Taiwanese Bankruptcy Prediction

Hubert Witkos
Han Chen
Soujan Niroula

University of Illinois Urbana-Champaign

May 1, 2025

Abstract

In this paper we trained four machine learning models, specifically Support Vector Machines, Random Forests, XGBoost, and penalized logistic regression in order to predict the bankruptcy of certain Taiwanese Companies. This is a classification problem where the outcome is binary; 0 representing no bankruptcy and 1 representing bankruptcy. The dataset we worked with in this paper is severely imbalanced with 96.77% of the companies being labeled as non-bankrupt. In order to remedy this problem of imbalance, we utilized an oversampling technique called synthetic minority oversampling technique (SMOTE). We chose the F2 score as our metric of choice to determine the predictive power of our models, and we received varied results from the different models. Surprisingly, logistic regression performed the best overall with XGBoost and Support Vector Machines falling not too far behind and Random Forest performing the worst out of the four models.

1 Introduction and Literature Review

The dataset, Taiwanese Bankruptcy Prediction, is from the UC Irvine Machine Learning repository website and was collected from the Taiwan Economic Journal for the years 1999 to 2009. The goal is to predict whether a given Taiwanese company is bankrupt based on the 95 other variables given. To do this we look to select the most important variables which explain the data the most and we train four different machine learning models to see which ones perform the best. We utilize the F2 score which is

$$F_\beta = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

with $\beta = 2$ in order to evaluate and compare the performance of the four machine learning models. Due to the severe imbalance of outcomes present in the dataset, accuracy in itself is not a sufficient metric for model performance evaluation. From the paper "Undersampling bankruptcy prediction: Taiwan bankruptcy data" by Haoming Wang and Xiangdong Liu, we adopted the F2 score due to its better assessment of model performance on imbalanced data and because it gives more weight to sensitivity which is important to us.[1] We reasoned that predicting bankrupt companies as non-bankrupt would have more perilous economic and financial consequences than predicting non-bankrupt companies as bankrupt and thus that is also why we chose the F2 score which places a higher importance on sensitivity.

There exist several different over and undersampling methods to remedy the problem of severe class imbalance in datasets such as ROSE, Tomek Links, Edited Nearest Neighbors, and SMOTE. We decided to utilize SMOTE due to its popularity and familiarity in the fields of statistics and data science. According to the article "SMOTE: Synthetic Minority Over-Sampling Technique", SMOTE, unlike other oversampling techniques, generates synthetic examples of the minority class by operating in the feature space rather than the data space.[2] In SMOTE, the minority class is oversampled by generating synthetic examples along the line segments joining all of the k minority class

[1] Haoming Wang and Xiangdong Liu, "Undersampling Bankruptcy Prediction: Taiwan Bankruptcy Data," *PLoS ONE* 16, no. 7 (July 1, 2021): e0254030, `https://doi.org/10.1371/journal.pone.0254030`.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research* 16 (2002): 321–357, `https://doi.org/10.1613/jair.953`.

nearest neighbors. In simpler terms, SMOTE utilizes the k nearest neighbors of each minority class sample to generate synthetic data of that class label which is very similar, but not identical to that original data point.

XGBoost is a model which we didn't specifically learn about in class but we have heard about and wanted to utilize in our research. XGBoost is a boosting model which according to the paper "XGBoost: A Scalable Tree Boosting System" by Tianqi Chen and Carlos Guestrin, derives most of its success from its scalability to all scenarios.[3] The model can be scaled to billions of examples in distributed or memory limited settings and it utilizes parallel and distributed computing to increase speed and efficiency. Gradient boosting utilizes a gradient descent optimization algorithm to minimize the loss function. The scalability and resulting speed with which XGBoost can perform makes it a very appealing and useful model to use instead of comparable techniques such as AdaBoost and Neural Networks.

2 Details of the Dataset, Summary Statistics, and Data Visualization

The dataset contains 95 predictor features/variables and one output variable, which is the Bankrupt column in our case which has 0 representing not-bankrupt and 1 representing bankrupt. The variables contain different information about a company's financial health such as realized sales gross margin, cash flow rate, operating gross margin, net value per share, total asset growth rate, etc. These financial ratios and indicators are supposed to inform us about the financial health and stability of a Taiwanese company.

