Project Report - Group Project 1

Group Project Contribution Table	
Project Team Member	Contribution
Steven Spence	Data Exploration / Preprocessing / Probability of Default Model
Kareem Rogers	Drafting of Project Report and PowerPoint Presentation
Srihari Kosanam	Data Exploration / Preprocessing / Models / Prediction Files
Manali Padhye	Data Exploration / Preprocessing / Loss Given Default Model
Yuqiao Xu	Drafting of Project Report and PowerPoint Presentation

1. Project Goal

This project has three scenarios, each with slightly different real-world boundary condition parameters. The key to these scenarios is to minimize the risk of financial investment and maximize the profit returns on financial investments based on the clients they choose to lend/loan money. The risk will be assessed by reviewing a client's probability of defaulting alongside their potential loss given default. Opportunities will be assessed by evaluating the amount the company can earn from a client based on interest rate, loan amount, and life of a

loan. Together these factors will give us an expected gain or loss which the bank will use in a loan application decision.

For the first scenario, the bank has $1.4 billion to be divided among 25,471 clients. In this scenario, the bank has enough money to give all clients their requested loan; however, the goal isn't to loan funds to every client on the roster but only to those who can generate high expected gain with a low default rate probability. As a result, the prediction model must reflect that logic and figure out which person will be approved and which will be rejected by the bank.

For the second scenario, the bank only has $450 million, so the company doesn't have enough capital for giving out loans to everyone (25,471 clients). Therefore, the goal is to select the top clients who will generate a profitable return. An algorithm/code will be created to find how many people we are giving a significant portion of the $450 million to and what is the profit return will be for each client. Based on this algorithm, we will only give out loans to people who have lower default rates and high expected gains.

For the last scenario, the bank has the same capital as scenario one ($1.4 billion). However, in this scenario, the model will have to take into account the different proposed interest rates for each customer. The proposed interest rate for each person will be different, so we need to figure out a different expected gain and loss for each customer to decide if they are approved or rejected by the bank.

2. Overview of data, including data exploration analysis

The banking dataset that was given to our group at first glance was overwhelming in terms of the amount of data and provided information. Instead of the different names of the clients, the clients/customers identified in the form of ID numbers and different variables. Additionally, a disadvantage we experience was the poor labeling of the columns because we seek to anticipate

and incorporate both the default and the severity of the losses that result. In doing so, we are building a bridge between traditional banking, where we are looking at reducing the consumption of economic capital, to an asset-management perspective, where we optimize on the risk to the financial investor. The anonymous variable titles limited us from applying any domain knowledge; therefore we had to strictly rely on data preparation and correlation analysis, which brings us to the task of the preparation of the dataset.

In preparing the data, we had to do an extensive amount of data cleaning. First, we reviewed the structure of the data to see what types of variables were present. The data set only contains numeric variables only. As a result, we found that there are not much restructuring of needed data types. Next, we reviewed the amount of missing data in the data set, which are called out in the data structure section below. Then we discussed the data set for variables that had a zero variance or near zero variance, as well as those were highly correlated to each other.

What we realized was the dataset had a "loss" column, so clients who had zero in the "loss" column indicates he/she paid their loans off without defaulting. On the other hand, those who had numerical value more than zero (0) had defaulted in their loans some point in time. For example, If Mr. Bob Roberts had a numerical value of (5) five in the default column, that would indicate he paid off 95% of his loan before defaulting.

Another interesting fact that we found out was that the majority of the defaulting clients we encountered paid 75% of their loan before the defaulting, as seen in figure (1). Only a small amount of individuals defaulted on the entire loan, the reason for that is unknown. However, our group came up with some rational reasons as to why and where it may cause that happen. We thought it might because of several factors such as interest rate, loss of steady income, or some other financial hardship. It is difficult to state precisely why without additional domain knowledge.

