ADVANCED DATA MINING AND PREDICTIVE ANALYSIS

FINAL PROJECT REPORT

GROUP 5:

NAMES	CONTRIBUTION
Nikitha Chigurupati	Loss Given Default Model and Report
Rajeev Varma	Probability of Default, Report and Power Point
Nithin Varma	Probability of Default code and Report

TABLES OF CONTENT

I. PROJECT GOAL:

The objective of this project is to predict if a loan will default and the amount of loss that will be incurred if it does. Unlike traditional finance-based approaches to this problem, which make a binary distinction between good and bad counterparties, we strive to foresee and incorporate both default and the severity of the resulting losses. In doing so, we are bridging the gap between traditional banking, where we seek to reduce the consumption of economic capital, and asset management, where we optimize the risk to the financial investor.

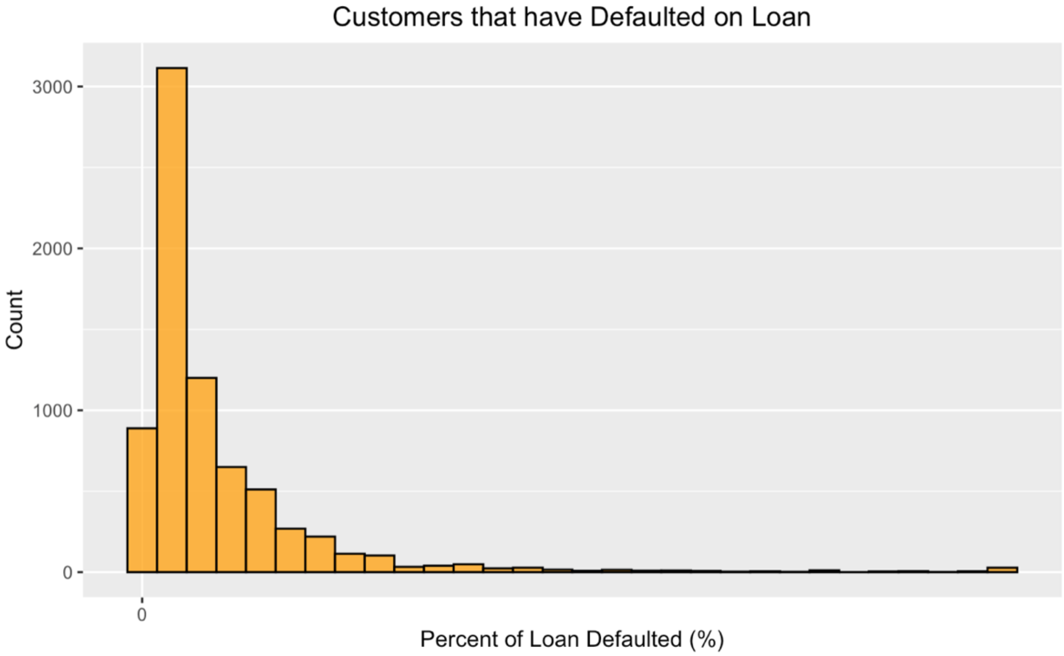
II. OVERVIEW/ STRUCTURE OF THE DATA:

The Data consists of 80000 observations and 763 variables which are all numeric data. We mainly considered those variables which had a visible effect on the dependent variable loss. Understanding and analyzing the effect of loss and the amount of loss is what we observed through this project via graphs, plots, and tables.

We had to do a lot of data cleaning while preparing the data. First, we looked at the data structure to see what kinds of variables were present. The data collection contains just numeric variables. As a result, we discovered that there is not much reorganization of required data kinds. After that, we looked at the amount of missing data in the data set, which is mentioned in the data structure section below. Then we analyzed the data set for variables with zero or close to zero variance, as well as those that were highly associated with one another.

We discovered that the dataset contained a "loss" column, and clients with a zero in the "loss" column had paid off their debts without defaulting. Those with a numerical value greater than zero (0), on the other hand, had defaulted on their loans at some point in time.

Another noteworthy piece of information we discovered was that most of the default clients we came across paid 75% of their loan before defaulting, as illustrated below. Only a few clients defaulted on the entire loan, the reason for this is unknown. However, our group came up with some reasonable justifications for why and where this might have occurred. We suspected it was due to a wide range of causes such as interest rates, a loss of regular revenue, or some other financial difficulties. Without additional subject expertise, it is impossible to explain why.



