Credit Lending Decisions

2021 University of Georgia Data Science Competition

Cody Dailey[[1]](#footnote-20)

Ishaan Dave1

Yang Ge1

Vipul Shinde[[2]](#footnote-21)

15 April 2021

Table of Contents

For more details, please see appendix (add exactly where – figure XXX).

# Background

# 1 Competition Prompt

We are expected to work as a team to develop ***credit decision models*** using the simulated datasets for Bank XYZ, to interpret the results, document findings, and present to a panel of judges.

Specifically, we are asked to  
1. Conduct an exploratory data analysis (Section 2)  
2. Develop and fit a logistic regression model (Section 4)  
3. Develop an additional machine learning (ML) model: random forest, gradient boosting, or feedforward neural network (Section 5)  
4. Compare the logistic regression model with ML model, select one for credit approval, and describe the reason for our selection (Section 6)  
5. Answer the following questions: (Section 7)  
- Do customers who already have an account with the financial institution receive any favorable treatment in our model?  
- Suppose a credit card application is rejected using our model and the applicant asks us to provide an explanation as to why it was rejected. How would we explain the results to the customer?

# Bank XYZ and Credit Bureaus’ Credit Data

# 2 Exploratory Data Analysis

## 2.1 Methods

Data were summarized using means and standard deviations or frequency and percentages as appropriate. For discrete count data, data were summarized using medians and interquartile range. Additional exploratory data analyses were conducted to examine normality of continuous variables and cross-tabulations of categorical variables between outcome levels. Bivariate relationships were examined using correlations (point-biserial between categorical and continuous, and Pearson correlation between continuous variables) and plotted in a correlation matrix (darker colors represent stronger correlations).

# 3 Results

Table 3.1 shows data summaries for each variable in the training dataset stratified by the Default indicator.

Table 3.1: Descriptive Statistics of Training Data

| Variable | Levels | Defaulted | Did Not Default | Statistic | Parameter | p |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | n = 1586 (7.9) | n = 18414 (92.1) |  |  |  |
| tot\_credit\_debt |  | 91915.57 (29920.36) | 94791.79 (22901.39) | -3.74 | 1,748.61 | 0.000 |
| avg\_card\_debt |  | 17880.43 (21195.06) | 13761.61 (7363.26) | 7.70 | 1,618.11 | 0.000 |
| credit\_age |  | 275.16 (61.99) | 298.55 (61.34) | -14.43 | 1,862.48 | 0.000 |
| credit\_good\_age |  | 140.54 (33.56) | 150.57 (33.94) | -11.41 | 1,875.32 | 0.000 |
| card\_age |  | 247.71 (58.6) | 269.76 (59.11) | -14.37 | 1,873.68 | 0.000 |
| non\_mtg\_acc\_past\_due\_12\_months\_num |  | 0 [0, 2] | 0 [0, 0] | 19,325,868.00 |  | 0.000 |
| non\_mtg\_acc\_past\_due\_6\_months\_num |  | 0 [0, 0] | 0 [0, 0] | 17,862,939.00 |  | 0.000 |
| mortgages\_past\_due\_6\_months\_num |  | 0 [0, 1] | 0 [0, 0] | 18,383,330.00 |  | 0.000 |
| credit\_past\_due\_amount |  | 3060.92 (5777.48) | 94.01 (1048.88) | 20.42 | 1,594.01 | 0.000 |
| inq\_12\_month\_num |  | 2 [1, 4] | 1 [0, 3] | 17,234,540.00 |  | 0.000 |
| card\_inq\_24\_month\_num |  | 4 [2, 6] | 3 [1, 5] | 16,837,121.00 |  | 0.000 |
| card\_open\_36\_month\_num |  | 0 [0, 0] | 0 [0, 0] | 15,023,057.00 |  | 0.002 |
| auto\_open\_.36\_month\_num |  | 0 [0, 0] | 0 [0, 0] | 14,743,462.00 |  | 0.288 |
| uti\_card |  | 0.56 (0.11) | 0.5 (0.11) | 21.49 | 1,853.57 | 0.000 |
| uti\_50plus\_pct |  | 0.56 (0.11) | 0.51 (0.11) | 16.99 | 1,855.96 | 0.000 |
| uti\_max\_credit\_line |  | 0.55 (0.11) | 0.5 (0.11) | 16.51 | 1,858.46 | 0.000 |
| uti\_card\_50plus\_pct |  | 0.54 (0.12) | 0.48 (0.12) | 17.48 | 1,641.42 | 0.000 |
|  | Missing | 1881 (10.2) | 174 (11) |  |  |  |
| ind\_acc\_XYZ | Account w/ XYZ | 328 (20.7) | 4843 (26.3) | 23.76 | 1.00 | 0.000 |
|  | No Account | 1258 (79.3) | 13571 (73.7) |  |  |  |
| rep\_income |  | 74522.99 (16775.13) | 75583.34 (16323.78) | -2.32 | 1,701.36 | 0.020 |
|  | Missing | 1441 (7.8) | 129 (8.1) |  |  |  |
| States | AL | 229 (14.4) | 2664 (14.5) | 4.06 | 6.00 | 0.669 |
|  | FL | 219 (13.8) | 2638 (14.3) |  |  |  |
|  | GA | 209 (13.2) | 2648 (14.4) |  |  |  |
|  | LA | 228 (14.4) | 2621 (14.2) |  |  |  |
|  | MS | 241 (15.2) | 2586 (14) |  |  |  |
|  | NC | 224 (14.1) | 2674 (14.5) |  |  |  |
|  | SC | 236 (14.9) | 2583 (14) |  |  |  |

