Ian Brandenburg & Zsófia Rebeka Katona
https://github.com/Iandrewburg/DA3_Brandenburg/tree/main/Assignment_3

Default Prediction of Firms for 2015: Technical Report

Introduction

The manufacturing industry for computer, electronic, and optical products may fluctuate frequently, and with the correct modeling, these fluctuations can be forecasted to an extent. This project aims at developing predictive modeling to determine which firms may fail in 2015. This is done by utilizing company data from 2010 through 2013 as training data for predicting a holdout sample for the year 2014.

Detailed company **data** from a middle-sized country in the European Union All registered companies in 2005-2016 in three selected industries (auto manufacturing, equipment manufacturing, hotels and restaurants) This rich database was constructed from multiple publicly available sources by Bisnode, a business data and analytics company for educational purposes.

Methodological Overview and Data Prep

Overview of analytical process
The 2005-2016 data was loaded into GitHub as a CSV, and called directly from the GitHub link into the Python code for easier collaborative purposes. The data was split into a training sample (2010-2013) and a holdout sample (2014). The data from here underwent feature and label engineering, filtering and data cleaning. This allowed for more enhanced predictive modeling. Four different predictive models were run on the training set to determine which model would run the most effectively. These four models included logistic regression, LASSO, Random Forest, and Gradient Boosting.

Variable Descriptions and Management
The dataset included a large array of company dataset variables that were employed for this study. The variables are listed below, in addition to how they were managed for this study. The variables were managed through log transformations, dummy transformations, 0 and median imputation, column and row dropping due to missing values, and winsorizing tails. These included:

- **comp_id:** Company ID, used for data sorting.
- **begin:** Beginning of the period. Converted to ‘datetime’ variable in data cleaning.
- **end:** End of the period. Converted to ‘datetime’ variable in data cleaning.
- **COGS:** Cost of Goods Sold. Dropped due to too many missing values.
- **amort:** Amortization. Missing values imputed with 0.
- **curr_assets:** Current Assets. Missing values imputed with 0; negative values transformed to 0.
- **curr_liab:** Current Liabilities. Missing values imputed with 0; included in balance sheet ratios.
- **extra_exp:** Extraordinary Expenses. Missing values imputed with 0; scaled by sales, flagged for high values and errors.
- **extra_inc:** Extraordinary Income. Missing values imputed with 0; scaled by sales, flagged for high values and errors.

- **extra_profit_loss:** Extraordinary Profit/Loss. Missing values imputed with 0; scaled by sales, flagged for high/low values, errors, and included quadratic term.
- **finished_prod:** Finished Products. Dropped due to too many missing values.
- **fixed_assets:** Fixed Assets. Missing values imputed with 0; negative values transformed to 0.
- **inc_bef_tax:** Income Before Tax. Missing values imputed with median; scaled by sales, flagged for high/low values, errors, and included quadratic term.
- **intang_assets:** Intangible Assets. Missing values imputed with 0; negative values transformed to 0.
- **inventories:** Inventories. Missing values imputed with 0; scaled by sales and flagged for high values and errors.
- **liq_assets:** Liquid Assets. Missing values imputed with 0.
- **material_exp:** Material Expenses. Missing values imputed with 0; scaled by sales and flagged for high values and errors.
- **net_dom_sales:** Net Domestic Sales. Dropped due to too many missing values.
- **net_exp_sales:** Net Export Sales. Dropped due to too many missing values.
- **personnel_exp:** Personnel Expenses. Missing values imputed with 0; scaled by sales and flagged for high values and errors.
- **profit_loss_year:** Profit/Loss for the Year. Missing values imputed with median; scaled by sales, flagged for high/low values, errors, and included quadratic term.
- **sales:** Sales: Transformed to millions (sales_mil) and natural logarithm applied (ln_sales, sales_mil_log); negative sales values replaced with 1; missing values imputed with median.
- **share_eq:** Shareholder Equity. Missing values imputed with median; included in balance sheet ratios.
- **subscribed_cap:** Subscribed Capital. Missing values imputed with median; included in balance sheet ratios.
- **tang_assets:** Tangible Assets. Missing values imputed with 0.
- **wages:** Wages. Dropped due to too many missing values.
- **D:** Dropped due to missing information. A new ‘default’ variable was created instead.
- **year:** Year of data. Used in calculating company age. Verified for numeric consistency during data cleaning.
- **founded_year:** Used in calculating company age.
- **exit_year:** Used in calculating company age. Verified for numeric consistency during data cleaning.
- **ceo_count:** CEO Count. Rows with missing values were dropped during data cleaning.
- **foreign:** Indicator for foreign ownership. Transformed to foreign_management as dummies. Rows with missing values were dropped during data cleaning.
- **female:** Indicator for female CEO. Rows with missing values were dropped during data cleaning.
- **birth_year:** CEO's Birth Year. Used in calculating the CEO's age. Verified for numeric consistency during data cleaning.
- **inoffice_days:** Days in office. Rows with missing values were dropped during data cleaning.
- **gender:** Gender of CEO. Transformed to a categorical variable (gender_m). Rows with missing values were dropped during data cleaning.
- **origin:** Origin of CEO. Rows with missing values were dropped during data cleaning.

- **nace_main:** Main NACE classification. Rows with missing values were dropped during data cleaning.
- **ind2:** Industry classification code 2. Filtered to "ind2 == '26'". Rows with missing values were dropped during data cleaning.
- **ind:** General industry classification. Dropped due to lack of necessity in this project.
- **urban_m:** Urban metric. Converted to a categorical variable during data manipulation.
- **region_m:** Region metric. Missing values filled with 'Missing', transformed to a categorical variable (m_region_loc) during data cleaning.
- **founded_date:** Date of company foundation. Converted to 'datetime' during data cleaning.
- **exit_date:** Date of exit. Dropped due to too many missing values. This variable was also not necessary in this study, as we had a default variable to determine this.
- **labor_avg:** Average number of employees. Dropped due to too many missing values.