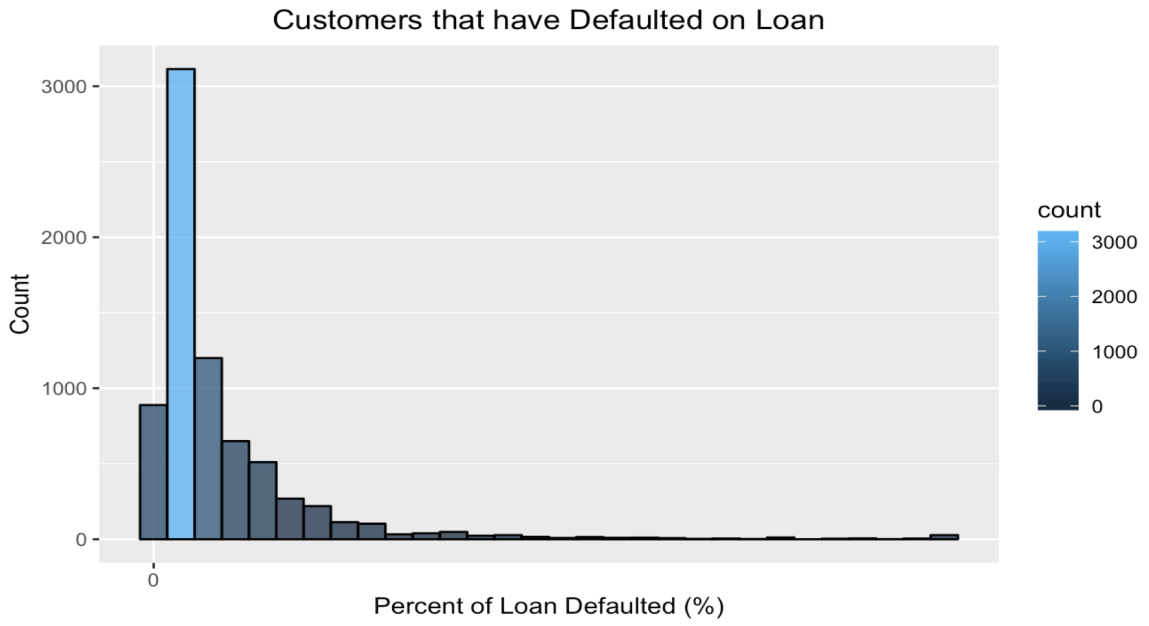


Figure 1 - Histogram of Default % on Defaulting Customers

The Structure of the Data

The given data structure is your primary dataset which consists of rows and columns. The dataset consists of 80,000 rows, which corresponds to each client, and 762 columns, which

corresponds to various anonymous variables to describe the client's profile. Additionally, all variables are numeric with 80,000 total customers/observations and 762 total amount of variables plus one of the variables representing the "loss" percentage of the loan for each customer. The variables are generic in this particular situation, which was difficult because they were blind to their true meaning. The dataset also contains two identical variables which were a link to the identification of each customer

Reviewing the Missing Values

When reviewing the missing data within the dataset we found that there were numerous missing values throughout the entire data set. For each customer entry/account, we found that missing values ranged anywhere from 0% missing to 47.97% missing. For each unknown variable, we found that the missing values for a specific variable can vary anywhere from 0% to 17.83%.

Comparison of Defaulting and Non-Defaulting Customers:

A review of the defaulting versus non-defaulting customers found that 72,621 customers paid their loans off in full, while 7,379 customers ended up defaulting on their loans at some time. These facts gave us a historic default rate of 10.16% for the given data set. We will provide the further depicted below in Figure 2.