We did some initial data preprocessing and exploratory data analysis in order to gain some insight into the data and to determine how to proceed with our task of classifying companies as bankrupt or not. The dataset consists of 6820 companies and contains no missing values, so we did not feel it necessary to drop any observations or perform data imputation. We noticed that some variables such as pre-tax net interest rate and after-tax net interest rate may be very correlated with each other, hence we decided to construct some correlation heat maps to see if we have multicollinearity present and

[3]Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 785–794, 2016, https://doi.org/10.1145/2939672.2939785.

if we can get rid of some of the correlated variables. We created several separate correlation heatmaps of smaller subsets of the variables since we couldn't create a heatmap with all 95 predictor variables and yet make it legible/readable. We also constructed a chart of 20 pairs of variables with the highest correlation, and this chart indicated to us that there are several problematic pairs with perfect positive and negative correlation. Perfect correlation is very problematic for prediction models because the model cannot distinguish the effects the correlated variables have on the output.

Var1	**Var2**	**Value**
Net.worth.Assets	Debt.ratio.	-1.0000000
Current.Liability.to.Liability	Current.Liabilities.Liability	1.0000000
Current.Liability.to.Equity	Current.Liabilities.Equity	1.0000000
Gross.Profit.to.Sales	Operating.Gross.Margin	1.0000000
Net.Value.Per.Share..C.	Net.Value.Per.Share..A.	0.9989373
Realized.Sales.Gross.Margin	Operating.Gross.Margin	0.9995183
Gross.Profit.to.Sales	Realized.Sales.Gross.Margin	0.9995182
Net.Value.Per.Share..A.	Net.Value.Per.Share..B.	0.9993420
Net.Value.Per.Share..C.	Net.Value.Per.Share..B.	0.9991786
Operating.profit.Paid.in.capital	Operating.Profit.Per.Share..Yuan...	0.9986962
Regular.Net.Profit.Growth.Rate	After.tax.Net.Profit.Growth.Rate	0.9961862
Continuous.interest.rate..after.tax.	Pre.tax.net.Interest.Rate	0.9936165
ROA.B..before.interest.and.depreciation.after.tax	ROA.C..before.interest.and.depreciation.before.interest	0.9868495
After.tax.net.Interest.Rate	Pre.tax.net.Interest.Rate	0.9863790
Continuous.interest.rate..after.tax.	After.tax.net.Interest.Rate	0.9844523
Liability.to.Equity	Current.Liabilities.Equity	0.9639084
Liability.to.Equity	Current.Liability.to.Equity	0.9639084
Net.profit.before.tax.Paid.in.capital	Per.Share.Net.profit.before.tax..Yuan...	0.9627229
Net.Income.to.Total.Assets	ROA.A..before.interest.and...after.tax	0.9615519
Net.profit.before.tax.Paid.in.capital	Persistent.EPS.in.the.Last.Four.Seasons	0.9594608

Table 1: The twenty highest correlated variable pairs with their correlation values.

By looking at the correlations between pairs of variables we were able to get rid of 16 highly correlated variables, where we determined high correlation as $|r| > 0.95$. By getting rid of these highly correlated variables we reduced our input dimension to 79.

After getting rid of the highly correlated variables, we examined the variance of every single predictor. We noticed that one predictor variable, "Net Income Flag", which is a categorical variable, had zero variance (its value was one for every observation). We came to the conclusion that such a variable with zero variance would not help our machine learning models gain any insight into the difference between bankrupt and non-bankrupt companies, and thus we decided to drop it, leaving us with 78 predictor variables. We also noticed that some variables had a very high variance, specifically we observed that 24 of the predictor variables had a variance $> 10^6$. Due to this

high variance present in some of the variables we decided that we will scale the data.

A unique feature of this dataset which we noticed is the severe imbalance present in the output variable with 96.77% of the outputs being 0 (non-bankrupt) and only 3.23% being 1 (bankrupt). Working with such data can lead to misleading results because one can simply create a model which always predicts 0 (non-bankrupt) and it will still get a very high accuracy although it evidently isn't a robust model.