Creation of the Dependent Variable: `default`

The dependent variable being tested in this project is the 'default' variable, which was developed in the Label Engineering section of the code. This was a dummy variable where 1 represented a company that defaulted and 0 meant the company stayed alive. This involved organizing the data by the "year" and "company ID" variables, and determining if the company had sales two years later. If the company had 0 sales two years later, this would suggest that it defaulted in the following year. This variable was created prior to the data cleaning process so that it could analyze negative or 0 values in sales prior to adjustments being made in the data cleaning process.

Additional Financial Ratio Variables

In addition to the log transformations, imputations, flagging and winsorizring tails noted above in the variable managing, additional financial ratio variables were calculated using the data provided. These include the following calculations:

- **Gross Profit Margin** = (sales - (material_exp+personnel_exp))
- **Net Profit Margin** = profit_loss_year / sales
- **Return on Equity** = profit_loss_year / share_eq
- **Debt-Equity Ratio** = curr_liab / share_eq
    - Should have been total liabilities instead of current liabilities, but this was not available
- **Current Ratio** = curr_assets / curr_liab
- **Quick Ratio** = (curr_assets - inventories) / curr_liab
- **Return on Assets** = profit_loss_year / total_assets_bs

Splitting the Dataset into the Training Set and Holdout Sample

The primary dataset was split into a training set and a holdout sample. The training set includes data from 2010 through 2013. The reason why the dataset is not extended to before 2010 is because this could generate a large amount of noise in the predictive modeling, as the economics change drastically over this amount of time, and could influence the default rate of firms. The training set was filtered to show firms with sales ranging from 1000 euros to 10 million euros. Additionally, the training sample was set to the industry code 26, since including other industries in this predictive modeling could increase the amount of noise as well.

The holdout sample was filtered to reflect 2014 data, while being filtered in sales from 1000 euros to 10 million euros, and specifying the industry code of 26. The following table shows the basic descriptive statistics for the holdout sample after it had been generated:

	Number of Firms	Mean Sales	Min Sales	Max Sales	Defaulted Firms	Stayed Alive Firms
0	1037	490202.217927	1070.370361	9576485.0	56	981

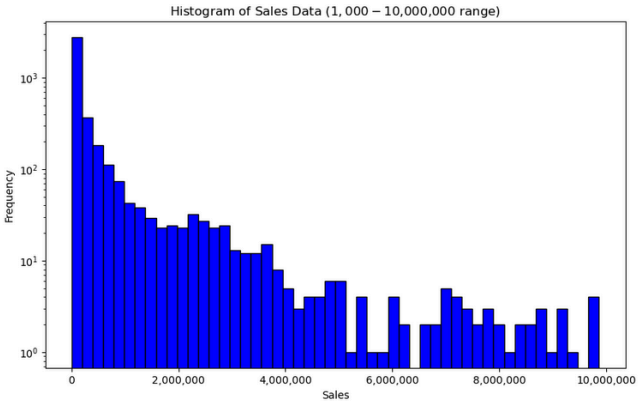
Data Cleaning Function: `data_dish_washer`
A data cleaning function was developed specifically for this project. This function was named `data_dish_washer` to make its functionality clear, and has nine primary steps throughout the function.

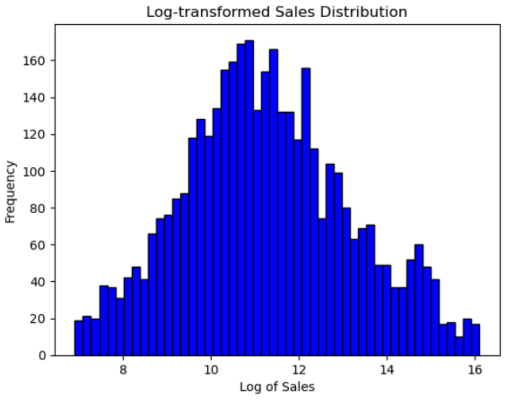
- **Step 1**: Imputing missing values in selected numerical columns with 0.
- **Step 2**: Imputing missing values in selected numerical columns with median.
- **Step 3**: Dropping rows in selected categorical columns with missing values
- **Step 4**: Filling missing values in selected categorical columns with 'Missing'
- **Step 5**: Convert selected date variables to datetime values
- **Step 6**: Set selected variables to numeric values
- **Step 7**: Transforming infinite values with NaN
- **Step 8**: Fill NaN values with 0
- **Step 9**: Drop columns with too many missing values in the original data

After the data_dish_washer function was executed on the training dataset and holdout sample, another function was run on these datasets that displays the basic descriptive details of the datasets. This custom function was titled `data_sherlock`, since it inspects the data. Pre-cleaning, the training dataset had 4,353 total rows, 38,798 null values, and 185 infinite values. Post-cleaning training dataset had a total of 3,956 rows, and 0 null or infinite values. Pre-cleaning, the holdout sample had 1,037 total rows, 8,599 null values, and 43 infinite values. Post-cleaning, the holdout sample had 1,006 total rows, and 0 null or infinite values.

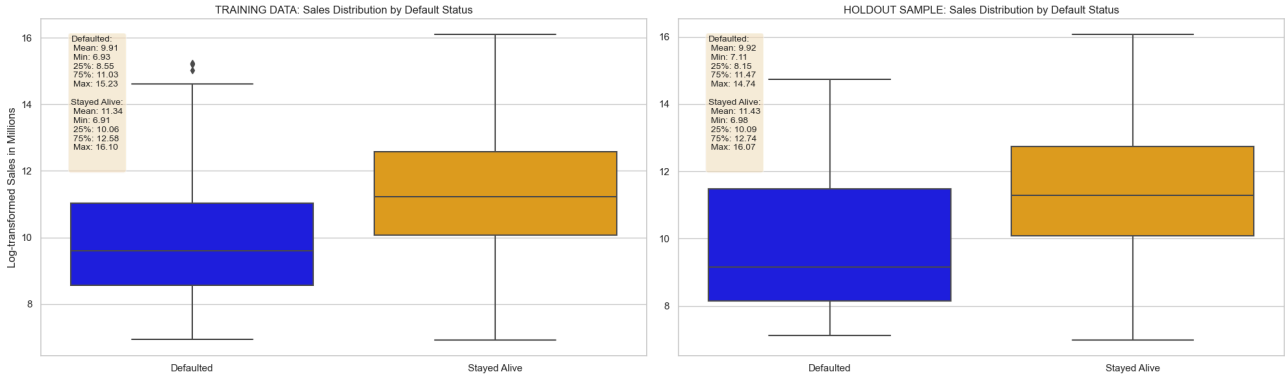
Important Variable Descriptives
The sales variable is potentially one of the more important variables in this dataset, as sales less than or equal to 0 will cause a company to default. A histogram analysis was conducted to validate the log transformation of the sales variable, as it is initially heavily skewed. This can be seen in the following histograms, with the left being pre-transformation, and the right being the log-transformed sales:

