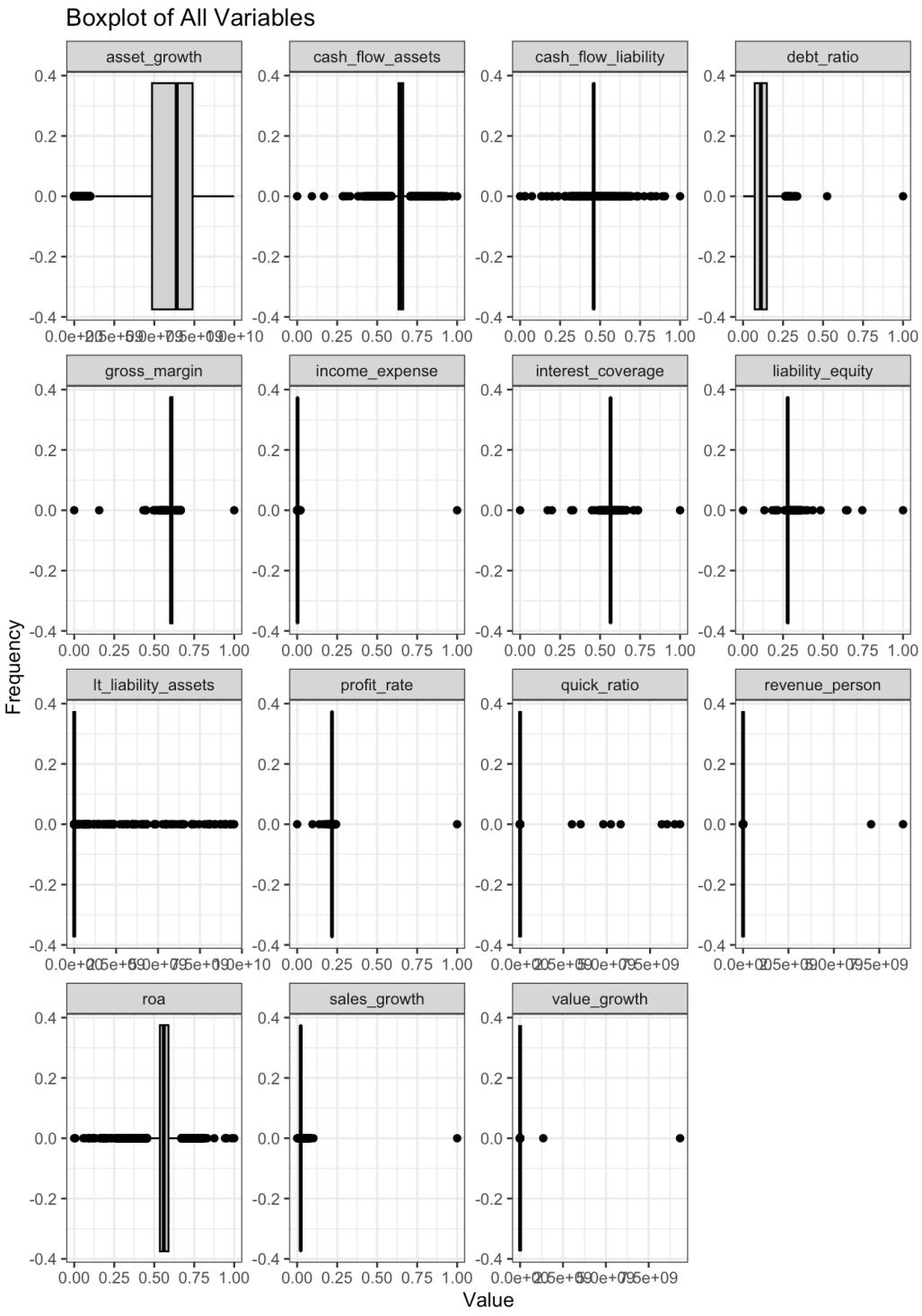
We will continue by visualizing the relationship between the selected features and the target variable ‘bankrupt’. This plot will help us understand the distribution of the features and their relationship with the target variable.