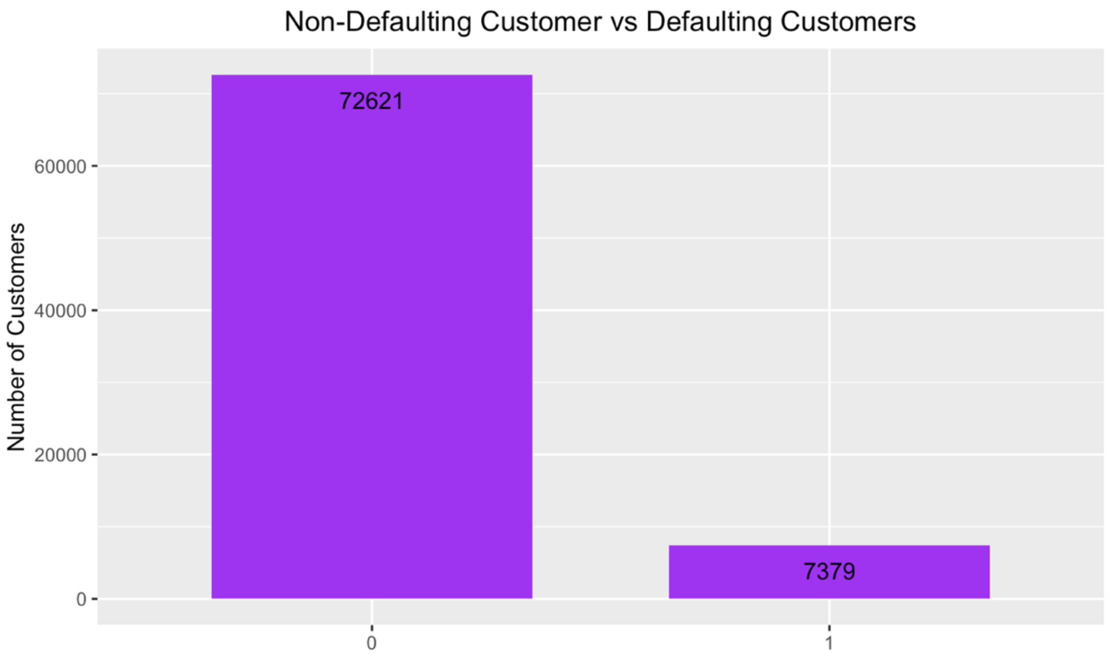
The given data structure represents your primary dataset, which is made up of rows and columns. The dataset is made up of 80,000 rows, one for each client, and 763 columns, 5 of which relate to various anonymous variables that describe the client's profile. In addition, all variables are numerical, with 80,000 total customers or observations and 763 total variables, one of which represents the "loss" percentage of the loan for each customer. The dataset also included two identical variables that served as a link to each customer's identification.

- **MISSING VALUES IN THE DATASET:**

When we examined the missing data in the dataset, we noticed that there were multiple missing values across the entire data set. We observed that missing values ranged from 0% to 47.97% for each customer entry or account. We observed that the missing values for each unknown variable may vary anywhere from 0% to 17.83%.

- **DEFAULTING AND NON-DEFAULTING CUSTOMERS COMPARISION:**

A comparison of defaulting against non-defaulting consumers showed that 72,621 customers paid off their debts in full, whereas 7,379 customers defaulted at some point. Based on these factors, we calculated a historic default rate of 10.16% for the given data set. We will supply the additional information represented in the below figure.