Ian Brandenburg & Zsófia Rebeka Katona
https://github.com/Iandrewburg/DA3_Brandenburg/tree/main/Assignment_3



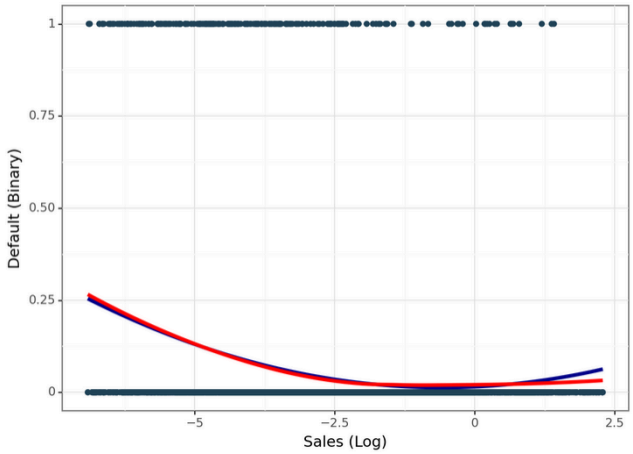


Additionally, a box plot was created to analyze the differences in sales between companies that defaulted and those that stayed alive. This box plot was run on the holdout sample and training set separately to help validate the two datasets.



It can be seen here that the sales in the defaulted companies is lower than in the companies that stayed alive. This is not conclusive or causal, but rather a visual representation of the dataset. Using box plots in this style helps visualize the distribution of continuous variables across a dummy variable.

Finally, to further visualize the association between the logarithmic sales and default variable, a lowess plot was generated, which displays lower sales increases the possibilities of a company defaulting. Ggplot was used to plot this figure:



Ian Brandenburg & Zsófia Rebeka Katona
https://github.com/Iandrewburg/DA3_Brandenburg/tree/main/Assignment_3

Model Development

The variables were grouped into different lists for the purpose of distributing them across the models. Here are the variable groups that were formed.

- **Raw Variables (rawvars):** Basic financial figures from balance sheets and profit/loss statements, forming the foundation for initial analyses.
- **Quality Variables (qualityvars):** Indicate the completeness and reliability of the financial data, aiding in assessing data quality.
- **Engineered Variables (engvar, engvar2, engvar3):** Derived metrics that provide deeper insights into a firm's financial health by normalizing raw figures, creating ratios, and flagging outliers or anomalies.
- **Financial Ratios (financial_ratios):** Standard industry metrics that compare different financial variables, giving a clear picture of a firm's performance relative to its size or assets.
- **Human Resource Variables (hr):** Capture the demographic and management structure's influence on firm performance.
- **Firm Variables (firm):** Include firm age and regional location, reflecting the firm's experience and environmental context.
- **Interactions:** Explore how combinations of variables (like financial metrics and CEO age) interplay, offering nuanced insights into factors driving firm performance.

The variable groups were then split up into nine different model setups. The following summarizes the intentions of the model groups:

- **M1:** Focuses on fundamental sales data and profitability variables.
- **M2:** Adds balance sheet data and demographic information in addition to the M1 variables, enhancing the financial perspective.
- **M3:** Incorporates firm specifics and engineered variables for a detailed financial health outlook.
- **M4 & M5:** Extended to include interactions and more complex variables, aiming to capture specific relationships potentially associated with firm performance.
- **M6:** Prioritizes variables with high feature importance, focusing on those significantly impacting the model's predictive ability. These feature importance were derived from the random forest modeling conducted later in the project.
- **Logit Lasso:** Incorporates sales data, engineer variables (quadratic and flagged), human resources factors, firm specifics, quality indicators, financial ratios and interaction terms. This set of variables aims to utilize lasso's ability to select the most relevant features to avoid and reduce overfitting.
- **Random Forest:** Merges raw financial data with sales changes, demographic and firm-specific variables, quality indicators, and financial ratios. This variable set employs a wide array of variables, including interactions, to leverage regularization and random forest's feature selection capabilities.
- **Gradient Boosting:** This model utilizes the entire set of variables in the dataset to examine the predicting capabilities of all variables. Although this is not typical, it will allow for a comparison between the other models run on the entire dataset.

Each of the models underwent 5 k-fold cross-validation to measure the models against each other and determine which model was the best. The models were first run on the training set to determine how they performed, with the cross validated RMSE, AUC, Optimal Threshold and Expected Loss values being collected and analyzed. Finally, the top three trained models were tested on their predictive abilities on the holdout sample. For all the models, the false positive cost is set to 3 and the false negative cost is set to 15, signifying a greater amount of damage generated from a false negative.

Logistic Regression and LASSO Models

Logit models were run for M1-M6 to compare the performances of the different x-variable groups. The C-value was set to a very high value in order to turn off regularization, allowing for the logistic model to fit the training data as closely as possible. 5-fold cross validation was conducted on the models, with all the models performing similarly in their RMSE values. The LASSO variables were added to the modeling to determine how the LASSO model would perform.