Figure Correlation Matrix

# Modeling and Predicting Probabilities of Account Default

To predict whether or not individuals will default, 2 classes of models were fit: the first was an adaptive LASSO logistic regression model, and the second was a random forest machine learning algorithm for classification. Each model class was fit numerous times on the training data in an exploration of model-specific hyperparameters. All models were then used to classify the observations within the validation dataset. Within each model class, predictive performances were compared by examining differences in area under the curve (AUC) from the receiver operating characteristic curves (ROC). Of the top performing models (AUC in 2.5 percentile), probability thresholds were set according to the financial impact of correct (or incorrect) classification via assigning costs to correctly or incorrectly classifying an individual (Equation (3.1)).

where: TP, FP, TN, and FN refer to the frequencies of true positives, false positives, true negatives, and false negatives, respectively; and are the average monthly credit card debt for an individuals who did not default or who defaulted, respectively; and are the profit margins for a credit lendor with respect to purchases made (merchant) and interest from late payments (interest); and is the average number of delinquencies.[[3]](#footnote-28)

Finally, with a single threshold chosen for each model, the most accurate model (as determined via the distance to corner metric, Equation (3.2)) was chosen from each model class.

The results presented in this section pertain to model fits on the validation data.

# 4 Logistic Regression

Our main outcome of interest is whether or not an individual defaults when applying for a credit card. Traditionally, associations between predictors and outcome are tested individually through univariate regression models, then a subset of those covariates are selected for a multivariate regression analysis if univariate analyses yielded significant results. Given the large number of possible predictors, we suspected severe multicollinarity between some pairs of variables, and thus, overfitting of our model. To alleviate this, we “penalized” regression coefficients to select a parsimonious set of variables that leads to similar prediction results.

## 4.1 Methods

In a normal logistic regression with a binary outcome, our model is defined by logit(E(Y│X)=β^T (X), where Y is the binary outcome (whether or not an individual defaulted), X = (1,X\_1,…,X\_p) is the vector of covariate values, and β = (β\_0,β\_1,…,β\_p ) is the vector of regression parameters. In a penalized regression, estimation is generally in the form of an optimization problem that seeks to maximize the function in the form of L(β)-λ\*pen(β), where pen(β) is the “penalty” and λ is the “tuning parameter.” The penalty refers to a constraint surrounding the betas and the tuning parameter corresponds to the amount of “shrinkage” applied to the coefficient. In this analysis, an adaptive LASSO (least absolute shrinkage and selection operator) with logistic regression was employed to estimate coefficients and select the most important predictors. Adaptive LASSO is conducted in 2 parts – the first is a “ridge” regression is conducted that uses a penalty proportional to the sum of squares of the regression coefficients and efficiently handles multicollinearity (Equation (4.1)).

The second step is the LASSO step that imposes a constraint on the sum of the absolute value of the regression coefficients and uses data-dependent weights (the inverse of the ridge regression coefficients) to penalize regression coefficients accordingly (Equation (4.2)). Therefore strong predictors’ coefficients are shrunk less than coefficients of weak predictors.

Continuous predictors were standardized to have mean 0 and standard deviation 1 before penalized regression was conducted. The main focus of penalized regression is predictive performance rather than uncertainty around the coefficient itself. Confidence intervals for coefficients and predicted probabilities from penalized methods are problematic because coefficients are already biased toward the null because of the inherent penalty term and not presented with regression coefficients. In the table below, we show the standardized coefficient estimates for both ordinary logistic regression and adaptive lasso as well as the shrinkage associated with each coefficient.

The main purpose of LASSO is predictive performance rather than coefficient estimates. We used a training dataset to best inform the penalty and tuning parameters to be used in the LASSO. A validation test set was used to best determine the parameters in our model and our final model’s predictive performance was determined by running our model on a testing dataset and record the resulting AUC associated with classifying an individual as one who will or will not default. Moreover, a confusion matrix of actual vs. predicted counts of those who did and did not default was constructed to calculate predictive accuracy, classifications were performed at multiple predicted probability thresholds to determine that which resulted in highest predictive accuracy. To quantify predictive performance of our models, a monetary value was associated with both correctly and incorrectly classifying an individual.

In addition to an adaptive LASSO logistic regression model, a random forest machine learning algorithm was run to compare predictive accuracy

## 4.2 Results

Figure Adaptive LASSO - Training –> ROC & Cost on training and validation datasets