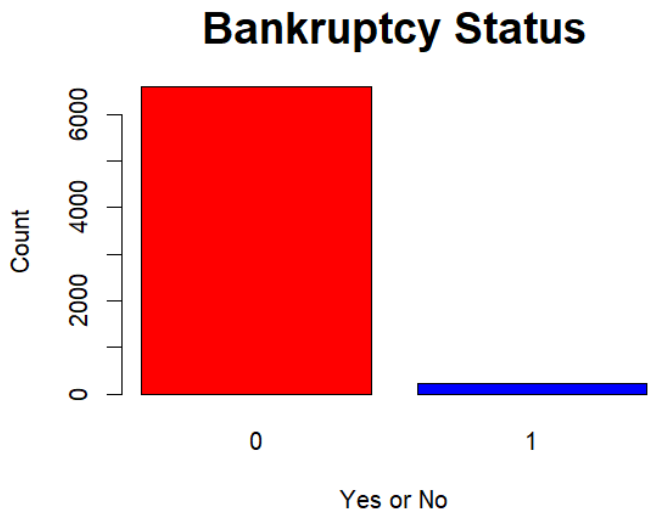

Figure 1: Barplot showing the frequency of the Bankrupt output. It is clear that the data is heavily imbalanced in favor of the non-bankrupt output.

To remedy this we utilized an oversampling technique called SMOTE. In R this function is called SMOTE and it comes from the smotefamily library. This function identifies the minority class and then finds the k nearest neighbors for each minority instance using Euclidean distance. Then, for each minority instance, the function generates synthetic data by interpolating between the instance and its neighbors. We applied this SMOTE function to our training data only, choosing $k = 5$ neighbors and 30 synthetic observations per minority instance. Oversampling is only performed on the training set since altering the class distribution of the test set can lead to data leakage and inaccurate/overoptimistic model performance. After utilizing SMOTE,

zeros (bankrupt) instances make up 51.5% of the training dataset. After we completed the oversampling we scaled all of the data, training and testing sets, in order to standardize the variances for all of the predictors.

As was stated earlier, accuracy is not a good metric for models dealing with imbalanced data, hence we chose the F2 score as our metric of choice for model evaluation. The F2 score also seems like a better choice for our model evaluation metric because it places a greater emphasis on recall, and we came to the conclusion that labeling a company, which in actuality is bankrupt, as financially sound and stable (non-bankrupt) can have more significant financial and economic repercussions than the inverse error.

3 Classification Tasks

3.1 Logistic Regression

Logistic regression was chosen as it works well with classification and we are able to add penalties (Lasso and Ridge) to the model to help improve performance. To train the logistic regression model, we used 10 fold cross validation with elastic net penalties using the cv.glmnet function in R to select the best model. Penalized logistic regression was chosen because it has the ability of variable selection while being good with multicollinearity. With that, alpha of 0.5 was chosen as our default and deviance was chosen as a measure as we are doing a classification problem. For the hyperparameter, lambda.1se was chosen as it gives a precise yet simpler model than lambda.min. On choosing the cutoff value, we wanted to capture most of the companies which are bankrupt (sensitivity) but without compromising accuracy too much, so we tested a few cutoff values from 0.3 to 0.5. Based on those results, 0.5 seemed to be the best balance between capturing as many true positives as possible but without lowering the accuracy too much. The penalized logistic regression model trained using cv.glmnet resulted in a F2 score of 0.567, sensitivity is 0.73585, accuracy of 0.9216.

Logistic Regression	Value
Sensitivity	0.736
Accuracy	0.922
F2 Score	0.567

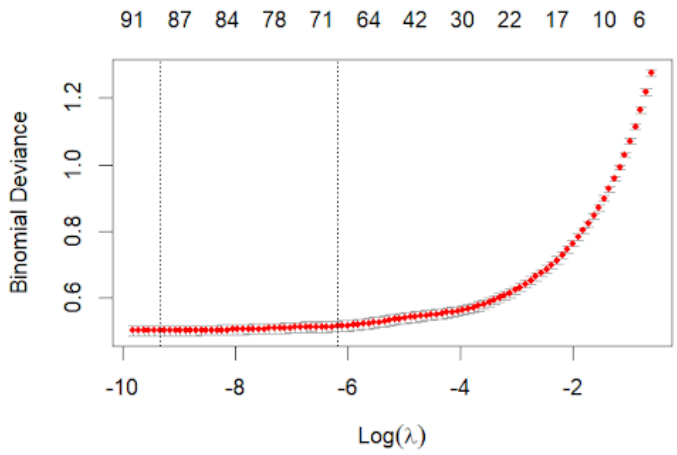

Figure 2: Log(lambda) plotted against the Deviance. Lambda.1se was selected as it gives a precise but simpler model than lambda.min.