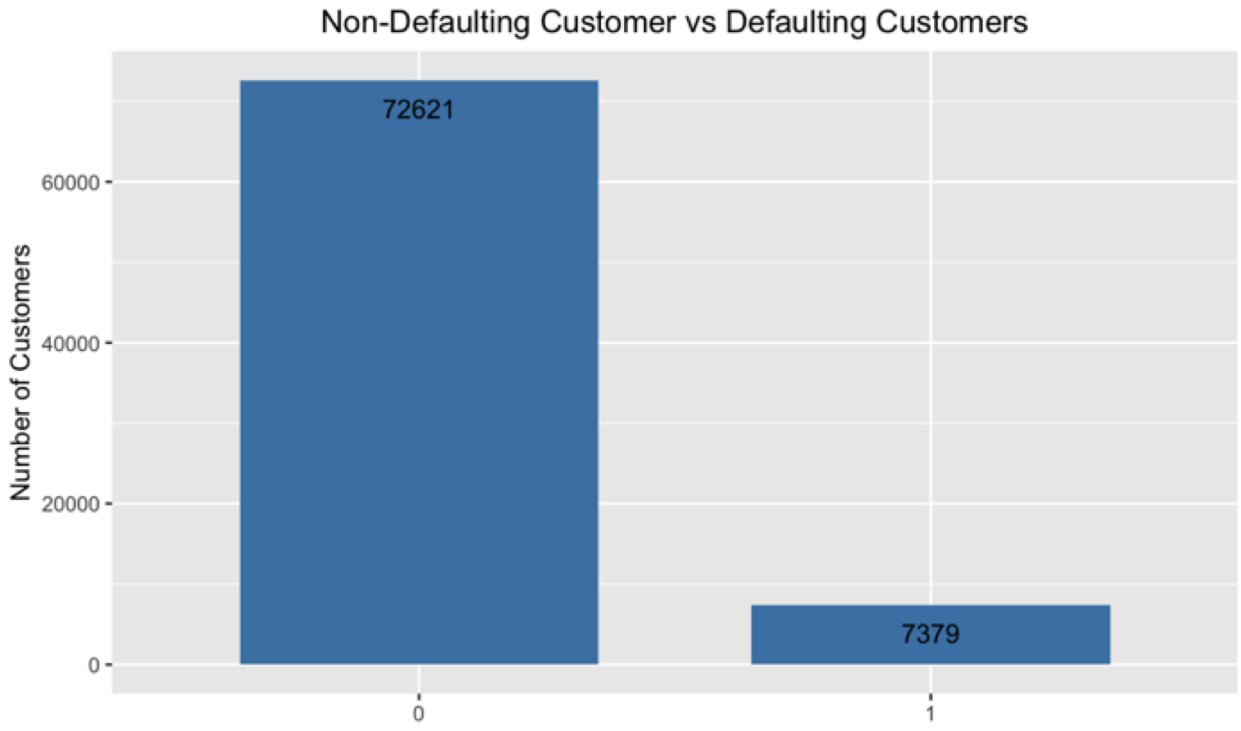


Figure 2 - Comparison of defaulting customer to non-defaulting customers

3. Details of Your Modeling Strategy

While conducting the modeling strategy, we looked at numerous techniques when creating a modeling strategy. The plan was to separate the models into two separate models, one to predict the probability of default (PD), and one to predict the expected loss given default (LGD). The initial steps for both models involved cleaning and reducing the data set. The first task was to shrink the data set into a more manageable size for model creation, known as feature selection. First of all, we removed near zero variance variables, as well as highly correlated variables. The result by doing that is that we got a reduction from 740 variables to 246 variables. Next, we imputed the missing values by a median imputation method before running through a regularization model, so that we further reduced the data set down to the most critical variables before the model building.

In this project, we used the lasso regression analysis method to perform both the variable selection and regularization, so that the bank could increase the prediction accuracy and interpretability of the people's default rates. The input for this model was 246 variables. The LASSO (Least Absolute Shrinkage and Selection Operator) is a regression method that when people need to penalize the absolute size of the regression coefficients. In this project, the bank wants to know for each customer if they are approved or rejected by the bank based on the customer's default history.

In Figure 3 below, the graph shows the Optimum Lamda value for the feature selection that was reduced from 246 variables to 180 variables. The first vertical dashed line indicates the lambda. Min value, while the second dash line shows the lambda value within one standard deviation to further reduce the variables. We decided to go with the first, lambda.Min value.