Table - Coefficients / ORs / Probabilities - Confusion Matrices

Figure 4.1

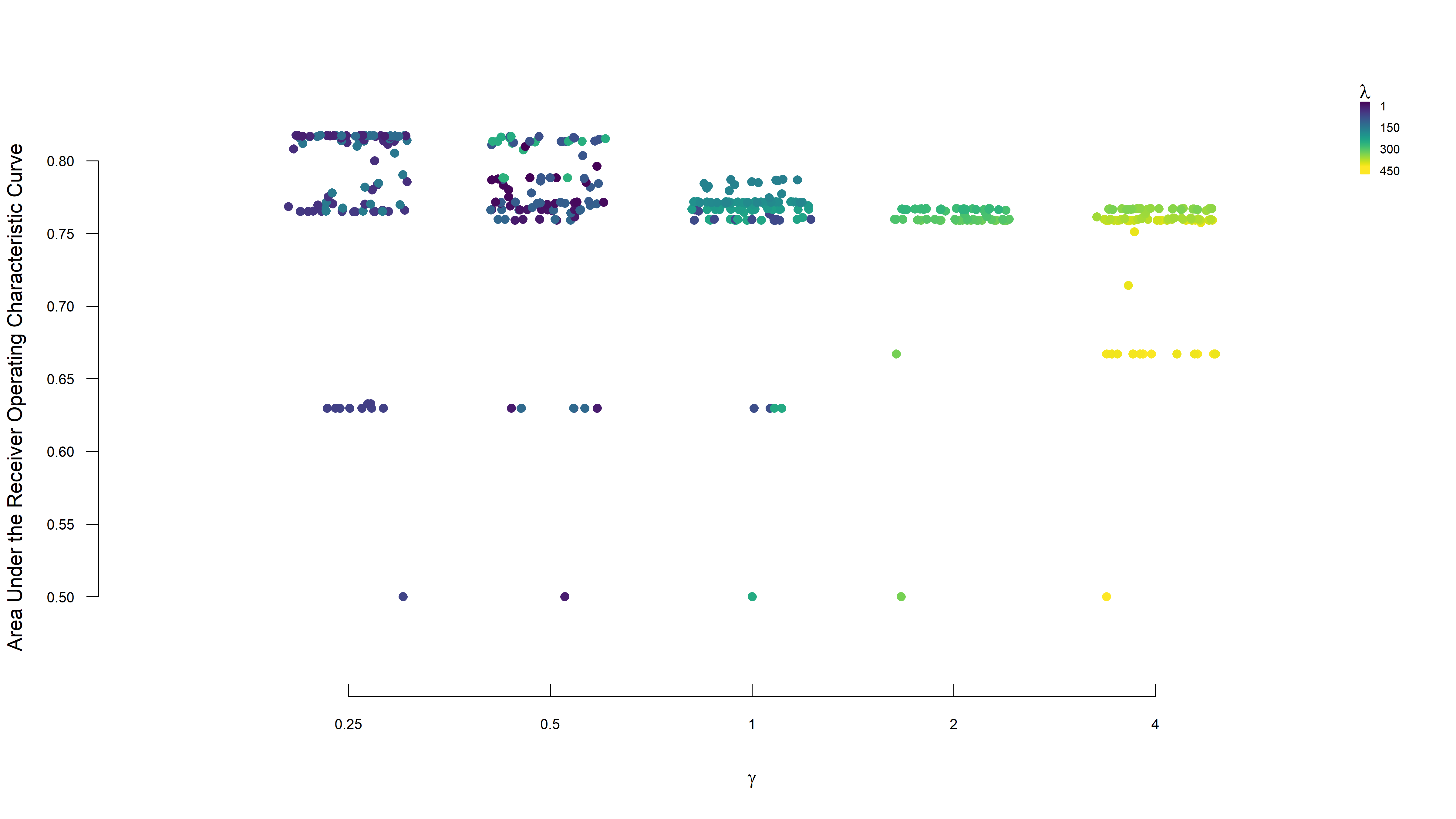


Figure 4.1: Adaptive LASSO Logistic Regression Hyperparameters and Areas Under the Receiver Operating Characteristic Curve