Show



The histograms of variables show that several of the variables (asset growth, cash flow to asset, cash flow to liability, debt ratio, return on asset, etc.) relatively normal distribution. Asset growth rate differs from the other variable distributions as it is mostly normal on values past 2.5e+09 but has a large tail on the left side. However, some variables entail one line of value which requires further analysis and indicates potential outliers. To further investigate the distribution of these variables, we will plot boxplots for each variable.

Show



Show

The boxplots confirm our previous hypothesis, showing that some variables with a single line have extreme outliers. For example, the variables income expense, profit rate, and sales growth rate all have extreme values on the high end. These outliers will need to be addressed to ensure they do not negatively impact the model's performance.

Outliers

Let's examine the entries that have extreme values for the variables with outliers.

Show

Search:

	bankrupt	roa	gross_margin	sales_growth	profit_rate	quick_
1		0.534	0.609	1.000	0.218	
2		0.544	1.000	0.025	0.218	
3		0.224	0.587	0.022	0.217	
4		0.474	0.000	0.022	0.217	
5		0.489	0.596	0.022	0.210	
6		0.578	0.608	0.022	1.000	
7		0.596	0.658	0.022	0.218	
8		0.283	0.638	0.033	0.218	

Showing 1 to 8 of 8 entries Previous 1 Next

We can see that seven of the observations with extreme outliers are not bankrupt. These outliers could heavily skew our data while training the model, additionally, they could represent high leverage points that are not representative of the majority of the data. Given the great proportion of non-bankrupt observations to bankrupt observations we will remove these outliers to ensure the model's performance is not negatively impacted. The extreme outlier observation that did go bankrupt has a net value growth rate of over 9.3 billion, which is indicative of a potential error in the data. We will remove this observation as well.

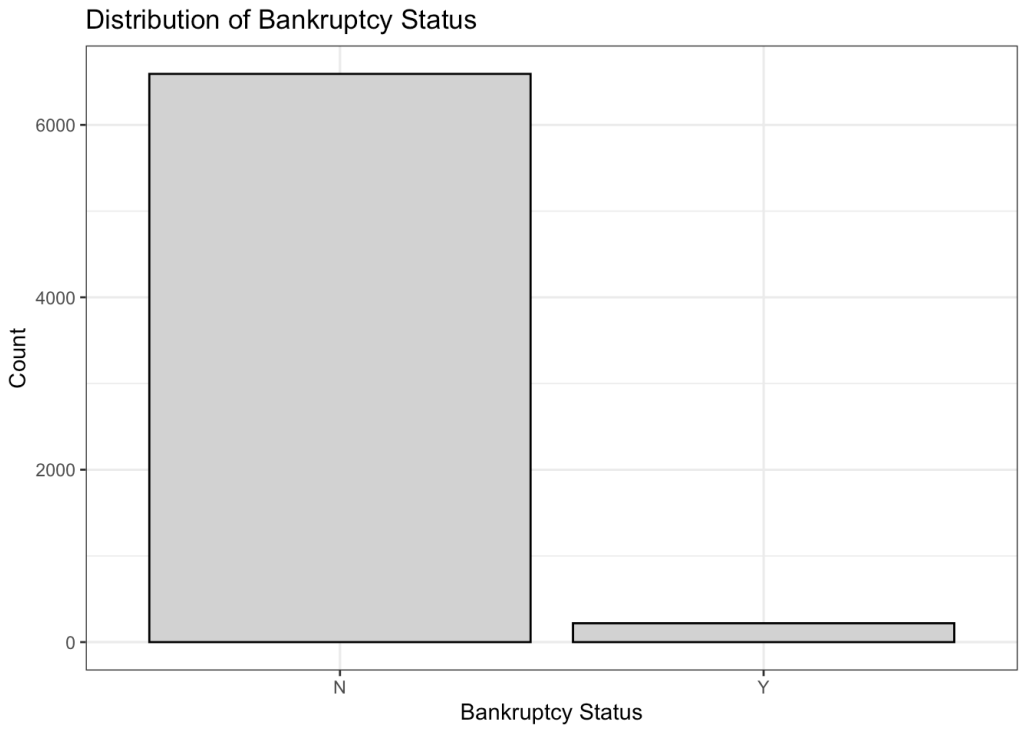
Show

Now we are down to 6811 observations from our original 6819 observations.

