Loan Predictions and Policies

Arham Ansari	**16830**
Abdul Rehman Admani	**21718**
Abdullah Sohail	**22369**
Usman Tahir	**16987**
Syed Muhammad Asadullah	**17420**

Group 3

Table of Contents

Executive Summary

In the contemporary business environment, from a financial institution's perspective, one of the things that matters the most is to understand clients' utilization patterns through various screening criterions and classify them accordingly. In the world of data science, such issues are dealt using automated techniques. Having said that, we have trained a machine learning model using logistic regression and decision tree analysis to predict the outcome of an applicant defaulting on their loan based on a thorough background check. Such a model will play a pivotal role for banks to filter out credit-worthy applicants from the rest of the bunch. It is easier to assume that banks can collateralize the borrowers' debts but as students of predictive analytics, it is better to assure that "last-resort" occurrences are minimized.

Dataset Description

Our dataset was about optimizing the predictive power of a borrower's credit worthiness analyzed by the machine learning model based on their concurrent and past profile to minimize the risk of future loan defaults for financial institutions. This would automate the process for banks to accept or reject an applicant. This was a supervised binary classification problem with a highly imbalanced data that was assessed through ROC AUC and Precision-Recall curves. The data was extracted from Kaggle which had 20000 rows of numerical and categorical data to begin with. The dataset consisted of the following variables:

- id: Unique ID of the loan application.
- grade: LC assigned loan grade.
- annual_inc: The self-reported annual income provided by the borrower during registration.
- short_emp: 1 when employed for 1 year or less.
- emp*length*num: Employment length in years. Possible values are - between 0 and 10 where 0 means less than one year and 10 means ten or more years.
- home_ownership: Type of home ownership.

- dti (Debt-To-Income Ratio): A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.
- purpose: A category provided by the borrower for the loan request.
- term: The number of payments on the loan. Values are in months and can be either 36 or 60.
- last*delinq*none: 1 when the borrower had at least one event of delinquency.
- last*major*derog_none: 1 borrower had at least 90 days of a bad rating.
- revol_util: Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
- total*rec*late_fee: Late fees received to date.
- od_ratio: Overdraft ratio.
- bad_loan: 1 when a loan was not paid

Problem Statement and Objective

The objective of this project is to train our machine learning model in such a way that it can predict the probability of a loan defaulting with the help of the available variables. Because loan providers are risking the amount of loan, developing a model that minimizes the risk of default by considering probabilities and best decision paths, and also maximizing profitability by providing different interest rates to the loan takers based on their probabilities of default is our aim.

Methodology

Limitations

The initial dataset has certain limitations tied with it. Firstly, the data is an assumption-based dataset which has been created to train machine learning models as a challenge. The data reflects loosely the bank data globally. However, this is considered as a limitation moving forward with the exploratory data analysis. In this particular case we assume that the data belongs to an XYZ bank of Pakistan and therefore aim to align our policies accordingly. Further limitations include missing values that are on an assumption, removed from the data considered as an anomaly. Lastly, the most important limitation of the dataset as that it constituted majorly of Factor and Categorical Variables. This became a challenge in not only running and interpreting our regression models, but also in trying to develop policies.

Feature Engineering

We have engineering two new variables to test the significance of the variables upon our results and to understand why or why not there is an impact of our variables on the loan defaults.

The variables made are described as follows

Above Average Debt to Income

Average Debt to Income was a **FACTOR VARIABLE** derived from pre-existing DEBT to INCOME RATIO. In excel, the IF command was used to create a variable which returned 1 if a certain ID of a loan receiver had more than the average Debt to Income ratio than that of the average of our entire data set. If the statement was false, the result returned 0. This binary factor variable would later help us train a model and to find the best possible solution for loan providers.

Below Average Income

Another **FACTOR VARIABLE** derived from the pre-existing Income variable. The IF command was used to create a variable which returned 0 if a certain ID of a loan receiver had more than the average Income ratio than that of the average of our entire data set. If the statement was false, the result returned 1.

Debt

Another **Numeric Variable** was created with the formula **(annual income * dti)/100.** Since the data for debt taken was missing, it was produced by multiplying income and debt to income ratio and then dividing it by 100 to return the debt amount taken against an id.

Data Cleaning

In the initial stage of assessing the data, data cleaning was performed in two steps on Excel.

Anomalies

Anomalies in the data were such that the data contained 0 values in Numeric and Integer Variables which was treated as **Missing Data.** The missing data rows were entirely removed from the dataset to avoid misrepresentation of data. The variables that were considered in this case were **Debt to Income Ratio** which had 154 missing values.