Figure 4.2

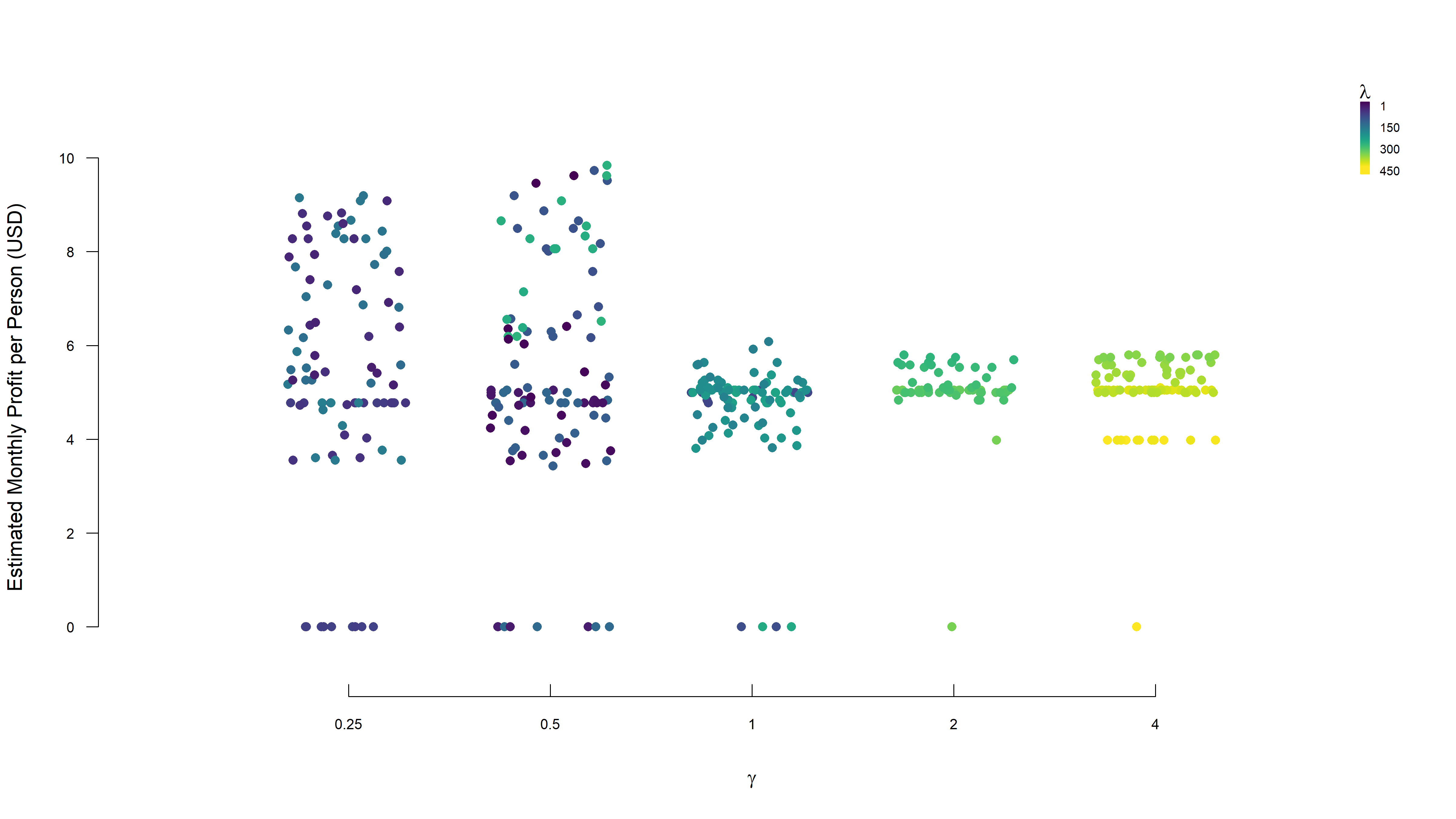


Figure 4.2: Adaptive LASSO Logistic Regression Hyperparameters and Estimated Monthly Profit per Applicant

# 5 Random Forest

## 5.1 Methods

## 5.2 Results

Figure - Training –> ROC & Cost on training and validation datasets - Variable importance - Proximity? - “Best” tree?

Table - Confusion matrices

Figure 5.1

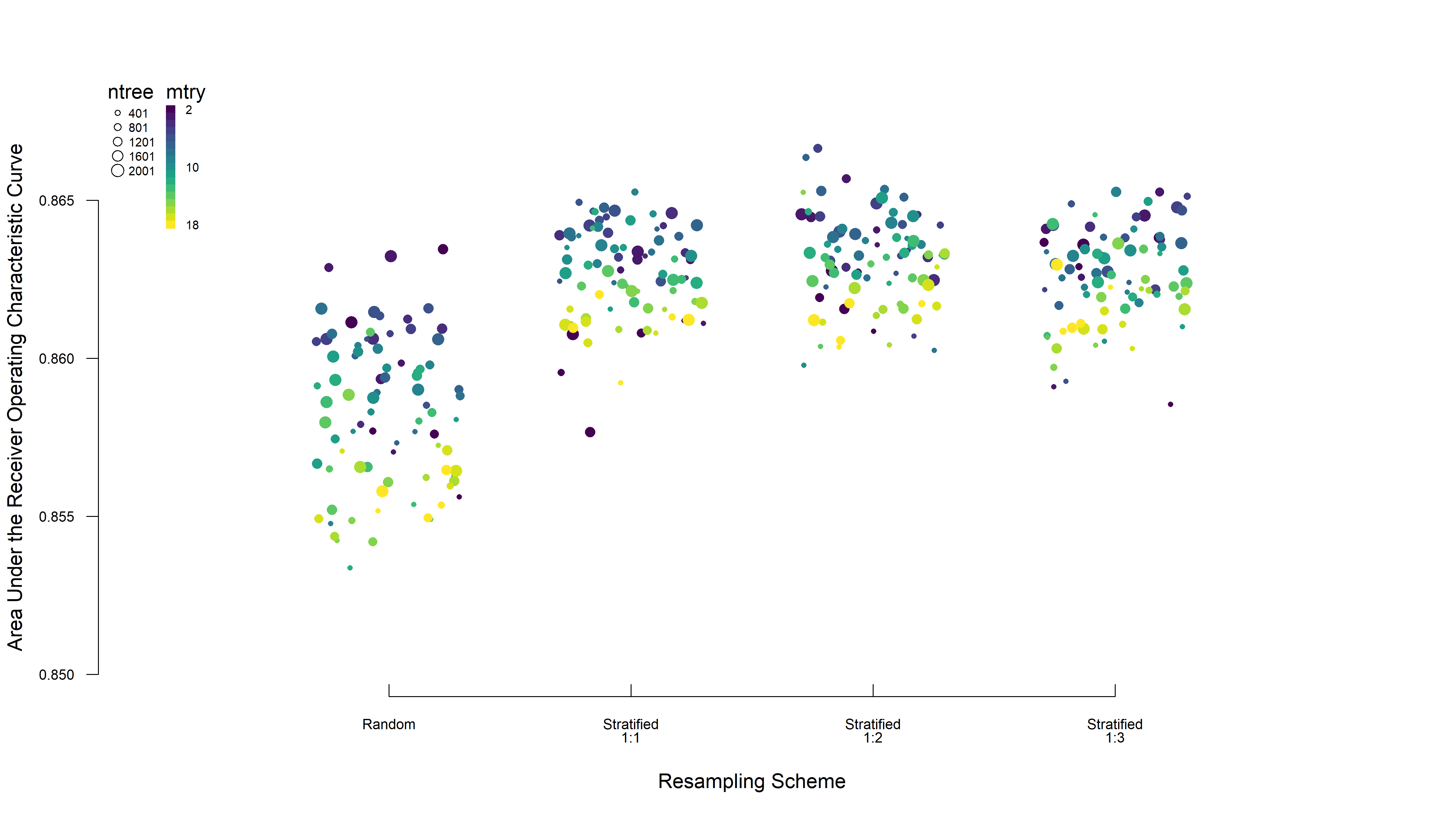


Figure 5.1: Random Forest Hyperparameters and Areas Under the Receiver Operating Characteristic Curve