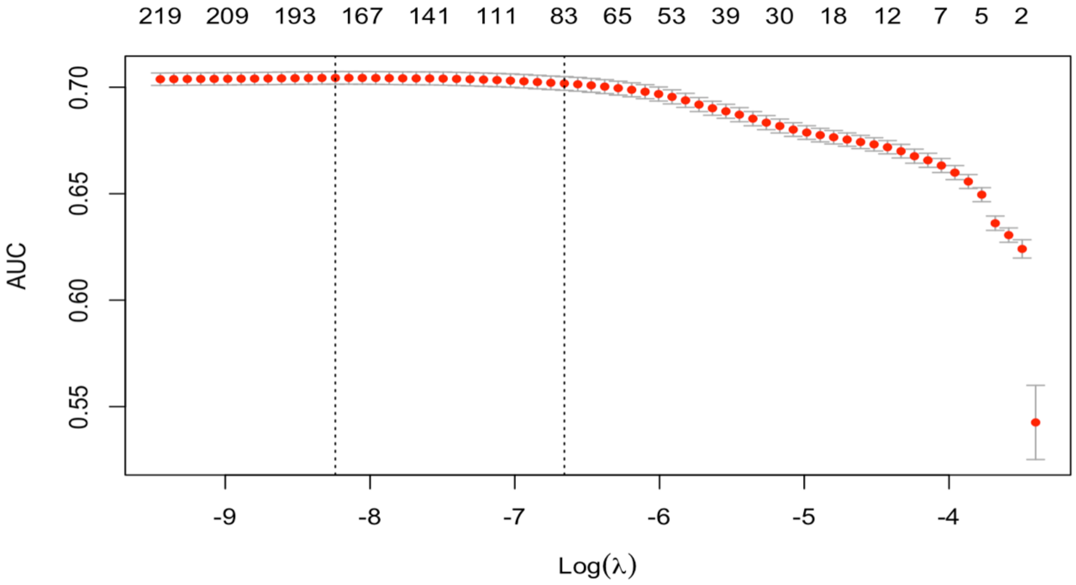
III. DETAILS OF MODELLING STRATEGY:

We examined a variety of methods for developing a modeling strategy while conducting the modeling approach. The idea was to divide the models into two distinct models, one to forecast Probability of Default (PD) and the other to forecast projected Loss Given Default (LGD). Both models' initial stages involved cleaning and reducing the data set. The first step, known as feature selection, was to reduce the size of the dataset so that it could be used to build a model.

First, we eliminated variables with close to zero variance and those with high correlation. Therefore, we were able to reduce the number of variables from 763 to 253. After that, we used a regularization model to further limit the dataset to the most important and crucial variables before developing the model, and then replacing the missing values using a median imputation method.

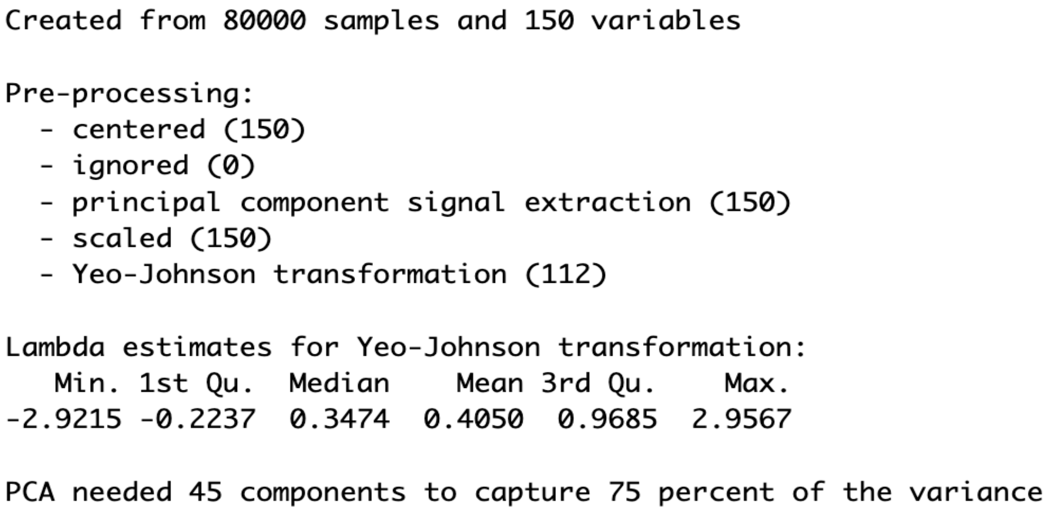
To improve the predictability and understanding of the customer's default rates for the bank, we used the lasso regression analysis approach in this project to accomplish both Feature selection and Regularization. This model included 253 variables as its input. When it's necessary to penalize the absolute magnitude of the regression coefficients, the LASSO (Least Absolute Shrinkage and Selection Operator) regression approach is used. In this project, the bank needs to determine whether each customer or a client has been approved based on their history of defaults.

The graph displays the Optimum Lambda value for the feature selection, which was decreased from 253 variables to 180 variables. The lambda is indicated by the first vertical dashed line. The first dashed line displays the minimum value, while the second dashed line shows the lambda value within one standard deviation to further minimize the variables. We chose the first option, lambda.Minimum value.