Outliers

Similarly, considering the mean and the relative **standard deviation** of maximum of minimum values of a particular variable it was also identified that some variables contained outliers. The top 2.5% and the bottom 2.5% percentile values were treated and removed as outliers. 851 outliers were removed.

Result and Data Splitting

The resulting dataset had 18994 out of 20000 values remaining. The output data was divided into train and test data. The data was first randomized on R Studio and then the data was weight

67 and 33% for Train and Test Data respectively.

Type	Percentage Data	No. of Values
Train	67%	12,726
Test	33%	6,270

Trend Analysis

For the non-binary factor variables that we consider as categorical variables with more than 2 levels, which include Grade, Purpose and Home Ownership, we developed a trend analyses and plotted their graphs against the bad loans. The graphs and the tables can be seen in the appendix. In the initial phase of our study, this step was essential because of the relatively larger number of variables that were categorical. Studying any noticeable trends in these categories also helped us align our regression model with the same trends. As an example, the default percentages decreased as the Grade Increased. Therefore, a development in our study was made that the grades could be used as tiers of loan borrowers. In further research this data can also be used to create a criteria to understand tiers and to distribute and minimize the risk throughout the grades.

Similarly, home ownership type and purpose trends help us understand the effect of different categories on our results and see if our predictions and model are aligned.

Variable Importance

The variables used in our model were all the given models in the dataset. Because of an assumption-based model and the limitation of data reliability, no new variables were extracted from other sources. It was discovered, by keeping Bad Loans as the Y variable that the most significant variables for us were;

Backward Stepwise Generalized Linear Model in R

The first model created on R Studio was the Backward Stepwise GLM model. For this model, the Y variable used was Bad Loans which in itself was a factor variable for our data. Therefore, in order to run the regression, the categorical and factor variables were converted to factor variables using the following code:

as.factor(dataname$variablename)

The following variables were converted to Factors: **Grade, short_emp, emp*length*num, home_ownership, purpose, term, last*delinq*none, last*major*derog_none, bad_loan, avg_income** and **avg_dti**. The backward stepwise regression model returned variables such as A category of the grade, short employment 1, employment length and term months 30. The appendix includes the detailed summary of GLM Stepwise Model and the returned significant variables. However, another essential step in our working was before the stepwise glm model. The first GLM model for trained data showed us that out of 12 purposes, 10 were insignificant. Therefore, we manually combined those purpose in a new category named as “Other” under purpose. The other two significant types were Debt Consolidation and Credit Card.

Predictions from the Stepwise Regression model were then extracted and used to plot Recall, Precision and F1 scores against different thresholds.

Classification of a problem

Confusion Matrix

		Prediction	
		0	**1**
Actual	**0**	**TN**	**FP**
	1	**FN**	**TP**

		Prediction	
		0	**1**
Actual	**0**	**No default / predict no default**	**No default / predict default**
	1	**Loan default / predict no default**	**Loan default / predict default**

One of these 2 Problems may arise

Recall: This is appropriate when we aim to **minimize false negatives**.

Precision: This is appropriate when we aim to **minimize false positives.**

Here, the case depicts the Recall Problem. To minimize the risk of loan default, given the fact that the model predicts that it will not default, we need to minimize fall negatives. Minimizing fall negatives can be done when Recall is increased and maximized.

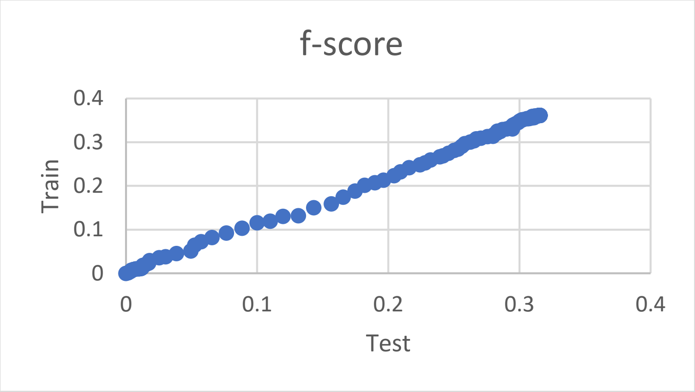
Hence, we are working for an excellent prediction of positive class and maximizing Recall. This can, however, be challenging, as often increase in recall often comes at the expense of decreases in predictions.

The F-score is used here which instead of picking one measure or the other, choose a new metric that combines both precision and recall in one score.

*Formula to calculate F-score: (2*Precision*Recall)/(Precision+Recall).*

This is the harmonic mean of Recall and Precision, and we worked on maximizing it. The highest score for the train dataset can be found at the **0.05 threshold** and the highest score for the test dataset can be found at the **0.06 threshold**.