Distribution of Bankruptcy Status

We will now explore the distribution of the target variable 'bankrupt' to understand the class distribution.

Show



The distribution shows us that the dataset is imbalanced with a significant higher number of non-bankrupt companies compared to bankrupt companies. This imbalance will need to be addressed during model training and evaluation to ensure accurate predictions.

Setting Up Models

Splitting the Data

We will now perform an initial split on the data and stratify by the response variable ‘bankrupt’ to ensure that the training and testing sets have a similar distribution of bankrupt and non-bankrupt companies. We will split 70% of the data into the training set and the remaining 30% into the testing set. We will set the seed at 123 to ensure reproducibility.

Show

Recipe Building

We will create a recipe that will preprocess the data before training the model. The recipe will standardize the data and remove any potential outliers. We will be creating one centralized recipe as all models will be using the same preprocessing steps. We will use the features we identified earlier in the EDA section to build the recipe. Since we do not have categorical features, we will not need to dummy code any variables. For the numberical predictors, we will center and scale them to ensure that the model is not biased towards any particular variable.

Show

K-fold Cross-Validation

We will also use a v-fold cross-validation with 10 folds to further ensure that the model is trained and evaluated on a variety of data. We will also ensure stratification by the response variable ‘bankrupt’. Cross validation allows our model to be trained and evaluated on multiple

subsets of the data, which helps to ensure that the model is not overfitting to the training data.

Show

Model Building

For this project, we will be building four classification models and evaluating their performance. The models we will be building are: Logistic Regression, Elastic Net Regression, K-Nearest-Neighbor, Linear Discriminant Analysis, Quadratic Discriminant Analysis, and Random Forest. We will evaluate the models based on the AUC ROC metric since our response variable is imbalanced. The models will be built on a separate Rmd file to ensure legibility and organization, as well as ensuring run time is not too long.

The models will be built using the following steps:

1. Setting up the model by specifying its engine, type, and mode.
2. Setting up the workflow by combining the recipe and model.
3. Creating the tuning grid to specify the hyperparameters to be tuned.
4. Tuning the model with specific hyperparameters.
5. Evaluating the model's performance using the ROC AUC metric and finalizing the model's workflow.
6. Fit the selected model on the training data.
7. Saving the model as an RDA file.

Since we are building our models on a separate file, we will need to extract the recipe and the training data to be used in the model building process.

Show

Model Results

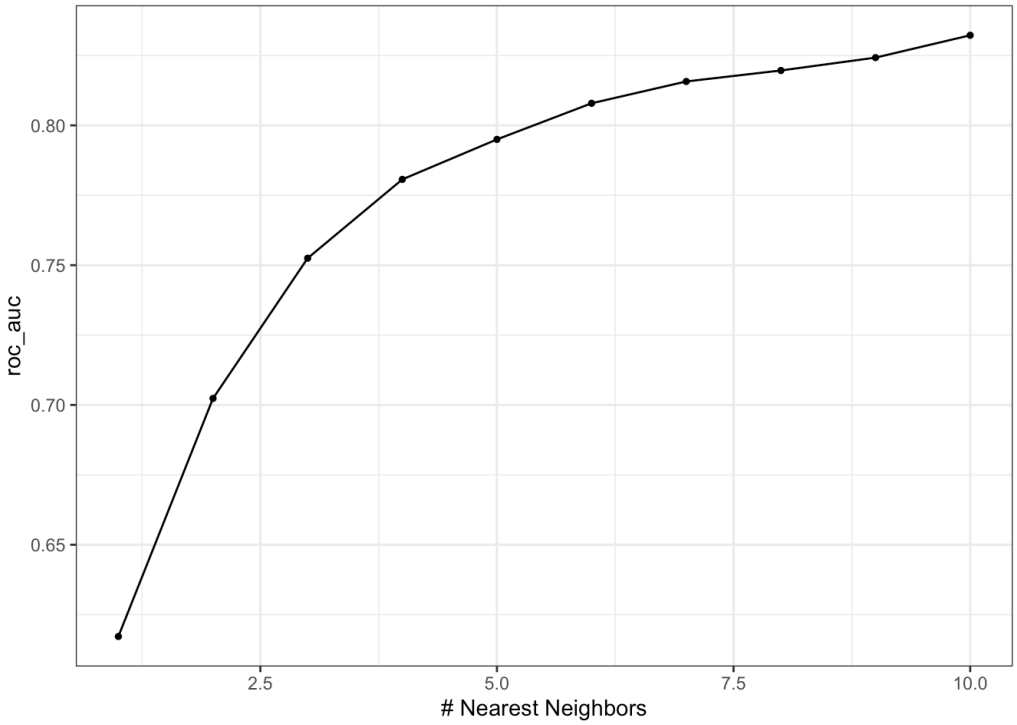
We will load the best performing results into our model. Their ROC AUC will be used to compare the models' performance between parameters and across models. This would help us narrow down and determine the best model to conduct our analyses on. For our logistic regression, LDA, and QDA models, there are only one set of hyperparameters, so we will not need to tune them. We will examine the ROC AUC of the other models with their differing hyperparameters then compare all the models in the end.