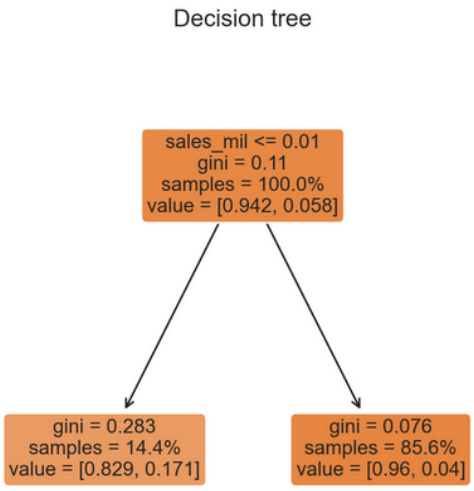
Next, the AUC values were calculated for each fold of the logistic regression models as well as the LASSO model. These values were placed into a data frame that included the number of coefficients in each model, and their average CV RMSE and AUC values.

	Number of Coefficients	CV RMSE	CV AUC
M1	6	0.229471	0.719717
M2	12	0.228539	0.754033
M3	26	0.228769	0.742763
M4	86	0.232336	0.736586
M5	79	0.225919	0.770994
M6	12	0.227458	0.733060
LASSO	19	0.225867	0.776757

M2 was selected as the best, as the RMSE and AUC values for each of the models were all very similar to each other. The M2 model has a more manageable number of coefficients, so it was selected as the best from this group of models.

Random Forest Model

The random forest model was developed by initializing design matrices from the training set and holdout sample. A decision tree was created prior to the random forest model being run, which included selected features.



Our decision tree consists of one terminal node and two other nodes. Despite setting the max_depth to 3, our tree has only two branches and splits the data once. The impurity of the terminal node is higher, the other nodes represent more homogeneous subsets of the data, hence the lower Gini indexes.

A probability forest was then developed and tuned. For each combination of hyperparameters, the Random Forest algorithm is fitting multiple decision trees and evaluates their performance based on the Gini coefficient. The averaging over trees refers to aggregating the predictions made by each individual decision tree in the ensemble to produce a final prediction. Furthermore, a grid was created to search through the specified hyperparameter combinations during model training. When the GridSearchCV was run on the Random Forest variables, a timer was added to keep track of the status of the code being run.

The cross validated RMSE and AUC were calculated, and the best parameters for the RF model were defined, with the max features being 6 and the minimum samples split is 11. Furthermore, a feature importance bar chart was developed to see which variables have the most significant influence on the predictability of the model. The CV AUC and RMSE values added to the summary table. Additionally, AUC, optimal threshold, and calibration curves were generated for the Random Forest model, which can be found in the appendices.

Gradient Boosting Model

The false positive and false negative costs were re-initialized, with the code being so dense just to verify their values. Additionally, since the gradient boosting model is being run on all the variables in the dataset, the columns are set to just drop the dependent variables `default`. Furthermore, only numerical features were selected in order to avoid any errors when the model was being run.

The code initializes the k-fold for 5-fold cross-validation, as well as the gradient boosting classifier which allows for the gradient boosting model to be run. The CV AUC, RMSE, Optimal Threshold, and Expected Loss are calculated for the gradient boosting model and appended to the summary table.

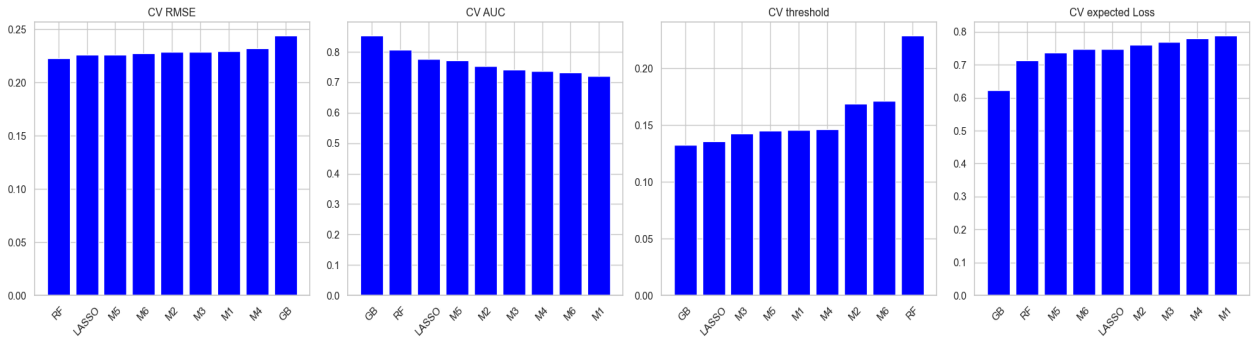
Ian Brandenburg & Zsófia Rebeka Katona
https://github.com/Iandrewburg/DA3_Brandenburg/tree/main/Assignment_3

Model Evaluation and Selection

The number of coefficients, CV RMSE, AUC, optimal threshold and expected loss were calculated and visualized in a summary table to compare all the models performances on the training data. The table is as follows:

	Number of Coefficients	CV RMSE	CV AUC	CV threshold	CV expected Loss
M1	6.0	0.229471	0.719717	0.145796	0.789422
M2	12.0	0.228539	0.754033	0.168825	0.759846
M3	26.0	0.228769	0.742763	0.142718	0.768944
M4	86.0	0.232336	0.736586	0.146017	0.779559
M5	79.0	0.225919	0.770994	0.145000	0.737863
M6	12.0	0.227458	0.733060	0.171528	0.746953
LASSO	19.0	0.225867	0.776757	0.135601	0.747717
RF	n.a.	0.222828	0.806722	0.228923	0.712832
GB	n.a.	0.244355	0.854283	0.132717	0.621840

This table was further visualized through the use of bar charts:



The bar charts were created by using a 'for' loop that references the last four columns of the table through indexing. Additionally, a custom function was created so that the code would rank certain values as either ascending true or ascending false, with the RMSE, optimal threshold, and expected loss being ranked from lowest to highest, and the AUC from highest to lowest. All of these values are just the training model values. Here, it seems as though the gradient boosting model has the lowest expected loss and optimal threshold, while maintaining the highest AUC; however, it has the highest RMSE. All of these could be attributed to the volume of variables included in the gradient boosting model.