3.2 Support Vector Machines (SVM)

SVM was picked for its powerful ability to work with binary classification tasks and flexibility for nonlinear decision boundaries. For fitting the SVM models we tried different kernels (polynomial-degrees, radial, linear) with a vector of different costs and then plotted them on a graph with cost on the x-axis and sensitivity on the y axis. We found that an SVM with a radial kernel and cost of 0.07 performed the best out of all of the SVMs. When the kernel is set to linear, it does not perform well because the data is high dimensional and most likely not linearly separable. When the kernel is set to polynomial, the sensitivity is 0.307 (accuracy is .857 and F2 score is 0.178). However, when the kernel is set to radial the model performed better with a sensitivity of 0.67925 (accuracy is 0.9216, F2 score is 0.542).

SVM	Value
Sensitivity	0.679
Accuracy	0.922
F2 Score	0.542

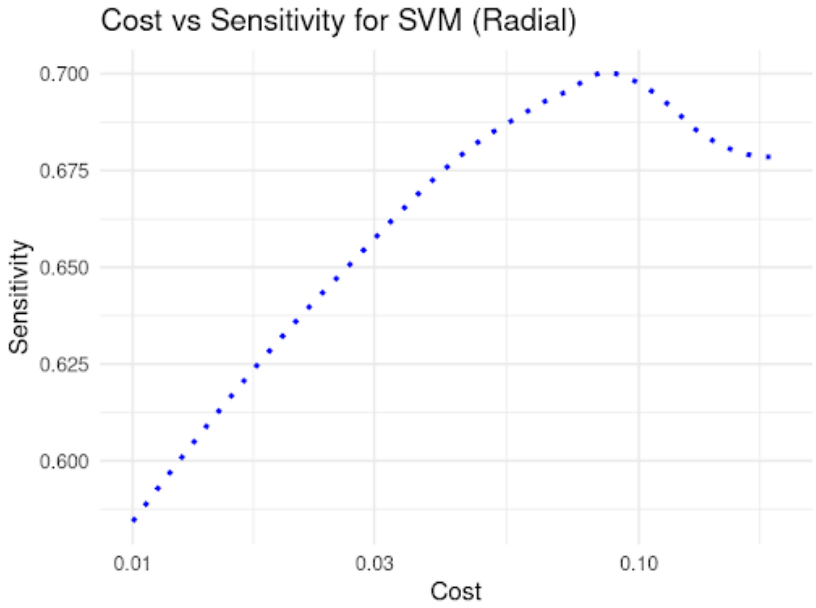

Figure 3: Cost plotted against sensitivity for SVM with radial kernel. Sensitivity is maximized at cost = .07.

3.3 XGBoost

XGBoost was chosen as it is a very powerful classifier which requires less runtime and computational power than comparable models. We configured the model with the objective set to "binary:logistic", nrounds set to 100 to prevent overfitting, and used AUC as the evaluation metric. To optimize performance, we selected a cutoff value that maximized sensitivity. A threshold of 0.1 yielded a sensitivity of 0.62264, accuracy of 0.9523, and an F2 score of 0.569. In comparison, the default cutoff of 0.5 resulted in a sensitivity of 0.45283, accuracy of 0.9641, and a lower F2 score of 0.469. Although accuracy slightly decreased at the lower cutoff, the improvement in sensitivity allowed us to identify significantly more potential bankruptcies. Finally, we extracted the top 10 most important features identified by the model, with Persistent.EPS.in.the.Last.Four.Seasons emerging as the most influential predictor.