The Lasso model identified approximately 180 factors as being important to the default ("0" or "1") target variable. Their coefficient values determined the top ten most significant variables. These were chosen to be excluded from further analysis, while the remaining variables would be subjected to a principal component analysis (PCA). The top ten lasso variables were saved in the variable "cv_lasso_coefs_top_10," while the remaining values were saved in the variable "cv_lasso_coefs_pca."

The PCA was used to reduce the remaining large dataset of variables into an even smaller quantity to better handle the information while simultaneously holding/keeping the most valuable piece of information. The output was generated using 80,000 samples and 150 variables (after excluding the Top 10 Lasso Variables). The output of the PCA analysis is shown in the lines below. To aid in the process, the data was centered, scaled, and transformed. The group decided on a 75% variation capture threshold by the remaining variables. Several iterations were performed to determine a manageable amount of data to process further. The resulting PCA produced 45 main components, which captured 75% of the variability.



```
Created from 80000 samples and 150 variables

Pre-processing:
  - centered (150)
  - ignored (0)
  - principal component signal extraction (150)
  - scaled (150)
  - Yeo-Johnson transformation (112)

Lambda estimates for Yeo-Johnson transformation:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-2.9215 -0.2237  0.3474  0.4050  0.9685  2.9567 

PCA needed 45 components to capture 75 percent of the variance
```

The results of this PCA were combined with the previously held out 10 variables (the Top 10 Lasso Variables) for a total of 55 and fed into a random forest and XGboost model for predicting the probability of default for each client in the bank.

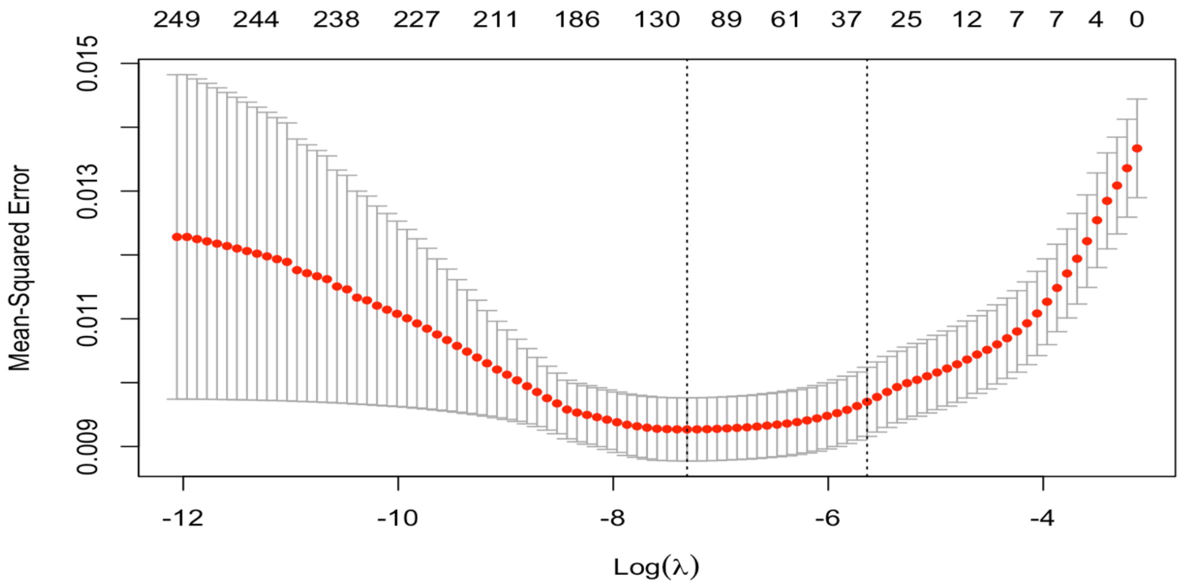
XGboost was developed to boost tree algorithms. XGBoost can use more computational resources and obtain a more accurate forecast by leveraging multi-threads and enforcing regularization. We opted to test the random forest technique because the way we applied it within our codes resulted in a greater rate of false negatives and longer processing times.