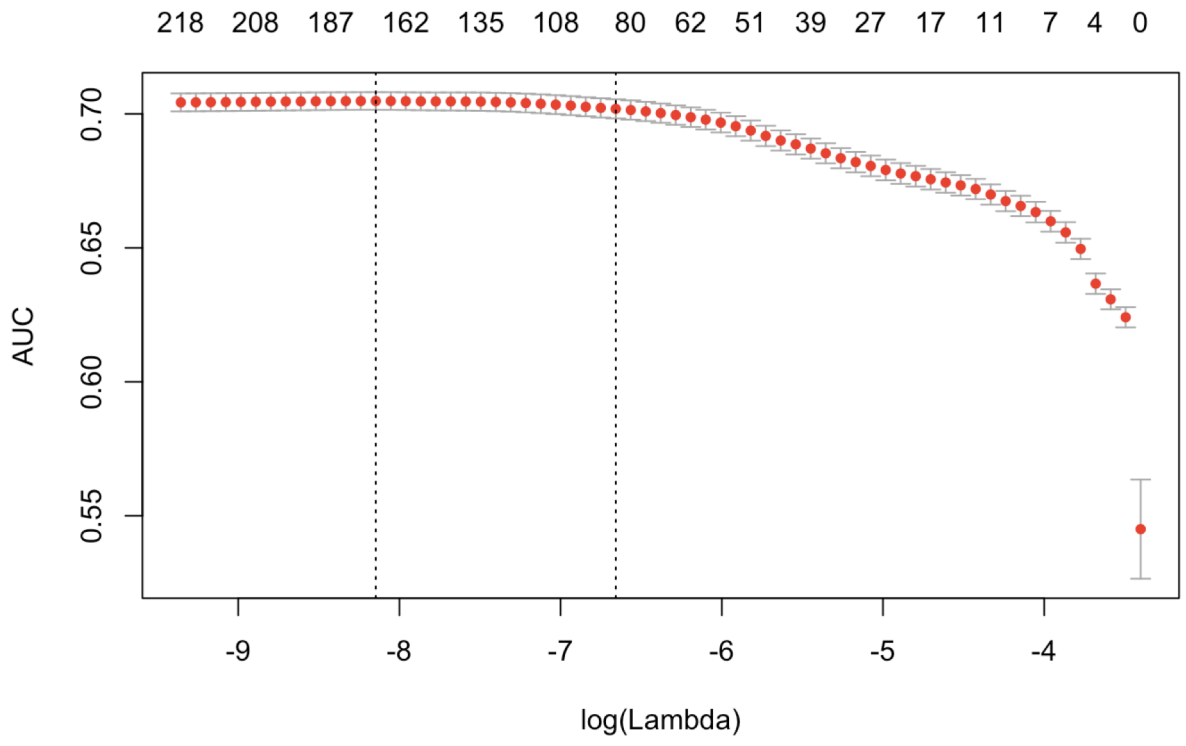


Figure 3 - Graph of the Lasso regression out and AUC values

The Lasso model returned a total of around 180 variables as important to the default ("0" or "1") target variable. Their coefficient values selected the top 10 most important variables. These were selected to be kept out of the further analysis, and the remaining variables would be run in a principal component analysis (PCA). The top 10 variables by lasso were stored in variable "cv_lasso_coefs_top_10**",** the remaining values were stored in a variable "cv_lasso_coefs_pca".

The PCA was used to reduce the remaining large dataset of variables into an even smaller amount to better handle the information but at the same time holds/keep the most valuable piece of information in the process. The out was created from 80, 000 samples and 170 variables. The following lines below show the output from the PCA analysis. The data was centered, scaled, and transformed to aid in the process. The group decided on a threshold of 75% variation captured by the remaining variables. Several iterations were run to determine a manageable amount of data to further process. The resulting PCA gave an output of 45 principal components to capture that 75% variability.

"Pre-processing:

- *Centered (150)*
- *Ignored (0)*
- *Principal component signal extraction (150)*
- *Scaled (150)*
- *Yeo-johnson transformation (114)*

Lambda estimates for Yeo-Johnson Transformation:

- *Min - (-2.9215)*
- *1st Qu - (-0.1564)*
- *Median - (0.2810)*
- *Mean - (0.3880)*

- *3rd Qu -(0.9313)*
- *Max (2.9567)*

45 Components were needed to capture 75 percent of the variance leftover in the variables."

The output from this PCA was combined with the previously held out 10 variables (for a total of 55) to be entered into a random forest and XGboost model for predicting the probability of default for each customer.

XGboost is designed and optimized for boosting trees algorithms. By employing multi-threads and imposing regularization, XGBoost is able to utilize more computational power and get a more accurate prediction. In terms of how we implemented it within our codes, it was giving us a higher rate of a false negative and longer processing times; hence we decided to try the random forest strategy.