XGBoost	Value
Sensitivity	0.623
Accuracy	0.952
F2 Score	0.569

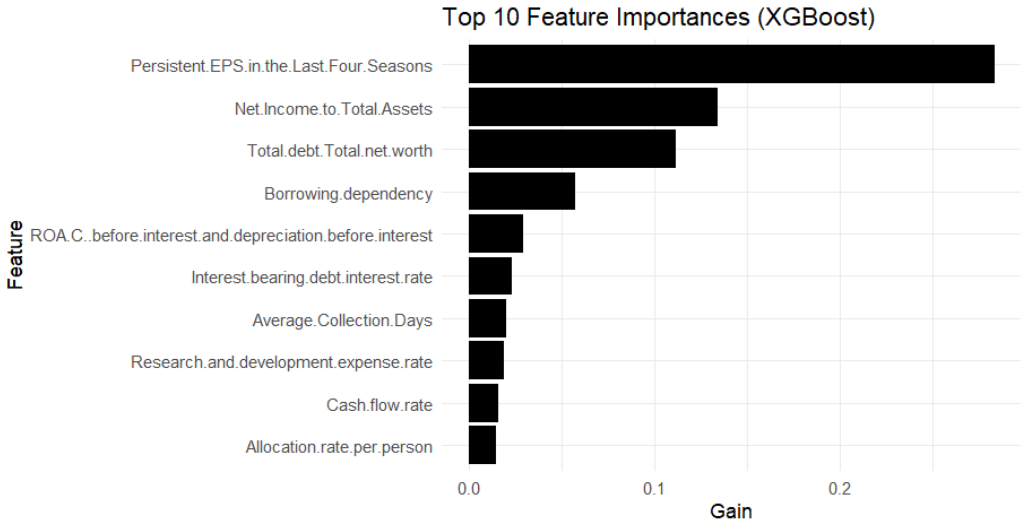

Figure 4: Top Ten Variables by Mean Decrease in Accuracy

3.4 Random Forest

We utilized the random forest ensemble method due to its non-parametric nature which allows us to avoid making any assumptions on the predictor function. When we applied the random forest classification technique we realized the model has trouble predicting bankrupt instances. Random forest predicted almost all of the outputs as 0 (non-bankrupt) which explains why in our best model we achieved an impressive accuracy of 96%, but we were not able to get the sensitivity and F2 score much higher than 0.5. This suggests that the model did not perform very well and was not able to properly predict the bankruptcy (true positive values). In our case this is especially discouraging for we need to have a high sensitivity and a model which is able to capture the bankruptcy instances accurately.

The mean decrease in accuracy plot in the appendix (Figure 7) highlights the top 10 variables with the strongest influence in the model's predictive power and these should be prioritized for further interpretation.

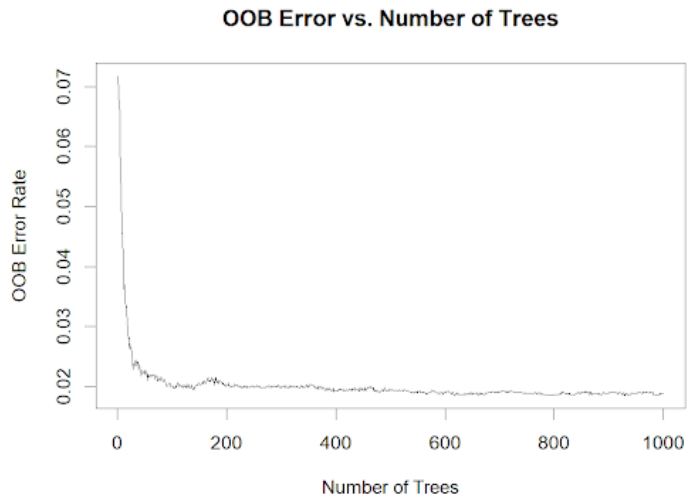

Figure 5: OOB Error Against Number of Trees. Analyzing the OOB error rate plot against number of trees, we can see that OOB error stabilizes after approximately 200-300 trees. Increasing further beyond this point does not contribute much to reducing the error.

4 Conclusion

Model	Sensitivity	Accuracy	F2 Score
Logistic	0.736	0.922	0.567
SVM	0.679	0.922	0.542
XGBoost	0.623	0.952	0.569
RandomForest	0.509	0.960	0.513

Table 2: Summary of the model results.

After preprocessing the data and applying SMOTE, our goal was to maximize the F2 score and sensitivity while also maintaining a reasonable accuracy. Among the models compared, logistic regression performed best in

capturing sensitivity, followed by SVM, XGBoost, and Random Forest. XGBoost achieved the best F2 score with logistic regression falling not far behind it. Overall, we recommend logistic regression considering its relatively high sensitivity and F2 score. In the future, we could improve our results by enhancing data preprocessing, exploring different minority over/undersampling methods such as edited nearest neighbors, and training different statistical learning models such as Naive Bayes and Quadratic Discriminant Analysis.

References

[1] Blagus, Rok, and Lara Lusa. SMOTE for High-dimensional Class-imbalanced Data. *BMC Bioinformatics* 14, no. 1 (March 22, 2013). https://doi.org/10.1186/1471-2105-14-106.

[2] Haoming Wang and Xiangdong Liu. Undersampling Bankruptcy Prediction: Taiwan Bankruptcy Data. *PLoS ONE* 16, no. 7 (July 1, 2021): e0254030. https://doi.org/10.1371/journal.pone.0254030.