When we use the random forest in our coding, it essentially builds a vast number of decision trees where each decision is linked back to every decision tree in the process. So, when we ran the random forest, we got a bunch of decision trees and an output with the average value for each customer's chance of defaulting. In the modeling process, two different random forest packages were used. The first of them was the "caret" package, which contained the "ranger" and "rf" functions. Although there are fewer hyper-tuning parameters in this package, its quicker execution time allowed us to conduct more iterations.

We used information gathered from solely the defaulting clients to train an LGD model. The original loss field was divided into default and loss, with loss being assigned a default value. The data was normalized by turning it into fractions rather than percentages.

Given that the target variable was different (loss given default rather than just defaulted or not), we had to start over with the feature selection for this model. To make the data set more manageable for model building, we eliminated the near-zero variance variables as well as the highly correlated variables, but just for the defaulted customers this time. This reduced the number of variables from 763 to 253. The missing data were then imputed using a median imputation method. This phase provided us with the data we needed to use Lasso. Using Lasso on this cleaned data reduced the number of variables to 120.

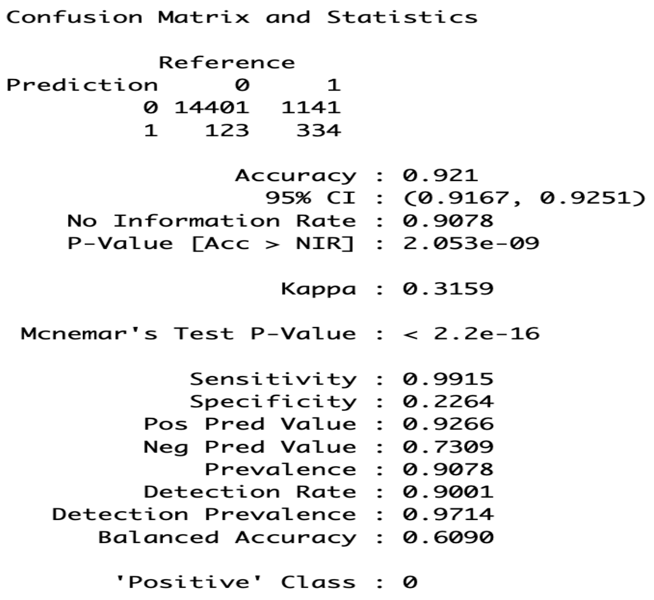
The graph depicts the Optimum Lambda value for feature selection, which was decreased from 253 to 120 variables.



We opted to utilize Ridge regression to acquire the predictions once the data had been reduced to a bearable quantity. We used all 120 variables in a Ridge Regression model to predict Loss Given Default because the data length was acceptable. The goal function was to reduce the mean absolute error (MAE).

IV. ESTIMATING THE MODEL'S PERFORMANCE:

CONFUSION MATRIX FOR RANDOM FOREST MODEL:

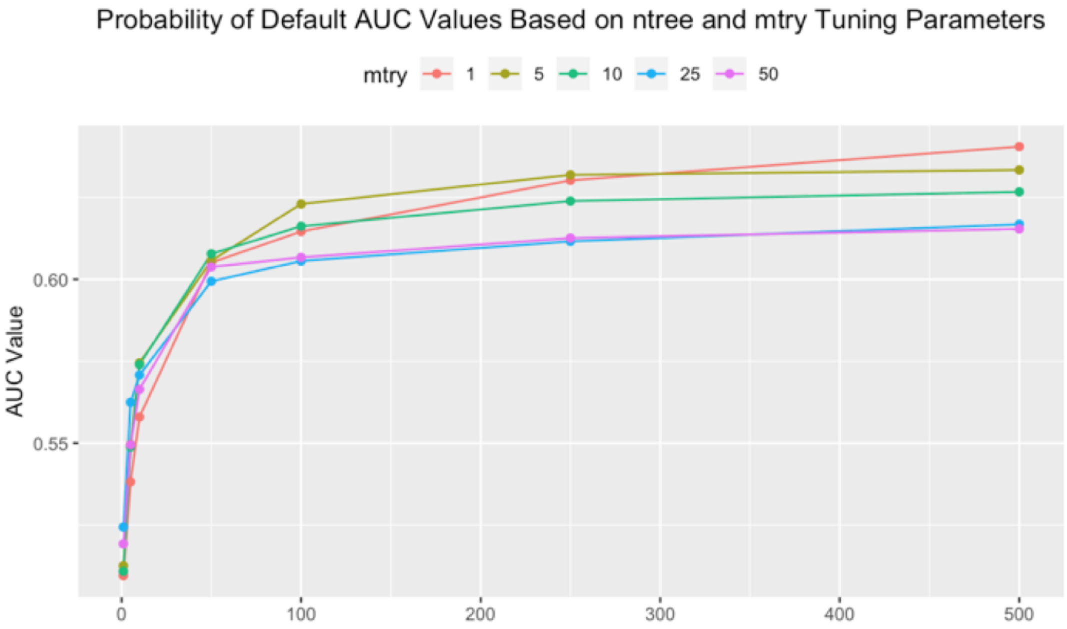


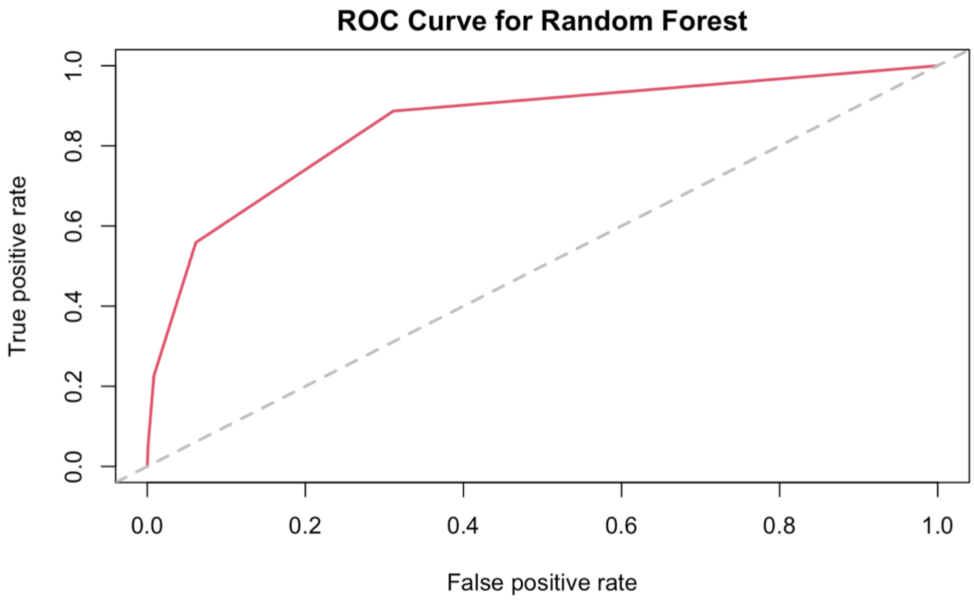
```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 14401  1141
         1   123   334

               Accuracy : 0.921
                 95% CI : (0.9167, 0.9251)
    No Information Rate : 0.9078
    P-Value [Acc > NIR] : 2.053e-09

                  Kappa : 0.3159

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9915
            Specificity : 0.2264
         Pos Pred Value : 0.9266
         Neg Pred Value : 0.7309
             Prevalence : 0.9078
         Detection Rate : 0.9001
   Detection Prevalence : 0.9714
      Balanced Accuracy : 0.6090

       'Positive' Class : 0
```

We can see that after building a Random Forest Model it produced an accuracy of 0.921 or 92.1% and sensitivity is 0.9915 or 99.15% and specificity is 0.2264 or 22.64%.

<u>RANDOM FOREST FOR PROBAILITY OF DEFAULT (PD):</u>

To assess model performance, we used AUC as a metric. As samples are chosen from the prediction model, the True positive and False positive probabilities are considered. The "RandomForest" package was utilized, and the "ntree" and "mtry" variables might be fine-tuned to further optimize the model. The increased AUC values with adjustments in those parameters are shown in Figures 4 and 5. The model was determined to be based on a "ntree" value of 500 and a "mtry" value of 5, yielding an AUC value of 0.6334.