The Random Forest performed very well on the training data, as it comes in second to the gradient boosting model in the expected loss and AUC, as well as having the lowest RMSE. The best three models, which were the gradient boosting, random forest, and logistic regression M2, were then tested on predicting the holdout set to determine which model most accurately predicts the holdout sample.

Testing the models on predicting the holdout sample

The three models tested on the holdout sample included the gradient boosting model, random forest model, and M2 logistic regression model. The model that had the best expected loss was the random forest model, with an expected loss of 0.584 on predicting the holdout sample. As this model had the best expected loss value, all the other metrics for predictability were calculated and displayed in a table. The metrics calculated included: RMSE (0.203), Brier Score (0.041), ROC AUC (0.836), Accuracy Score (0.936), Sensitivity (0.340), Specificity (0.968), Expected Loss (0.584), Optimal Threshold (0.229), Total Number of firms in the holdout set (1006), number of firms defaulted (50), number of firms stayed alive (956), means sales (480012.011), minimum sales (1070.37), and maximum sales (9,576,485). These are displayed in the following table generated by the code as well:

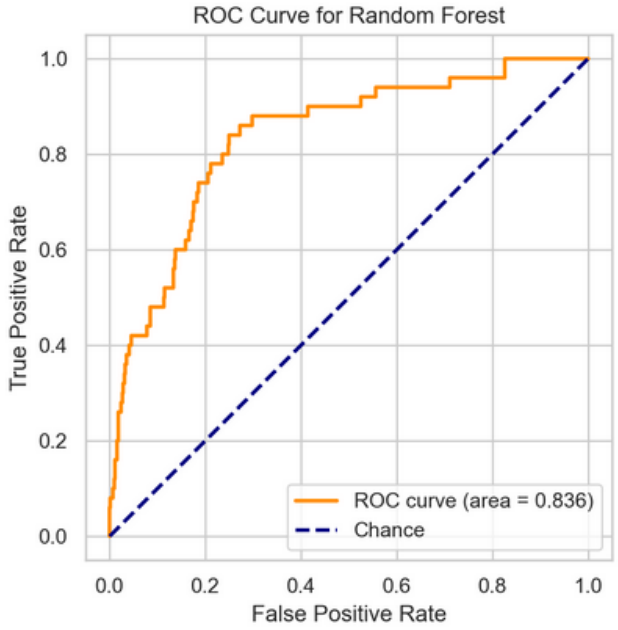
Trained Random Forest Predicting Holdout Sample Metrics

	Metric	Value
0	RMSE	0.203
1	Brier Score	0.041
2	ROC AUC	0.836
3	Accuracy	0.936
4	Sensitivity	0.340
5	Specificity	0.968
6	Expected Loss	0.584
7	Optimal Threshold	0.229
8	Total Number of Firms on the holdout set	1006.000
9	Firms Defaulted	50.000
10	Firms Stayed Alive	956.000
11	Mean Sales	480012.011
12	Min Sales	1070.370
13	Max Sales	9576485.000

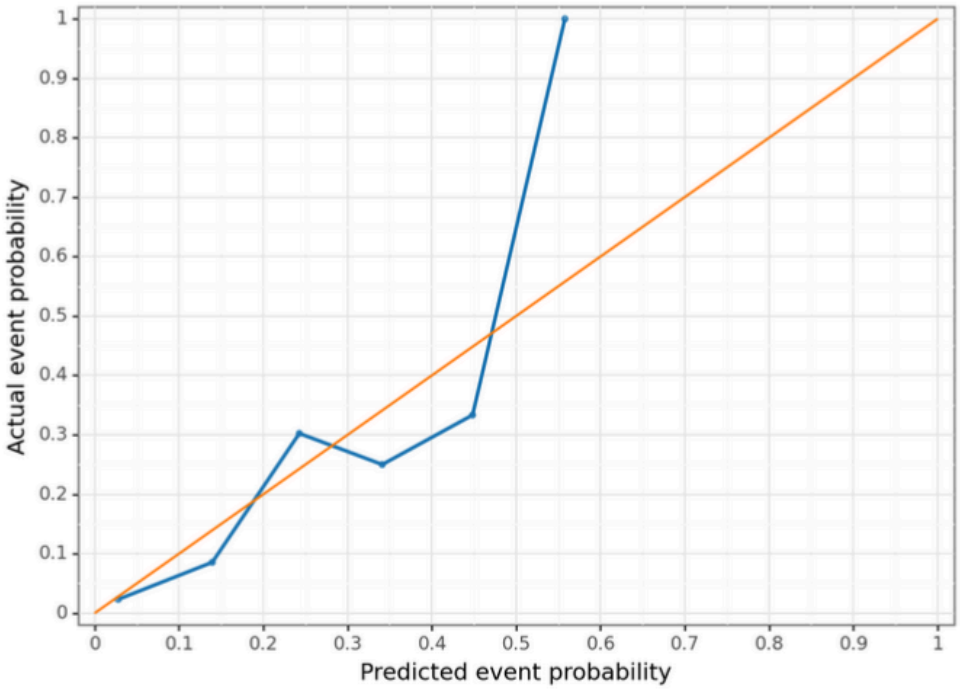
In addition to reporting the metrics for the random forest model predicting the holdout sample, a confusion matrix was developed to display the accuracy of prediction in a more interpretable sense:

	Predicted Stayed Alive	Predicted Default
Actual Stayed Alive	925	31
Actual Default	33	17