[3] Hung V. Pham, Tuan Chu Dao, Tuan M. Le, Hieu M. Tran, Huong T.K. Tran, Khanh N. Yen, Son V. T. Comprehensive Evaluation of Bankruptcy Prediction in Taiwanese Firms Using Multiple Machine Learning Models. *IJTech - International Journal of Technology*, n.d. https://ijtech.eng.ui.ac.id/article/view/7227.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. https://doi.org/10.1613/jair.953.

[5] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, 2016. https://doi.org/10.1145/2939672.2939785.

[6] UCI Machine Learning Repository. Taiwan Bankruptcy Prediction Dataset. https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction.

Note: ChatGPT was used to refine the writing, models, and latex encoding.

Appendix

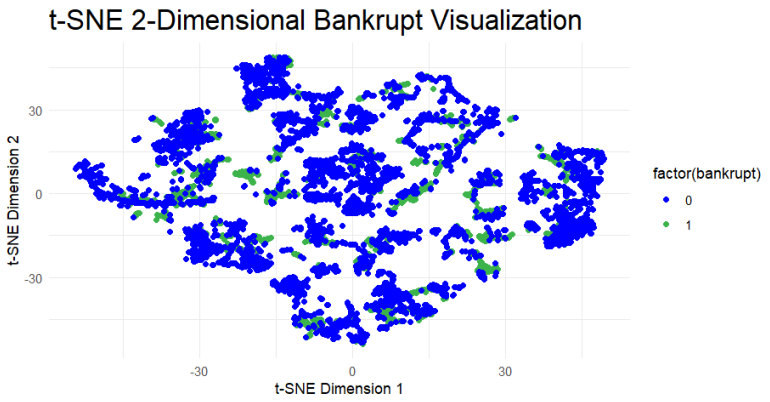

Figure 6: TSNE

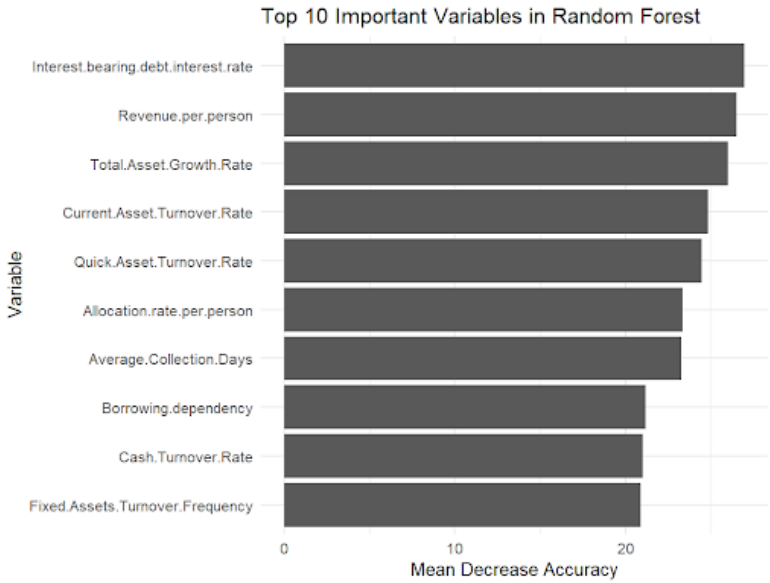

Figure 7: Top ten variables by Mean Decrease in Accuracy in the Random Forest model.

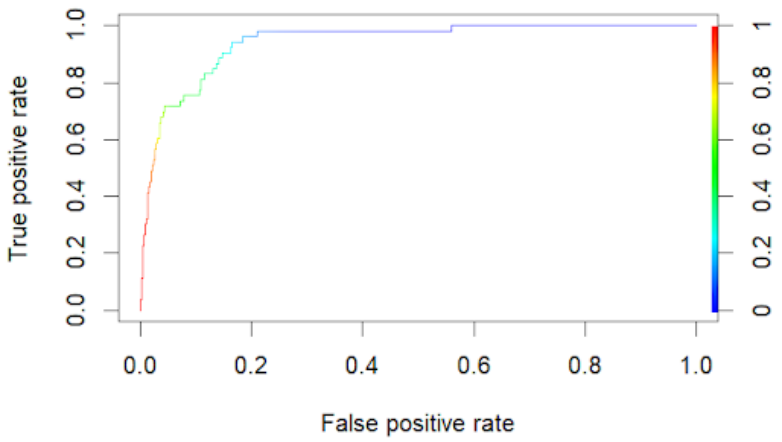

Figure 8: Logistic ROC Curve