Train/ Test Relation



Here the plot simply shows that the movement of the train data and test data is aligned together. Hence, the division of data into train and test data is justifiable.

Policy based on STEPWISE Model

The ROC and PR Curve area was relatively better than decision tree model therefore we would recommend to use STEPWISE model for better predictions.

Based on the interpretation of Stepwise model, we generated predictions for Bad loan. We used following matrix:

	Acutal 0 Pred 0	Actual 1 Pred 1	Actual 1 Pred 0	Actual 0 Pred 1			
Threshold	**TN**	**TP**	**FN**	**FP**	**Recall**	**Precision**	**f-score**

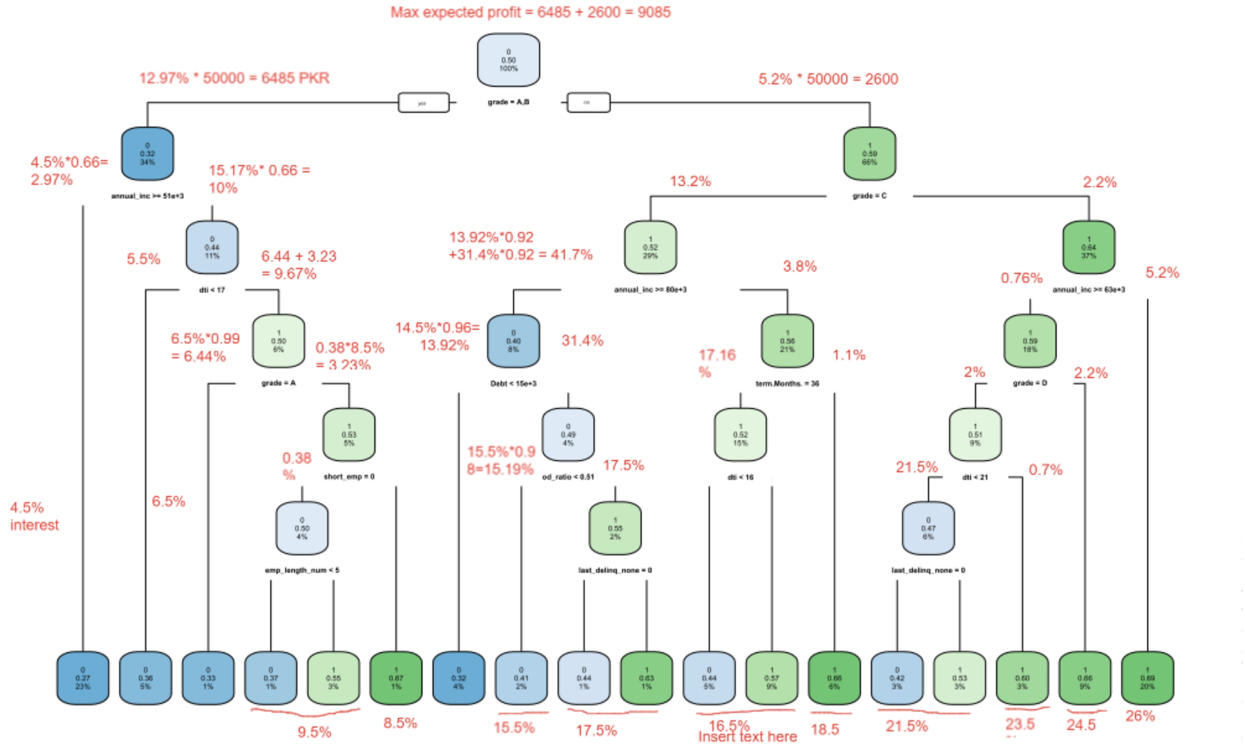
We used actual data and prediction to make a threshold from 0 to 1 and Recall, precision and f-score was calculated based on the result. Based on the data, we will give loan to the individuals where the prediction is 0 (which represents no default), therefore True negative will give us profit and false negative will give us loss of Principal minus Forced value of security.

The interest from Pakistan Bank lending rate was collected, which was 8.5%. Similarly, it was assumed that the bank will hold a security against loan. However, as the borrower will default, the back will try to recover its principal amount by selling the security. The bank will try to recover its amount on urgent basis thus it is assumed 80% of the value of security will be recovered by the bank.

Based on the data above we use 8.5% of Principal as Profit on True Negative and 20% loss of Principal value on false negative. This data gave us highest profit at 0.95 and above threshold.

Since the amount of loan is not given in the Data and it was not possible for us to collect this data from any other source, we assumed Profit/Loss in terms of Percentage. The percentage can be multiplied by the amount of loan in order to obtain Total Profit/Loss on Loan.