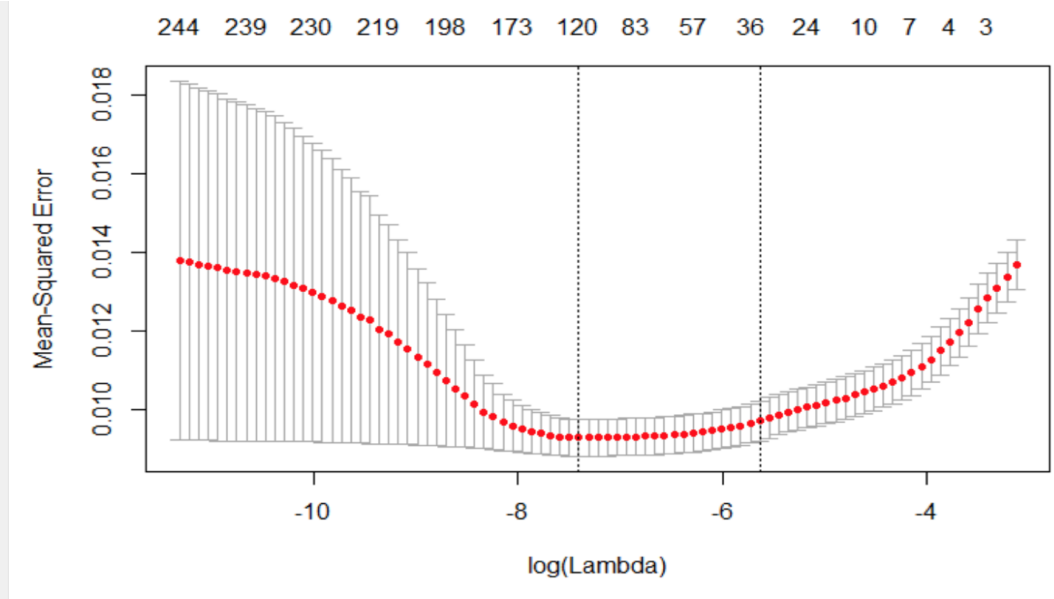
When using the random forest within our coding, it essentially creates a large number of decision trees where very decisions that occur is linked back to every decision tree in the process. So when executed the random forest it gave us a bunch of decision trees and created an output of the average value for each customer chance of defaulting. Two different packages for random forest were used in the modeling process. The first was the "caret" package with the "ranger" and "rf" functions. These packages allow for further hyper tuning parameters; however, it increased processing time significantly. The processing time took upwards of twenty hours, so the team decided to go with the "Randomforest" package in R. This package does not have as many hyper-tuning parameters, but the faster processing time allowed us to attempted more iterations.

For the LGD(Loss Given default) model the target variable was continuous, so it was clear that we had to use a regression model for the prediction. For training an LGD model the model, we used the data of only the defaulted customers. The original loss column was split into

default and loss and it is the loss given a default value. The values were normalized by converting it to fractions instead of percentages.

We made the feature selection again for this model since the target variable was different (loss given default and not just defaulted or not). To shrink the data set into a more manageable size for model creation we again removed the near-zero variance variables, as well as highly correlated variables, this time only for the defaulted customers. This resulted in a reduction from 764 variables to 253 variables. Next, we imputed the missing values by a median imputation. After this step, it gave us the data to apply Lasso. Applying Lasso on this cleaned data further reduced the variables to 121.

The graph shows the Optimum Lambda value for the feature selection which was reduced from 246 variables to 120 variables.



Once the data was reduced to a manageable size, we decided to use Ridge regression to get the predictions. Since the length of data was compliant, we used all 120 variables in building a Ridge Regression model to predict Loss Given Default. The objective function was to minimize the mean absolute error (MAE).

4. Estimation of the model's performance

Random Forest for Probability of Default (PD):

We used AUC as a metric to measure model performance. This process takes into account the True positive and False positive probabilities as samples are chosen from the prediction model. The "RandomForest" package was used, and the "ntree" and "mtry" variables were able to be hyper-tuned to improve the model further. Figure 4 and Figure 5 below shows the improved AUC values with changes in those parameters. It was determined that the model would be based on a "ntree" value of 500 and "mtry" value of 5, which returned an AUC value of 0.6334.