Show

K-Nearest-Neighbor

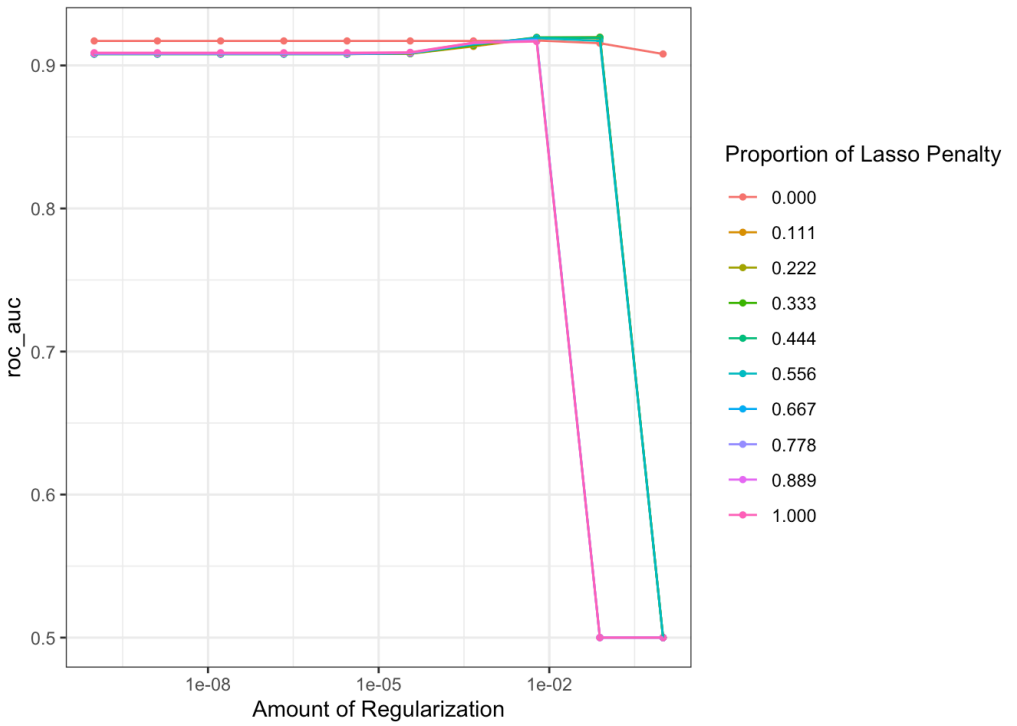
Apparent from the plot, as the number of nearest neighbors within the model increases, the ROC AUC also increases. The best performing KNN model has 10 nearest neighbors with an average ROC AUC of 0.832 and standard error of 0.021.