Figure 5.2

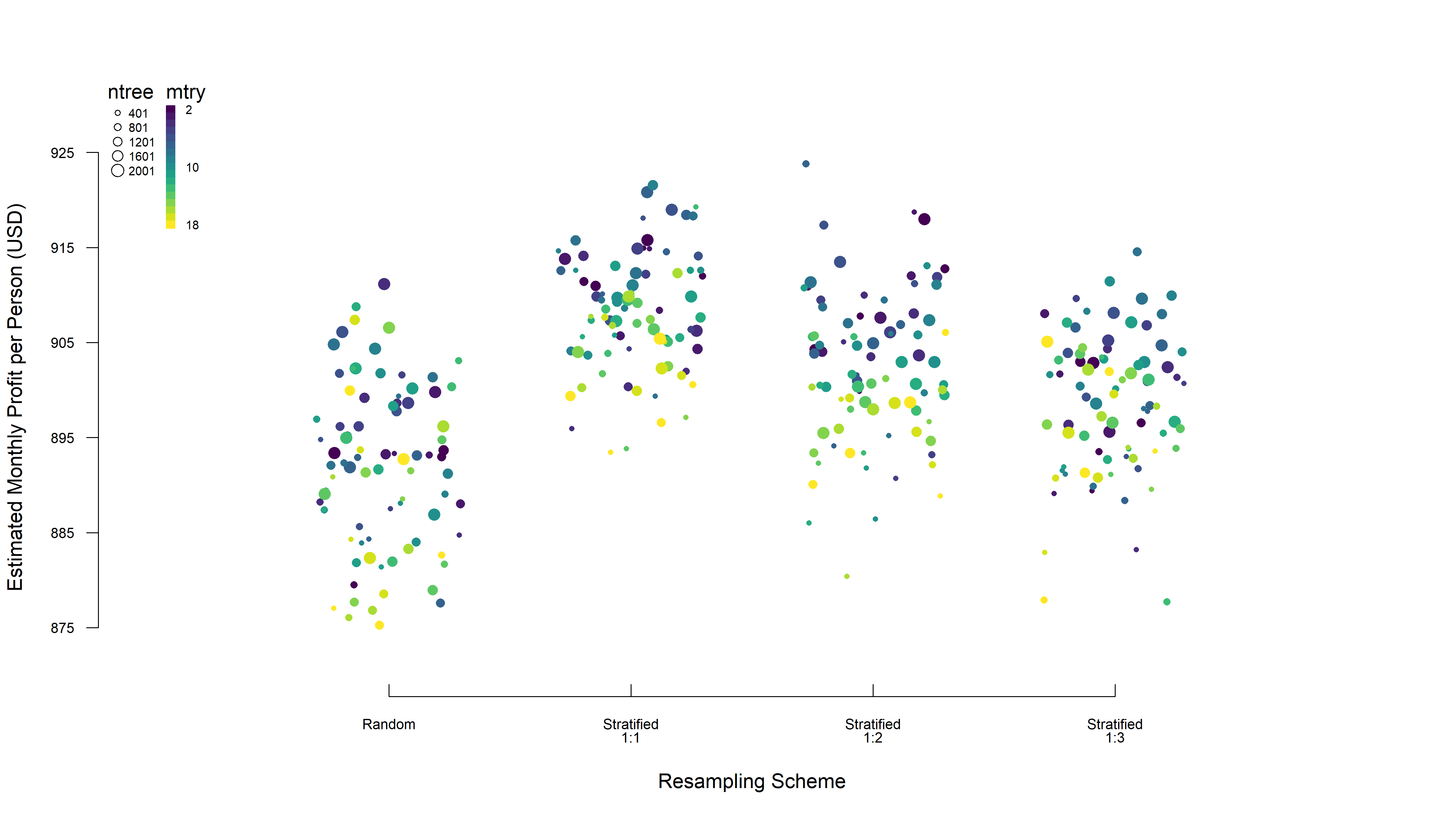


Figure 5.2: Random Forest Hyperparameters and Estimated Monthly Profit per Applicant