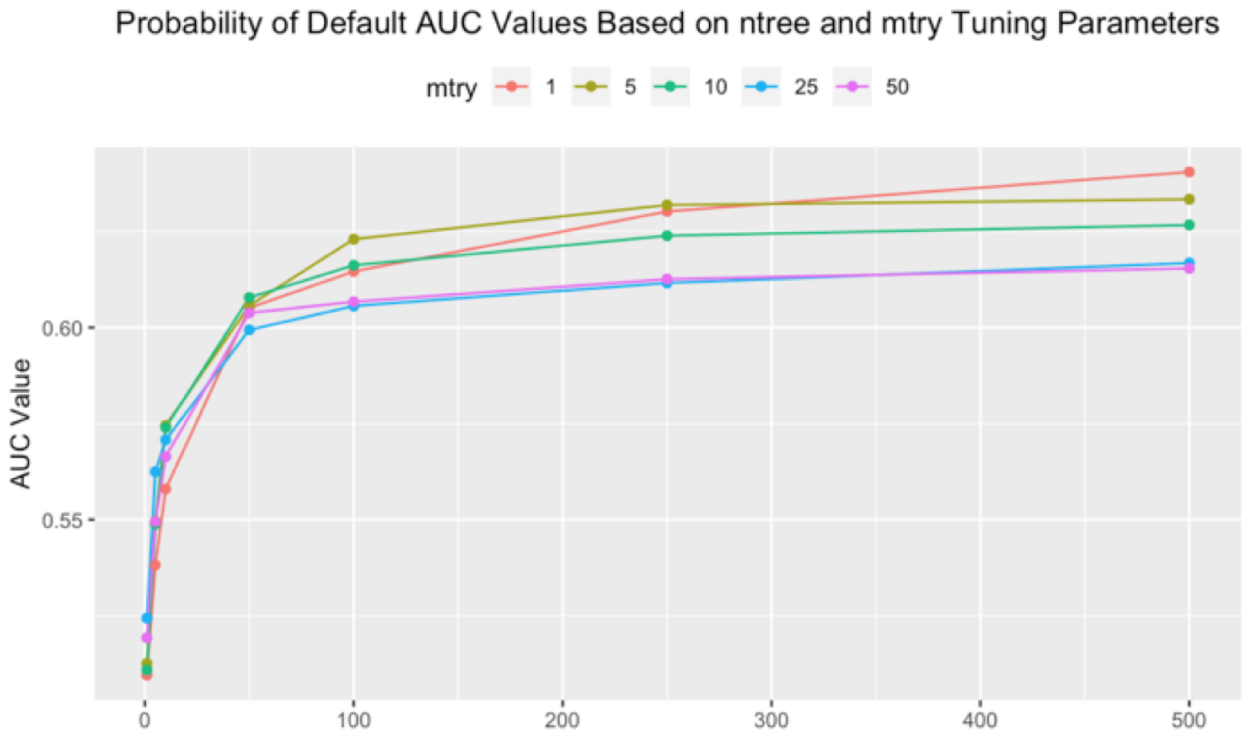


Figure 4 - Hypertuning: We adjusted N trees, and M try to find the best optimal value