Show



Elastic Net Regression

Show

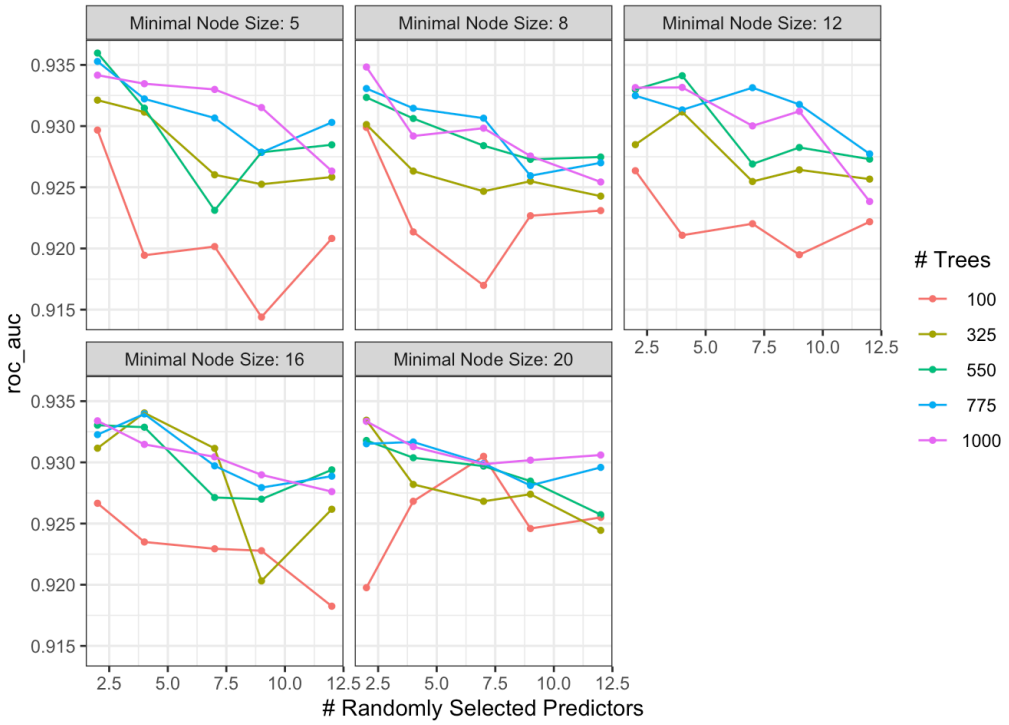


The Elastic Net Regression model shows that generally the lower lasso penalty models perform better than the higher penalties. Furthermore, the ROC AUC decreases sharply as the amount of regularization increases in the model after a certain point. The best performing Elastic Net Regression model has a lasso penalty of 0.0774 and a mixture of 0.222 that

produces an average ROC AUC of 0.920 and standard error of 0.0098. The Elastic Net Regression model outperforms the knn model in the ROC AUC's average while having a smaller standard error.

Random Forest

Show



The Random Forest model does not display as clear as trend as the other models, however, we can see that generally as the number of predictors increase, the ROC AUC tends to decrease. The number of trees in the forest does not have a consistent leader other than the 100 trees model which generally has the lowest ROC AUC. The best performing Random Forest model has 550 trees, 2 randomly selected variables, and minimal node size of 5 with a mean ROC AUC of 0.924 and standard deviation of 0.00895. The Random Forest model is considered the best performing model compared to the KNN and Elastic Net Regression models.

Model Comparison

We will now compare the best performing models from each of the six models we built and determine the best models to use for our analysis. We will compare the models based on their average ROC AUC and standard deviation. The ROC AUC will give us an overall idea of how well the model performs and will give us an idea of how well the model predicts bankrupt and non-bankrupt companies respectively.

Show

```
## # A tibble: 6 × 3
##   model                            mean std_err
##   <chr>                           <dbl>   <dbl>
## 1 Random Forest                   0.936 0.00895
## 2 Elastic Net Regression          0.920 0.00975
## 3 Logistic Regression             0.908 0.0109
## 4 Linear Discriminant Analysis    0.904 0.0147
## 5 K-Nearest-Neighbor              0.832 0.0208
## 6 Quadratic Discriminant Analysis 0.825 0.0165
```

The table displays that the best performing model for the ROC AUC mean and standard error is our Random Forest model. The second best performing model is the Elastic Net Regression model, followed by the Logistic Regression. The Random Forest model has the highest ROC AUC mean and the lowest standard error, making it the best model to use for our analysis. There is a notable drop off in model performance between the Linear Discriminant Analysis and KNN model.

To proceed, we will conduct our analyses and further evaluate our Random Forest model.