Figure 5.3

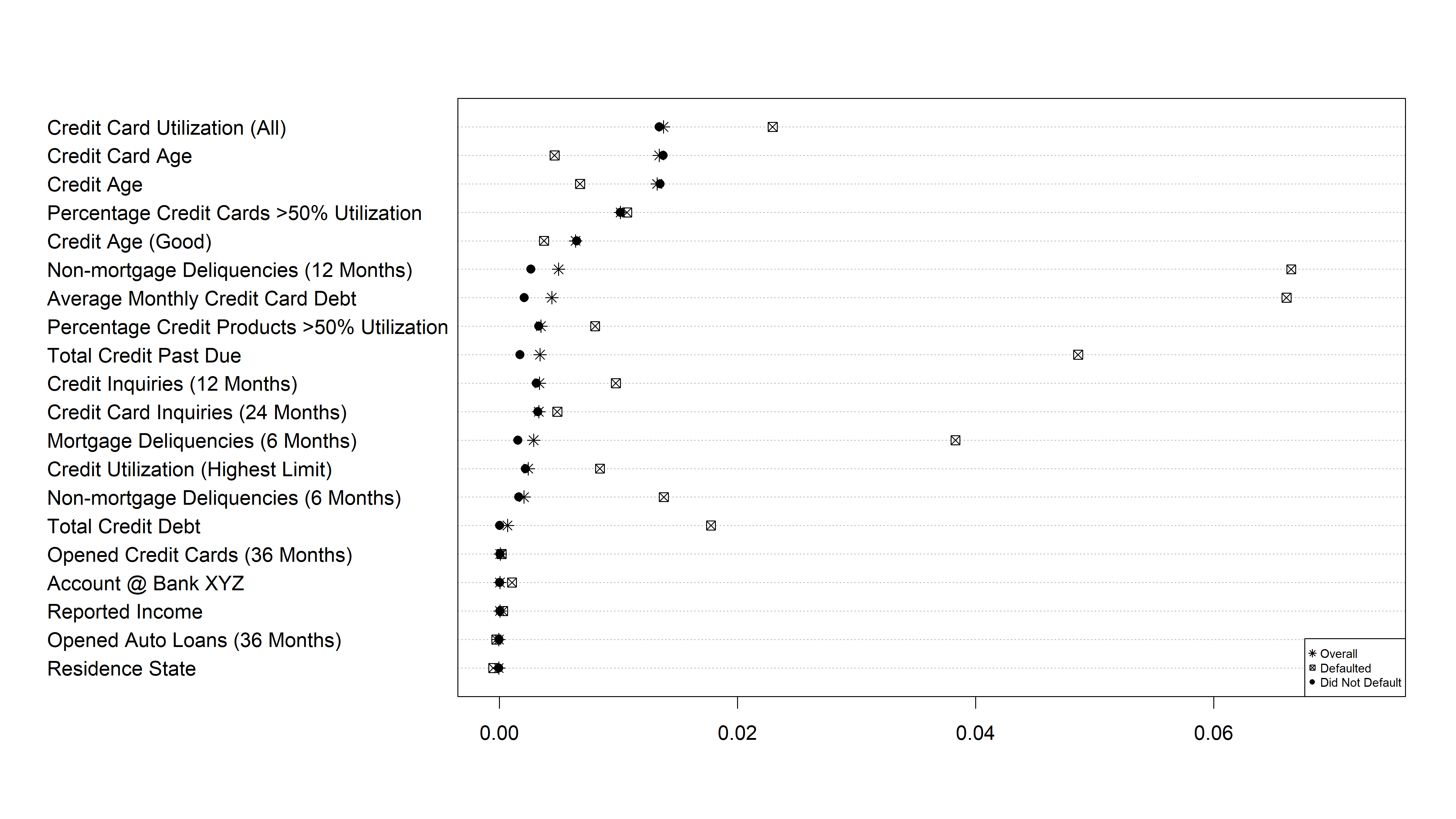


Figure 5.3: Variable Importance in Random Forest Model

# 6 Model Comparisons

Table 6.1

Table 6.1: Confusion Matrix for Adaptive LASSO Logistic Regression Model Predictions on the Test Data

|  | | **Truth** | |  |
| --- | --- | --- | --- | --- |
|  |  | *Defaulted* | *Did Not Default* |  |
| **Prediction** | *Defaulted* | 398 | 4,150 | 4,548 |
| *Did Not Default* | 3 | 449 | 452 |
|  |  | 401 | 4,599 | 5,000 |

Table 6.2

Table 6.2: Confusion Matrix for Random Forest Model Predictions on the Test Data

|  | | **Truth** | |  |
| --- | --- | --- | --- | --- |
|  |  | *Defaulted* | *Did Not Default* |  |
| **Prediction** | *Defaulted* | 390 | 3,060 | 3,450 |
| *Did Not Default* | 11 | 1,539 | 1,550 |
|  |  | 401 | 4,599 | 5,000 |

Table 6.3

Table 6.3: Various Performance Metrics Comparing Predictions on the Test Data from Adaptive LASSO Logistic Regression and Random Forest Models

|  |  |  |
| --- | --- | --- |
| *Metric* | **LASSO** | **RF** |
| Accuracy | 16.94% | 38.58% |
| Error | 83.06% | 61.42% |
| Profit | $31205.95 | $97091.97 |
| Profit.mp | $6.24 | $19.42 |
| PPV | 0.088 | 0.113 |
| NPV | 0.993 | 0.993 |
| Sensitivity | 0.993 | 0.973 |
| Specificity | 0.098 | 0.335 |
| Youden | 0.090 | 0.307 |
| d2c | 0.902 | 0.666 |
| LR.pos | 1.100 | 1.462 |
| LR.neg | 0.077 | 0.082 |
| DOR | 14.35 | 17.83 |

## 6.1 Accuracy

Figure - ROC & Cost on test data

## 6.2 Other Considerations

Transparency …

## 6.3 Selection

# Discussion

The two models developed allows Bank XYZ to predict who may or may not potentially default when asking for a credit card. The first method, an adaptive LASSO with logistic regression allows us to shrink parameters that are not important in predicting whether or not one will default. The 2 stage approach effectively handles feature selection and shrinkage simultaneously to remedy multicollinearity suspected between variables. The second model, the random forest, allows us to calculate several decision trees that ultimately best classify an individual’s default decision. In addition to high predictive performance, the random forest allows us to to visualize variable importance, which tells XYZ how much each feature contributes to the prediction. Both models’ parameters were tuned by examining fits over a wide range of values for weighting and penalization (LASSO), and number of decision trees / number of variables sampled at each split (Random Forest).

Predictive performance for each model was examined by comparing confusion matrices and calculating percentage of those correctly classified. In addition, a cost function was implemented where individuals were assigned a monetary value to allow Bank XYZ to see how much money would be gained or lost depending on how the model classified them.

# 7 Scenario Application

***Do customers who already have an account with Bank XYZ receive any favorable treatment?***

Given the way our logistic regression and random forest models are defined, we are able to say that those who already have an account with Bank XYZ do not receive favorable treatment. This is seen by looking at our variable importance plots and the variable indicating an existing account is the 4th from the bottom and also shows a value of nealy 0 importance, regardless of outcome group.

Was it picked up in any of our models? What advantage might it have that wouldn’t be evident in data?

***Suppose a credit card application is rejected, how would you explain the results to the customer?***

For TP and FP, does it differ?

Figure - “Tipping point” plots (if we took any value from specific applicant and replaced with mean of someone who would have been approved, does it change probability enough to cross threshold? iterative combinations of strongest more variable predictors or z-score of particular customer?)