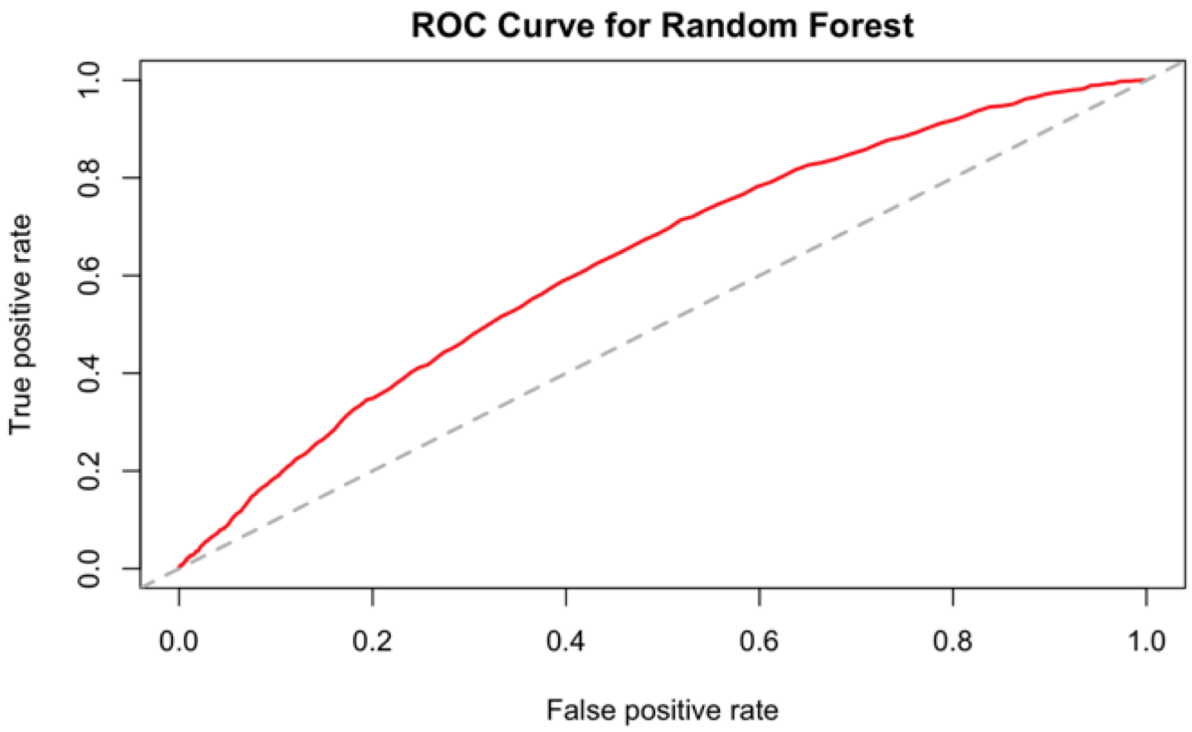
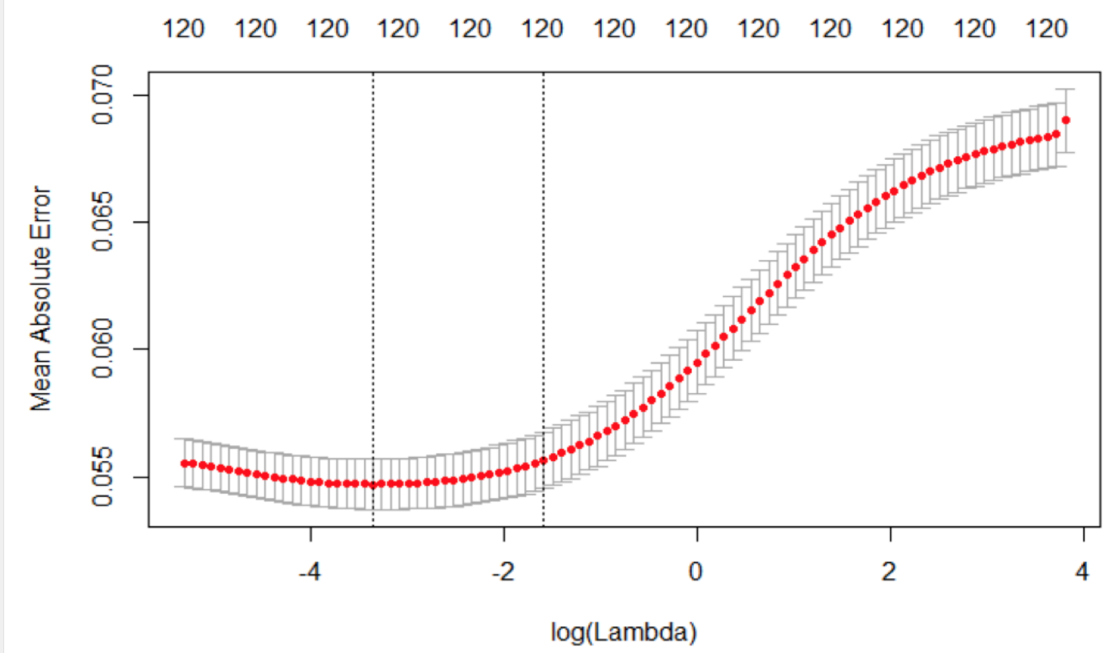


Figure 5 - AUC Value of 0.6334 for the probability of default model

Our group believed the performance could've been further improved and obtain better results if we have access to more powerful processing power. The hyper tuning process for the probability of default model was limited, due to the amount of time our systems took to run the model. The processing time more than 20 hours was observed on some of the models. Therefore, a quick, more efficient method was used to get an acceptable model.

Ridge Regression for Loss Given Default(LGD Model)

We used ridge regression for the calculation of LGD and MAE(Mean Absolute Error) as a metric to measure the performance of the model.MAE for the LGD Model is 0.05, and the number of Lambda values tried were 100. Out of the 100 lambda values worked the best lambda was selected as 0.0348.



5. Insights and Conclusions

Scenario 1

Problem:

In this scenario, we assume that your bank has a total capital of $1.4B for giving out loans. Loans are all fixed term (5-years), and the annual interest rate is 4.32% for all approved customers. To simplify the problem, we assume that the interest rate is not calculated as a compound rate. That is to say, for example, if Mrs. White is taking a loan of $20,000. She will return $20,000 (the capital) plus 5*4.32%*20,000=4320 after five years if she does not default. If she defaults at 80%, it means that she would pay back only 20% of the capital 20,000*20%=$4,000 and zero interest (i.e., the loss is $16,000 for your bank).

You are given the training dataset which contains a list of variables and the target variable that is “loss.” “loss” defines the percentage of the loan at which the customer defaulted. If “loss” is zero you can imagine that the customer has fully paid back the capital and interest. If the “loss” is greater than zero, it means that the customer has defaulted. “loss” is expressed in

percentage so if the loss is 10, then it means that the customer has paid pack 90% of the capital but zero interests.