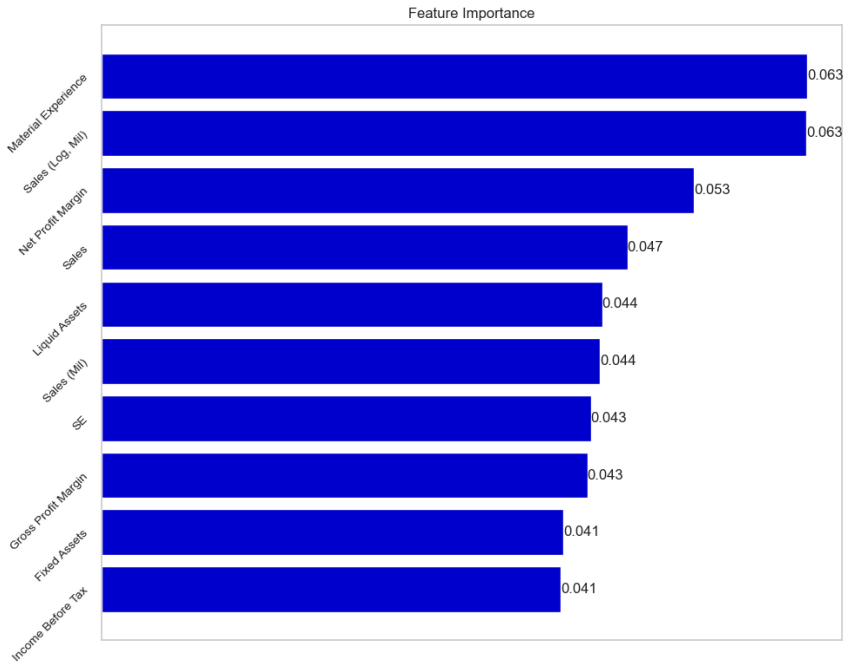
The ROC curve for the random forest predicting the holdout sample was generated:



A calibration curve was generated for the random forest model predicting the holdout sample, which shows that the data is reasonably calibrated, but could still be improved:



Finally, a feature importance bar chart was generated for the random forest model to display the most important features when it comes to predicting. These are found in the chart below:



The feature importance values for the Random Forest model show that the most impactful variables are `material_exp`, `d1_sales_mil_log` and `net_profit_margin`. `inc_bef_tax` and `foreign_management` have lower importance values, suggesting that factors beyond simple revenue figures play important roles in determining if a company defaulted or not.

The same calculations were conducted on the gradient boosting and M2 logistic regression, using the same general technique in reporting their metrics. Their figures can be found in the appendices, as they are not critical to report in the primary technical report here.

Discussion and Conclusion

The dataset was relatively “messy” and involved a large amount of data cleaning in order to conduct the analysis and predictive models. There were many issues with infinite values, as well as NaN values, but were resolved in the data_dish_washer function. Further research could be done on model building to identify more optimal models in predicting the default status of firms. This research suggests that the best model for predicting defaulted firms in 2015 is the random forest model, with the most important features being `material_exp`, `d1_sales_mil_log` and `net_profit_margin`, suggesting the sales and profit play a primary role in determining if a firm will default.