Figure 7.1

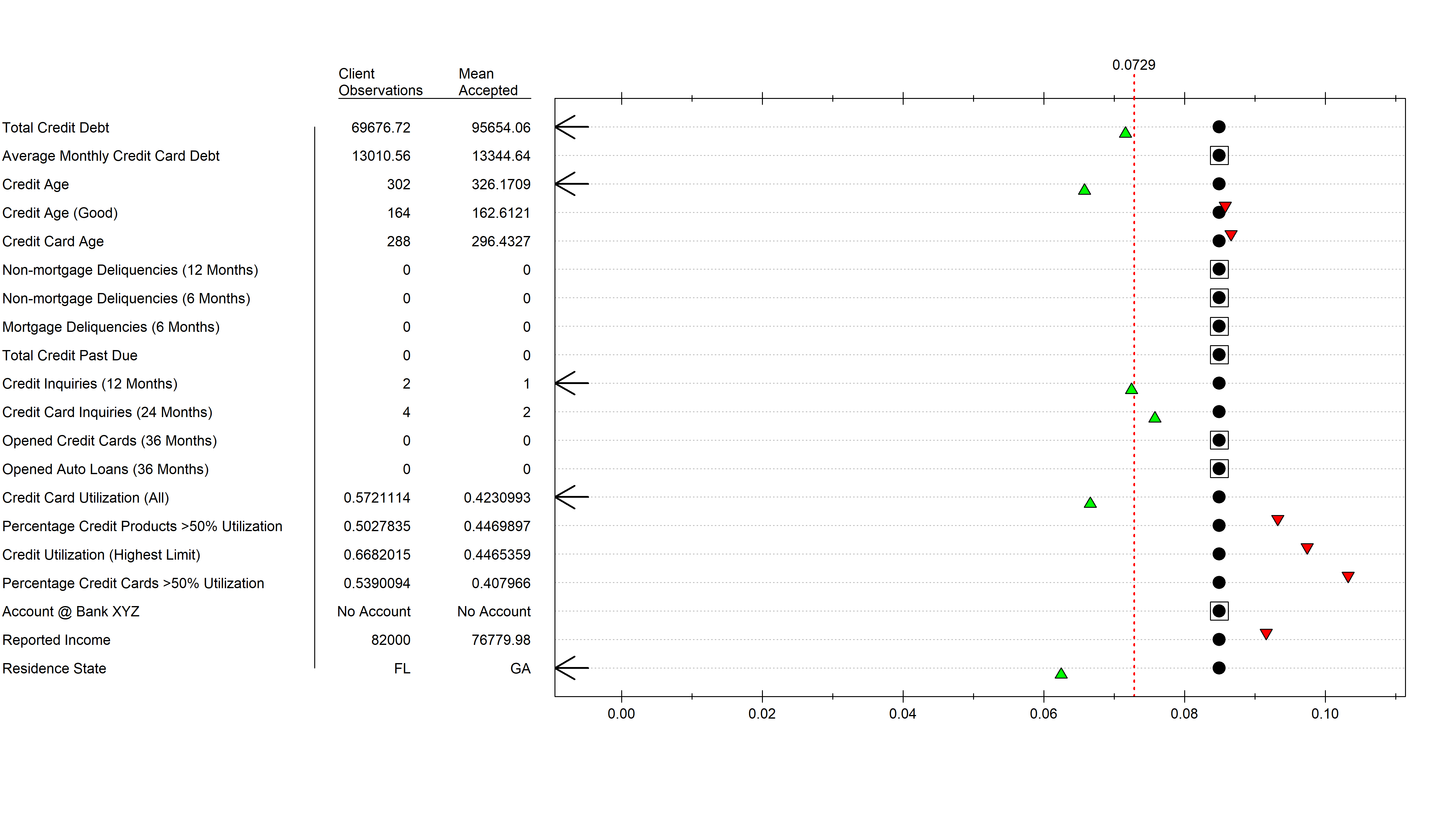


Figure 7.1: Explanation of A Rejected Applicant

# 8 Conclusions

Dilemma of basing credit approvals on default data (inherently missing approvals that should have been made, identifying only those that shouldn’t have been)

# References

1. Breiman L, Cutler A. Manual–setting up, using, and understanding random forests V4. 0. 2003. *URL https://www stat berkeley edu/~ breiman/Using\_random\_forests\_v4 0 pdf*. Published online 2011.

2. Guo P, Zeng F, Hu X, et al. Improved variable selection algorithm using a LASSO-type penalty, with an application to assessing hepatitis b infection relevant factors in community residents. *PLoS One*. 2015;10(7):e0134151.

3. Zou H. The adaptive lasso and its oracle properties. *Journal of the American statistical association*. 2006;101(476):1418-1429.

4. Gaujoux R. Using the doRNG package. Published online 2014.

5. *One ROC Curve and Cutoff Analysis*. NCSS; 2017. <https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/One_ROC_Curve_and_Cutoff_Analysis.pdf>

6. Xie Y. *Bookdown: Authoring Books and Technical Documents with r Markdown*.; 2020. <https://github.com/rstudio/bookdown>

7. Bürkner P-C. *Brms: Bayesian Regression Models Using Stan*.; 2021. <https://CRAN.R-project.org/package=brms>

8. Kuhn M. *Caret: Classification and Regression Training*.; 2020. <https://github.com/topepo/caret/>

9. Wei T, Simko V. *Corrplot: Visualization of a Correlation Matrix*.; 2017. <https://github.com/taiyun/corrplot>

10. Cui B. *DataExplorer: Automate Data Exploration and Treatment*.; 2020. <http://boxuancui.github.io/DataExplorer/>

11. Ryu C. *Dlookr: Tools for Data Diagnosis, Exploration, Transformation*.; 2021. <https://CRAN.R-project.org/package=dlookr>

12. Corporation M, Weston S. *doParallel: Foreach Parallel Adaptor for the Parallel Package*.; 2020. <https://CRAN.R-project.org/package=doParallel>

13. Gaujoux R. *doRNG: Generic Reproducible Parallel Backend for Foreach Loops*.; 2020. <https://renozao.github.io/doRNG>

14. Wickham H, François R, Henry L, Müller K. *Dplyr: A Grammar of Data Manipulation*.; 2021. <https://CRAN.R-project.org/package=dplyr>