Based on this data, you will need to train model(s) to decide which customer listed in the "test_scenario1_2.csv" file (a total of 25471 customers) you would approve and which one you would reject. Your goal is to maximize the profit for your bank. Based on your decisions, I will calculate the total return after five years (which consists of benefits from customers whom you approved and paid back the capital and interest and losses from those whom you have accepted and have defaulted).

For this part, you will submit a CSV file with 25471 rows each for one customer and a single column containing 1 and 0. 1 means approved, 0 means rejected.

Note that there is a column "requested_loan" which shows the requested loan amount in US dollar. This column does not exist in the training dataset. I leave it up to you how to use this column.

Solution:

In scenario 1, the bank was given 1.4 billion dollars to be loan out to 25,471 clients, to minimize potential risks and maximize the profit returns generate through interest and the ability to pay 25,468 clients because those clients have the highest expected gain and lowest expective lose. The total amount of request loans totaled just over 1.2 billion dollars; therefore, our group had enough money to loan all the clients but needed a way to limit our risk.

The expected gains from each customer were calculated by taking the values of one minus the probability of default and multiplying by the interest rate, years, and requested loan amount. The expected loss from each customer was taking into account the probability of default and multiplying it by the loss given default percentage and multiplying that by the request loan amount. Next, the expected gains were compared to the expected losses for each customer. After analyzing these values, it was found that only three customers had a negative expected loss.

Therefore, in this scenario, it was determined that we would approve all customers with positive expected values.

Scenario 2

Problem:

Exactly similar to scenario 1 but in this case, your bank budget to give loans is $450 M. Again you need to submit a csv file with 25471 rows each for one customer and a single column containing 1 and 0. 1 means approved, 0 means rejected.

Solution:

In Scenario 2, the bank is only allowed to give out a max of 450 million dollars, with a client list of 25,471 persons. This scenario was slightly different from the first scenario because the bank does not have enough money to loan every single customer. Therefore, we had to quantify the customers with the most significant expected gains. This was completed similarly to the first scenario. Expected losses and gains were compared to get a different value. Then, the differences were ranked from highest to lowest and approved all customers with the highest increases until we reached our $450 million limits.

After careful consideration and applying said code that will be displayed in the "R codes and script" section, we successfully issued an amount of 449,950,652 dollars out of 450,000,000 to 5,653 clients. The top 5 highest expected gain of $18,865.35, $18,694.38, $18,533.78 and $18,503.40. This was a result of ranking the customers from the highest expected gains to lowest expected gains.

Scenario 3

Problem:

In this case, you can see each customer is proposing an interest rate (column "Proposed_Intrest_Rate"). So the interest rate varies for different customers. For this scenario, we assume your bank has $1.4B available to give loans. The requested loan amounts and proposed interest rates are included in the file "test_scenario3.csv"

For this part, you will submit a CSV file with 25471 rows each for one customer and a single column containing 1 and 0. 1 means approved, 0 means rejected.

Solution:

This scenario is similar to the first scenario. The bank has enough money to approve all the requested loans; however, each customer has a different proposed interest rate and required loan amount. As a result, the expected loss and gains were different for the customers. Therefore, we re-ran the formulas to account for the changes in loans and interest rates. Next, we ranked the clients from the highest gains to the expected lowest gains.

In this scenario, we decided to be more strict in regards to the number of clients and the amount of money to loan out. Therefore, we created a threshold for clients/customers to distinguish between which clients will be approved and which ones will be rejected. As a result, we decided to approve all loans were the probability of default was found to be below 20%. Subsequently, anyone above 20% probability of default would be rejected. In the end, we found that about 2,209 people will be rejected and 23,271 were approved and with a total of $1,159,908,633 loaned out. This resulted in a rejection rate of approximately 10%, which lines up with the historical data and rate of default for customers.

References

2017. An Introduction to XGBoost R package, Retrieve from: https://www.r-bloggers.com/an-introduction-to-xgboost-r-package/amp/

2018. R-Random Forest, Retrieve from: https://www.tutorialspoint.com/r/r_random_forest.htm

NCSS Statistical Software, Ridge Regression, Retrieve from: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

2018, Introduction to Principal Components and FactorAnalysis, Retrieve from: ftp://statgen.ncsu.edu/pub/thorne/molevoclass/AtchleyOct19.pdf

Stephanie, 2015, Lasso Regression: Simple Definition, Retrieve from, https://www.statisticshowto.datasciencecentral.com/lasso-regression/