Appendices

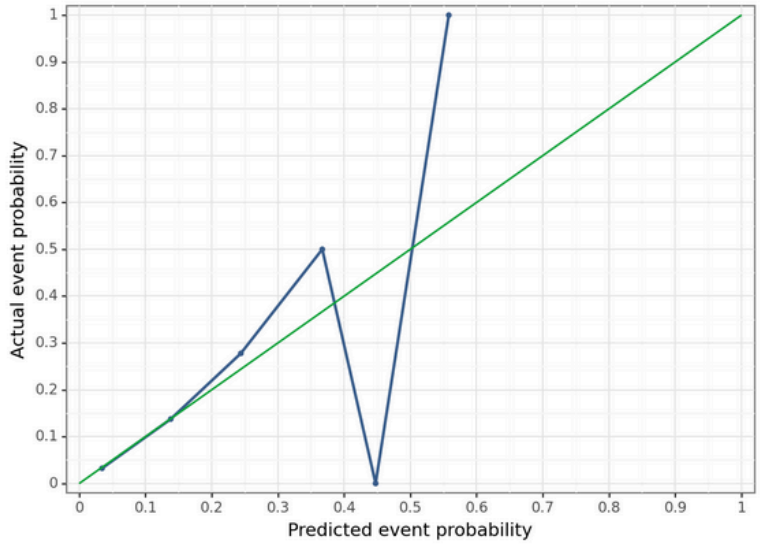


Figure 1: M2 Logistic Regression Calibration Curve

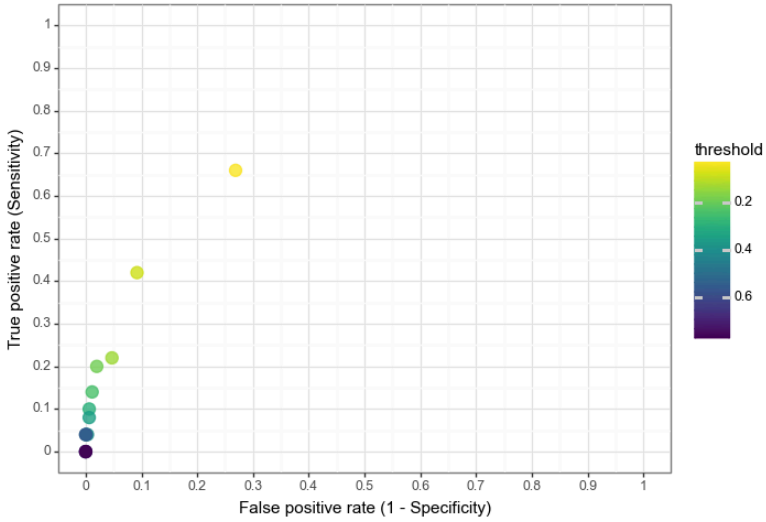


Figure 2: M2 Logistic Regression ROC Threshold Plot

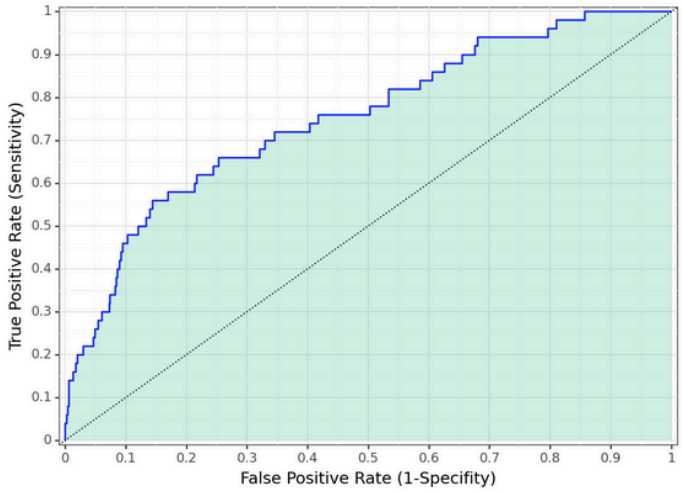


Figure 3: M2 Logistic Regression Continuous ROC Plot

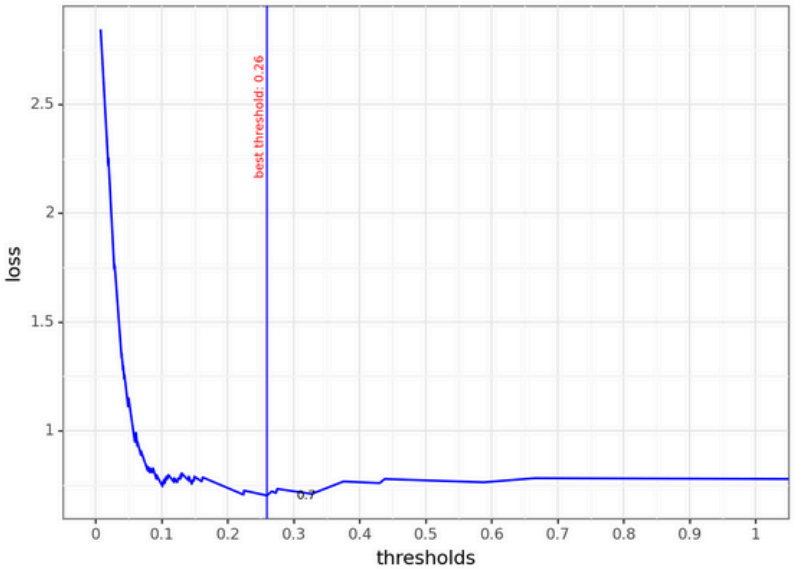


Figure 4: M2 Logistic Regression Loss/Optimal Threshold 5-fold CV Plot

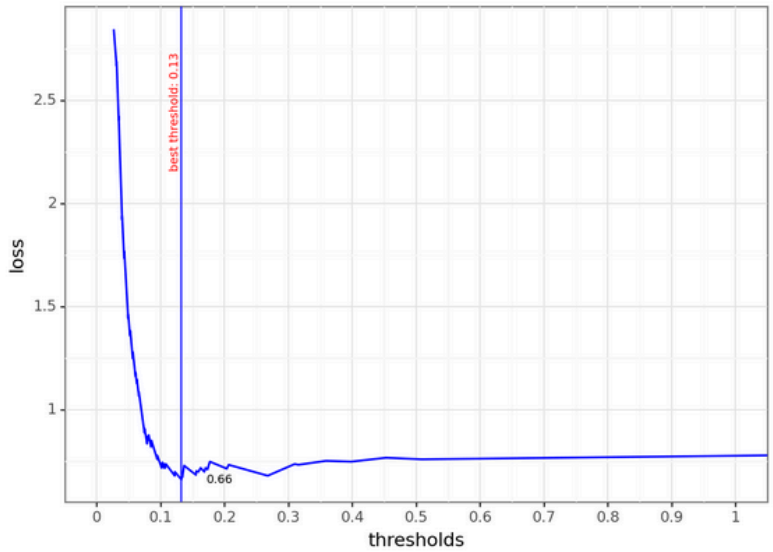