Model Result on Test Data

We will now evaluate the Random Forest model on the test data to determine how well the model generalizes to new data. We will use the ROC AUC as our main metric to evaluate the model's performance on the test data. We have fitted the Random Forest model on the training data in the model building section, so we will use the fitted model already loaded from the Model Building Rmd.

Metric Performance

The Random Forest model has an ROC AUC of 0.968 which is an improvement from the training data. The Random Forest model's high ROC AUC, indicates that the model has a high ability to differentiate the companies that will go bankrupt and those that will not. The model does an excellet job at ranking bankruptcy probability correctly. The F1 score is 0.994, which is a very high score and indicates that the model effectively balances recall and precision. Thus, it not only identifies most actual bankruptcies but also maintain a low rate of false positives. Thus, the model could be reliably used to predict bankruptcy without excessive risk of false predictions.

Show

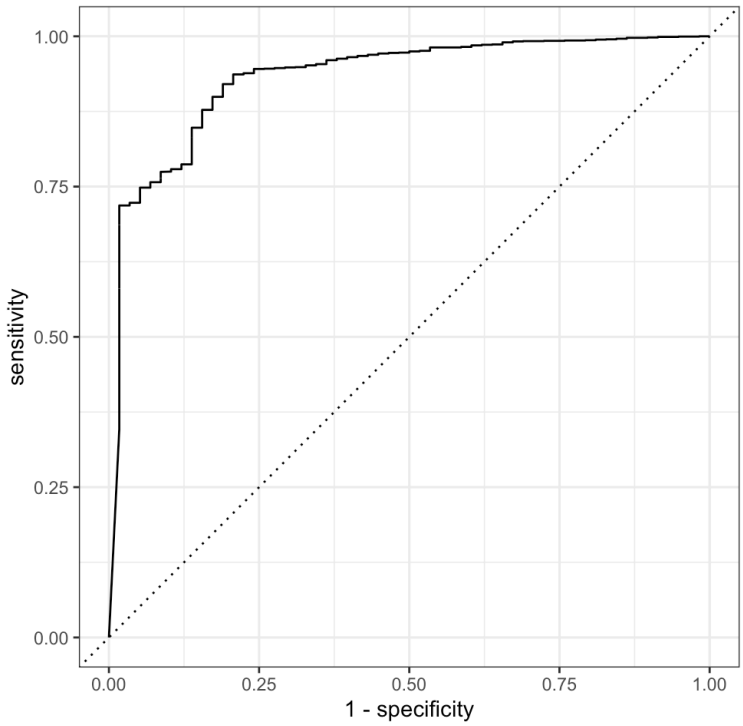
```
## # A tibble: 2 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.931
## 2 f_meas  binary         0.986
```

ROC Curve

The ROC curve also demonstrates the Random Forest model's exceptional performance in predicting bankruptcy. The curve is close to the top left corner, indicating that the model has a high true positive rate and a low false positive rate. The steep initial climb indicates that the model achieves high sensitivity without incurring many false positives, which is highly desirable in our model.

Show

Show



Confusion Matrix

The confusion matrix gives us further insight into our model performance. The matrix shows a high degree of accuracy in predicting not bankrupt companies with only 1 miscategorization, however the model has a higher rate of miscategorization for bankrupt companies with 22 of the 58 bankrupt observations misclassified. Therefore, the matrix reveals that the model is at risk of false negative predictions which can be considered risky depending on the risk tolerance of potential users of the model. Potential next steps to adjust for the false negatives could be to adjust the threshold of the model to increase the sensitivity of the model to bankrupt companies although it could reduce the precision of our model.