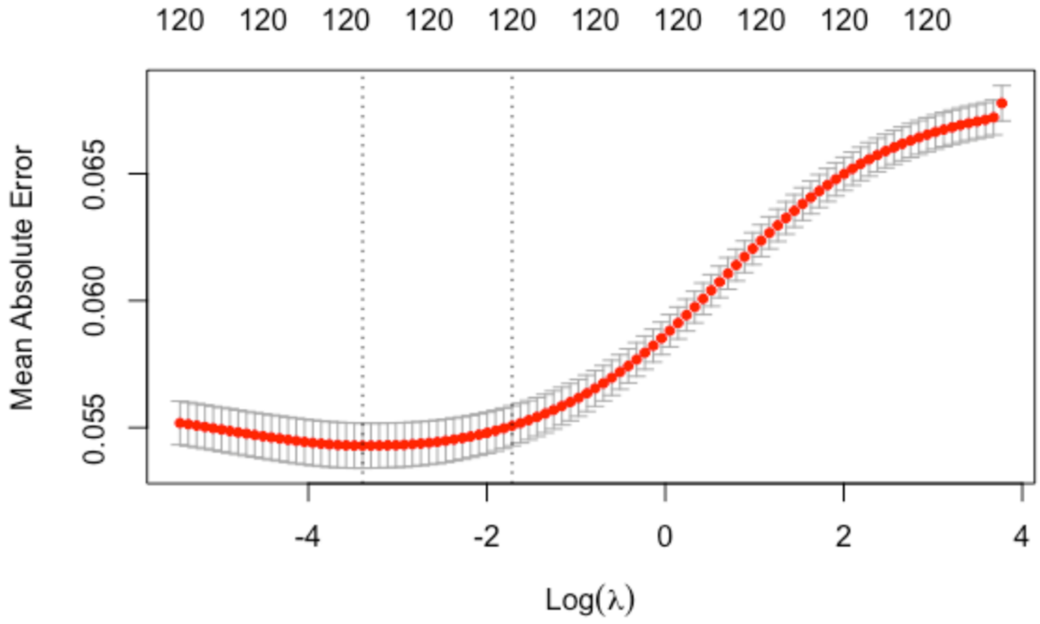
Our group concluded that if we had access to a more powerful processing power, we could have raised the performance even further and obtained better outcomes. Due to the length of time, it took our systems to execute the model, the hyper tuning procedure for the likelihood of the default model was limited.

LOSS GIVEN DEFAULT (LGD MODEL) RIDGE REGRESSION

We calculated LGD using ridge regression and used MAE (Mean Absolute Error) as a statistic to assess the model's performance. The LGD Model's MAE is 0.05, and 100 different Lambda values were tried. Out of the 100 lambda values tested, 0.0348 was chosen as the best lambda.

The Lasso regression model that was used for feature selection and regularization was evaluated based on its performance. The model's performance was assessed using the Receiver Operating Characteristic (ROC) curve, which is a graphical representation of the true positive rate (TPR) versus the false positive rate (FPR) at various threshold settings. The area under the ROC curve (AUC) is a measure of the model's ability to distinguish between positive and negative classes. The higher the AUC value, the better the model's performance.

In this project, the AUC for the Probability of Default (PD) model was 0.8521, indicating good predictive performance. The performance of the models was further validated using cross-validation, where the data was split into training and testing sets, and the model was evaluated on the testing set.



VI. INSIGHTS AND CONCLUSIONS:

The Lasso model was able to accurately predict the default risk of each customer in the given data set, with an accuracy of 0.921 or 92.1%. The model could also predict which variables had the most influence on default risk, allowing for targeted steps to limit prospective losses.

Finally, the project aim was met by developing an accurate model that forecasts the chance of default and the estimated loss given default for each client in the given data set. The financial institution can use this model to make better loan decisions, reduce losses, and improve profits.

The Lasso regression model was used to predict the Probability of Default (PD) and the Loss Given Default (LGD) for customers in the given dataset. The ROC curve and the AUC were used to evaluate the model's performance, with the PD model obtaining an excellent predictive performance with an AUC of 0.85. Cross-validation was used to validate the model's performance, and the PD model was determined to be well-calibrated.

According to the data, most in-default consumers paid 75% of their debt before defaulting, with only a few defaulting on the entire amount. This shows that the bank might potentially lessen the severity of losses by imposing stricter loan approval requirements, such as requiring a larger down payment or lowering the maximum loan amount. Furthermore, the bank could benefit from creating personalized credit risk profiles for each customer, which could aid in identifying customers who are more likely to default and hence demand more severe loan terms.

Finally, this project shows how to utilize machine learning approaches to forecast the Probability of Default and Loss Given Default for customers in the given dataset. According to the findings, the bank might potentially lower the severity of losses by imposing stricter loan requirements and creating tailored credit risk profiles for each customer.