Figure 5: LASSO Loss/Optimal Threshold 5-fold CV Plot

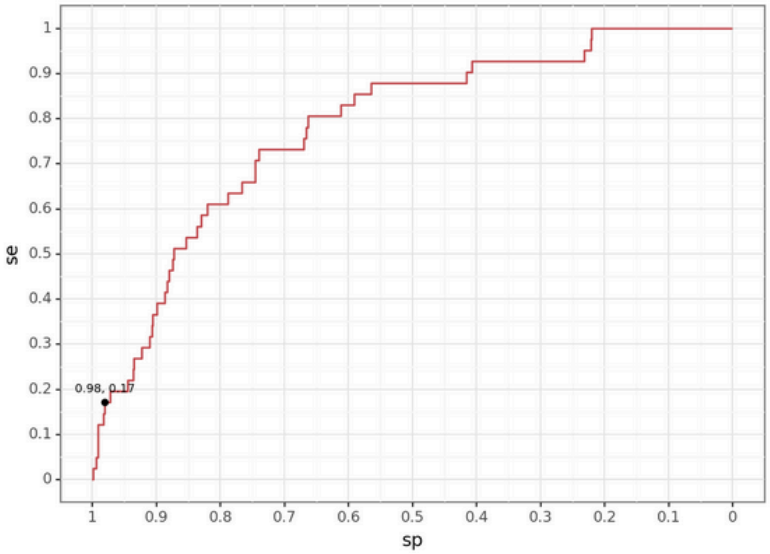


Figure 6: LASSO Optimal ROC Plot

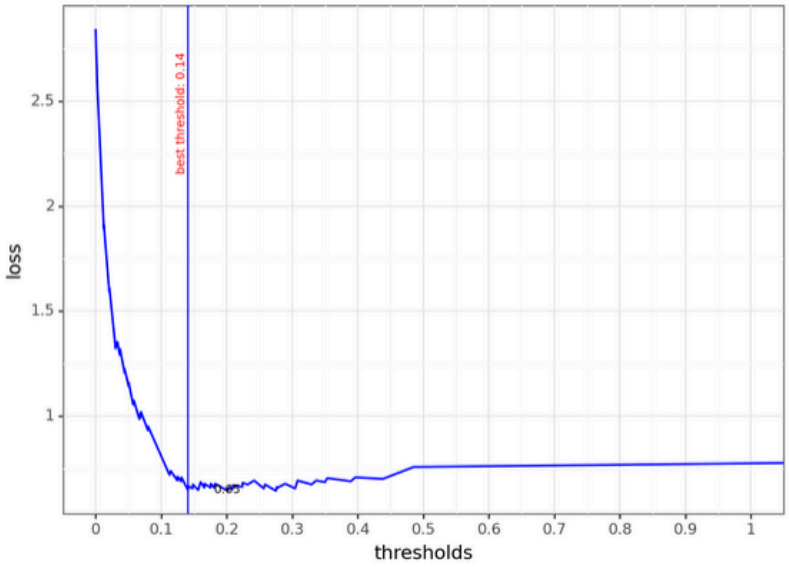


Figure 7: RF Loss/Optimal Threshold 5-fold CV Plot

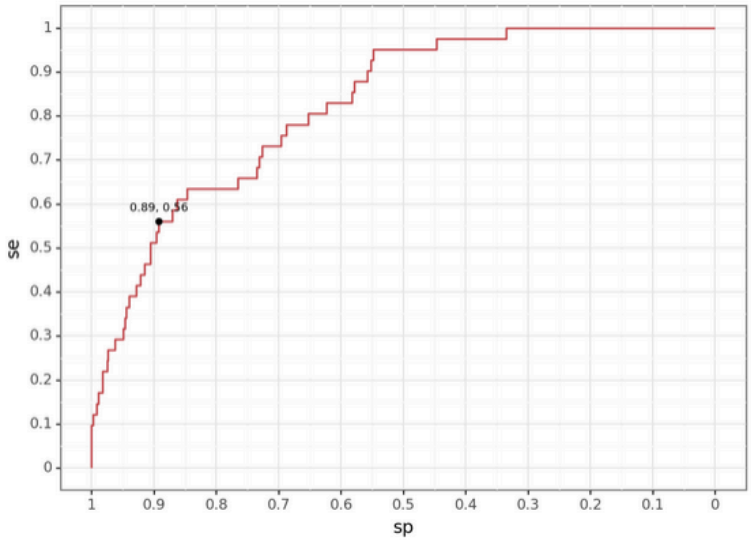


Figure 8: RF Optimal ROC Plot

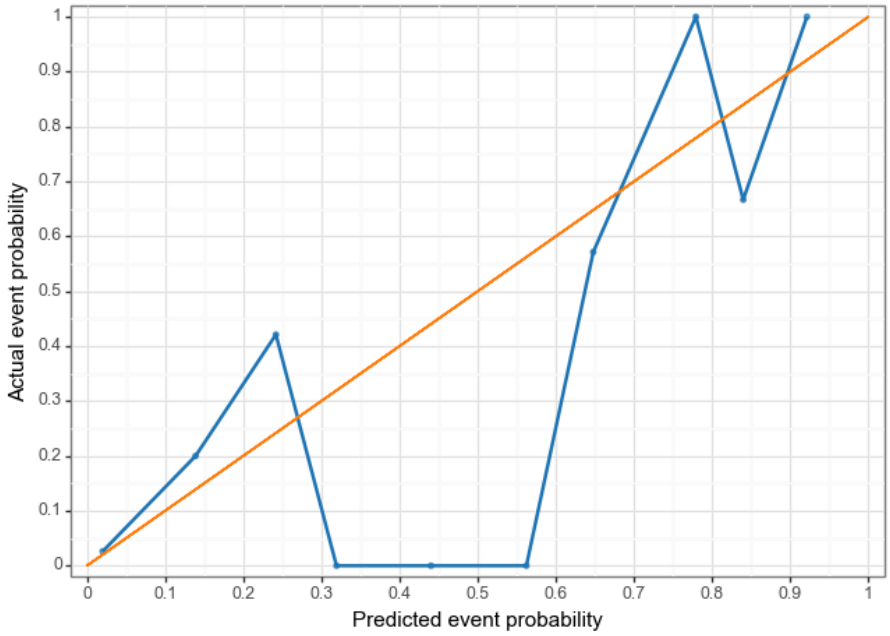


Figure 9: Gradient Boosting Calibration Curve

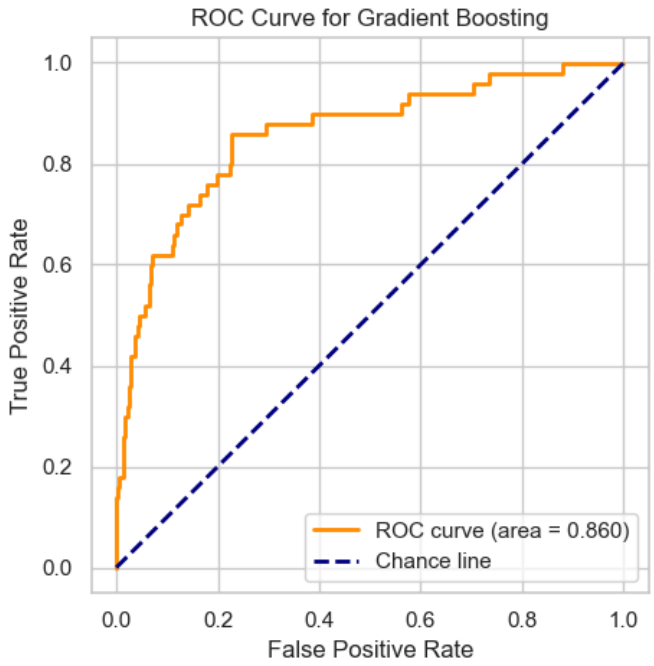


Figure 10: Gradient Boosting ROC Curve