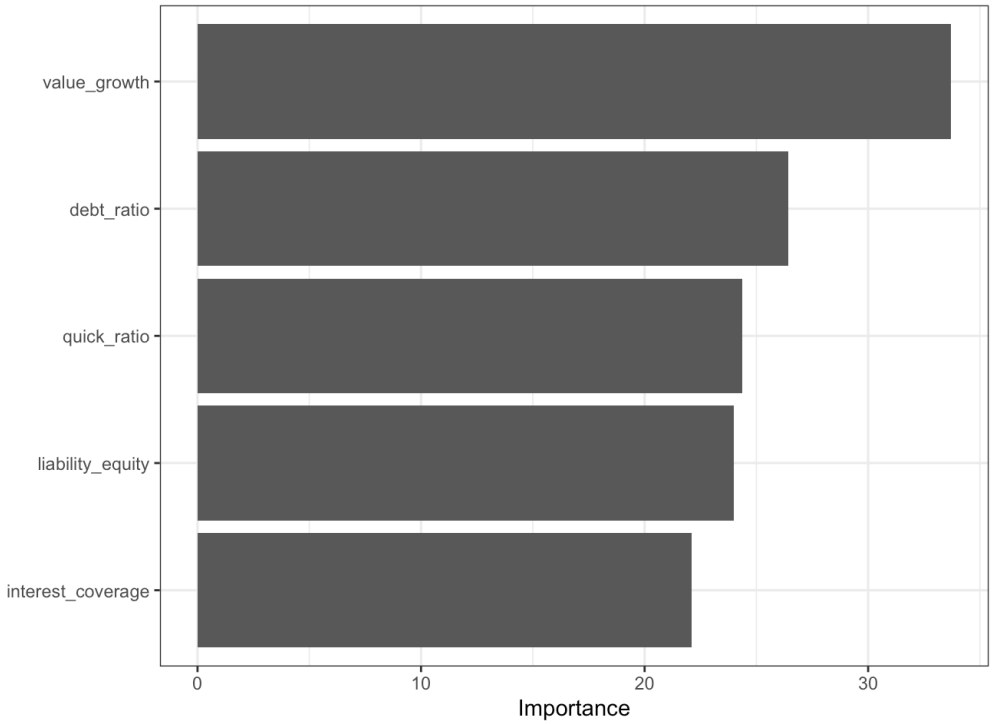
Show

```
##            Truth
## Prediction    N    Y
##          N 1982   52
##          Y    4    6
```

Variable Importance Plot

The variable importance plot shows the top 10 most important variables in the Random Forest model. The most important variable is the value_growth variable by a noticeable margin. The value_growth variable is followed by the debt_ratio and quick_ratio. The variable importance plot gives us an idea of which variables are most important in predicting bankruptcy and would be meaningful predictors in our model.

Show



Conclusion

After exploring our raw data and engaging in data preprocessing, we selected 15 variables to build our models. Using the 15 variables, we built six models: Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest-Neighbor, Elastic Net Regression, and Random Forest. We tuned the hyperparameters of the K-Nearest-Neighbor, Elastic Net Regression, and Random Forest models to optimize their performance. We then evaluated the models based on their ROC AUC and selected the Random Forest model as the best performing model. We evaluated the Random Forest model on the test data and found that the model had an ROC AUC of 0.968 and an F1 score of 0.994. The model had a high true positive rate and a low false positive rate, indicating that the model is highly effective at predicting bankruptcy. The model had a high degree of accuracy in predicting non-bankrupt companies, but had a higher rate of miscategorization for bankrupt companies. The variable importance plot showed that the value_growth variable was the most important variable in predicting bankruptcy. The Random Forest model is a highly effective model for predicting bankruptcy and could be used to predict bankruptcy with a low risk of false predictions.

For future work, we could explore other models such as Support Vector Machines and Neural Networks to see if they can outperform the Random Forest model. We could also explore other methods of feature selection to see if we can improve the model's performance since there are a lot of variables and datapoints available for analysis. We could also explore other metrics such as precision, recall, and F1 score to evaluate the model's performance. Additionally, we could process our original data further to better account for the distributional differences in bankrupt and non-bankrupt companies through techniques like Synthetic Minority Oversampling Technique as used by Ginelle D'Souza (https://www.kaggle.com/code/ginelledsouza/bankruptcy-analysis#Data-Modeling) Overall, the Random Forest model is a highly effective model for predicting bankruptcy and could be used to predict bankruptcy with a low risk of false predictions.

References

The database is downloaded from Fedesoriano Kaggle (https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction) which is sourced from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/572/taiwanese+bankruptcy+prediction).

Additional coding help comes from PSTAT 131 Slides and Labs, Arthur Kim, GitHub Copilot, and ChatGPT.