15. Gohel D. *Flextable: Functions for Tabular Reporting*.; 2021. <https://CRAN.R-project.org/package=flextable>

16. Friedman J, Hastie T, Tibshirani R, Narasimhan B, Tay K, Simon N. *Glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*.; 2021. <https://CRAN.R-project.org/package=glmnet>

17. Müller K. *Here: A Simpler Way to Find Your Files*.; 2020. <https://CRAN.R-project.org/package=here>

18. Harrell FE Jr. *Hmisc: Harrell Miscellaneous*.; 2021. <https://CRAN.R-project.org/package=Hmisc>

19. Xie Y. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*.; 2021. <https://yihui.org/knitr/>

20. Bache SM, Wickham H. *Magrittr: A Forward-Pipe Operator for r*.; 2020. <https://CRAN.R-project.org/package=magrittr>

21. Pedersen TL. *Patchwork: The Composer of Plots*.; 2020. <https://CRAN.R-project.org/package=patchwork>

22. Robin X, Turck N, Hainard A, et al. *pROC: Display and Analyze ROC Curves*.; 2021. <http://expasy.org/tools/pROC/>

23. Henry L, Wickham H. *Purrr: Functional Programming Tools*.; 2020. <https://CRAN.R-project.org/package=purrr>

24. Breiman L, Cutler A, Liaw A, Wiener M. *randomForest: Breiman and Cutler’s Random Forests for Classification and Regression*.; 2018. <https://www.stat.berkeley.edu/~breiman/RandomForests/>

25. Allaire J, Xie Y, McPherson J, et al. *Rmarkdown: Dynamic Documents for r*.; 2021. <https://CRAN.R-project.org/package=rmarkdown>

26. Waring E, Quinn M, McNamara A, Arino de la Rubia E, Zhu H, Ellis S. *Skimr: Compact and Flexible Summaries of Data*.; 2021. <https://CRAN.R-project.org/package=skimr>

27. Müller K, Wickham H. *Tibble: Simple Data Frames*.; 2021. <https://CRAN.R-project.org/package=tibble>

28. Wickham H. *Tidyverse: Easily Install and Load the Tidyverse*.; 2019. <https://CRAN.R-project.org/package=tidyverse>

29. Xie Y. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman; Hall/CRC; 2016. <https://github.com/rstudio/bookdown>

30. Bürkner P-C. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*. 2017;80(1):1-28. doi:[10.18637/jss.v080.i01](https://doi.org/10.18637/jss.v080.i01)

31. Bürkner P-C. Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*. 2018;10(1):395-411. doi:[10.32614/RJ-2018-017](https://doi.org/10.32614/RJ-2018-017)

32. Wei T, Simko V. *R Package "Corrplot": Visualization of a Correlation Matrix*.; 2017. <https://github.com/taiyun/corrplot>

33. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*. 2010;33(1):1-22. <https://www.jstatsoft.org/v33/i01/>

34. Simon N, Friedman J, Hastie T, Tibshirani R. Regularization paths for cox’s proportional hazards model via coordinate descent. *Journal of Statistical Software*. 2011;39(5):1-13. <https://www.jstatsoft.org/v39/i05/>

35. Xie Y. *Dynamic Documents with R and Knitr*. 2nd ed. Chapman; Hall/CRC; 2015. <https://yihui.org/knitr/>

36. Xie Y. Knitr: A comprehensive tool for reproducible research in R. In: Stodden V, Leisch F, Peng RD, eds. *Implementing Reproducible Computational Research*. Chapman; Hall/CRC; 2014. <http://www.crcpress.com/product/isbn/9781466561595>

37. Robin X, Turck N, Hainard A, et al. pROC: An open-source package for r and s+ to analyze and compare ROC curves. *BMC Bioinformatics*. 2011;12:77.

38. Liaw A, Wiener M. Classification and regression by randomForest. *R News*. 2002;2(3):18-22. <https://CRAN.R-project.org/doc/Rnews/>

39. Xie Y, Allaire JJ, Grolemund G. *R Markdown: The Definitive Guide*. Chapman; Hall/CRC; 2018. <https://bookdown.org/yihui/rmarkdown>

40. Xie Y, Dervieux C, Riederer E. *R Markdown Cookbook*. Chapman; Hall/CRC; 2020. <https://bookdown.org/yihui/rmarkdown-cookbook>

41. Wickham H, Averick M, Bryan J, et al. Welcome to the tidyverse. *Journal of Open Source Software*. 2019;4(43):1686. doi:[10.21105/joss.01686](https://doi.org/10.21105/joss.01686)

1. Department of Epidemiology and Biostatistics [↑](#footnote-ref-20)
2. Department of Computer Science [↑](#footnote-ref-21)
3. A positive in this sense refers to the prediction of Default, where a true positive refers to someone we predicted and who in fact defaulted. The true positives and the false positives were assumed to not have contributed in profit calculations as they represent credit denials (hence zeros in the equation). The average monthly credit card debt for an individual was calculated from the validation dataset using the variable avg\_card\_debt and its geometric mean for the those who did not default () and defaulted (), separately. We assumed the profit margins on purchases () and interest payments () to be 1% and 10% of the average monthly credit card debt. The average number of deliquencies () was calculated using the validation dataset and the non\_mtg\_acc\_past\_due\_12\_months\_num variable and its quadratic mean for those who did not default divided by 12 (to get monthly). We understand the avg\_card\_debt variable can correspond to many credit cards, but we did not feel capable of estimating an individual’s total number of credit cards to correct for this. Also, these calculations were made at the population level and, therefore, assumed the same values for each credit applicant; although, it may be possible to reconfigure to incorporate an individual’s data. [↑](#footnote-ref-28)