Policy based on Decision Tree



We have worked bottom to top to calculate the maximum expected profitability. The assumptions are that the amount of debt from each classified grader (A, B, C, D) is PKR 50000 which has remained constant throughout the calculations. Moreover, the duration is one year for all the loans. Since this dataset was fabricated, we had to assume the interest rates on subjective basis. The current KIBOR rate is 14.5%. We have started from the left by assigning 4.5% interest rate (KIBOR-10%) to the most credit-worthy borrowers. For grade B, the policy rate is 9.5%. Likewise, for grade C it is the KIBOR rate of 14.5%. For grade D and beyond, it is greater than 19.5%. Those borrowers who were unable to pass the Boolean checks and had to go through more than one screening criterion were charged an additional 1% interest rate. In short, as the number of screening tests grew, the cost of borrowing increased for the borrowers as interest rate increased. All those splits from the decision nodes led to sub-nodes and penultimately to the terminal nodes from where different expected probabilities were multiplied with the applied respective interest rates and then worked all the way up to the branches till the primary decision nodes. These final expected interest rates were then multiplied with the constant debts of fifty thousand to arrive at the optimal solution conjured by the best path of decision tree.

It should be taken into consideration that when we **printed** the decision trees on R encoded with the variables Rpart and Rpart2, the model was calculating incorrect yprobs. It looked like an algorithmic error. Here is an example:

```
n= 12726


node), split, n, loss, yval, (yprob)
      * denotes terminal node
 1) root 12726 10152 0 (0.5008850 0.4991150)
   2) grade=A,B 5224  2228 0 (0.6768673 0.3231327)
     4) annual_inc>=51202 3709  1248 0 (0.7313240 0.2686760) *
     5) annual_inc< 51202 1515   980 0 (0.5644444 0.4355556)
      10) dti< 17.3 744   360 0 (0.6449704 0.3550296) *
      11) dti>=17.3 771   616 1 (0.4983819 0.5016181)
        22) grade=A 135    60 0 (0.6666667 0.3333333) *
```

In the node labelled 2, when the loss value is divided by n, it should give a proportion of 0.426 but rather it is incorrectly inscribed as 0.323. This is consistent across all the nodes. Thus, unable to find the reason for this miscalculation, we decided to go along with the decision tree that we made and based on that, discovered the best path. The purpose to inform about this unusual discrepancy is to prevent readers being misled if they decided to cross-check their decision tree with ours using this same dataset. *This decision tree should only be interpreted for understanding purposes and not as a means of benchmark due to the aforementioned algorithmic error.*

Appendix

Contribution Table

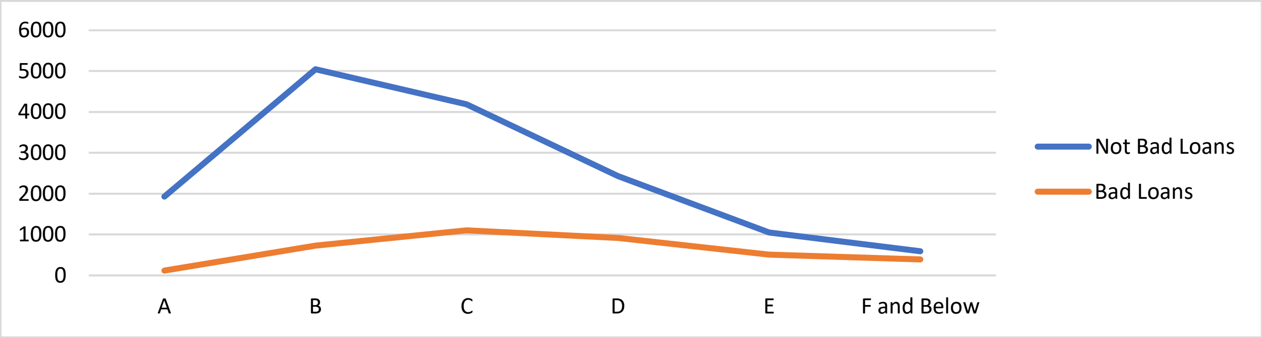
Name	Contribution
Usman Tahir	Creating New Variables GLM Phase 2 and All related workings Interpretations of GLM Policy Implementations Decision tree Data cleaning Cross validation Data balancing
Abdullah Sohail	Creating New Variables GLM Phase 2 and All related workings Interpretations of GLM Policy Implementations Decision Trees Data cleaning Data balancing
Syed Asadullah	Report Development Trend Analysis Data cleaning GLM model
Arham Ansari	Report Development Trend Analysis Data cleaning Decision tree

Abdul Rehman Admaani	Glm recall precision f score Data cleaning Policy Implementations of Decision Tree Models Variable Testing Intrepretation of glm model

Trend Analysis Tables

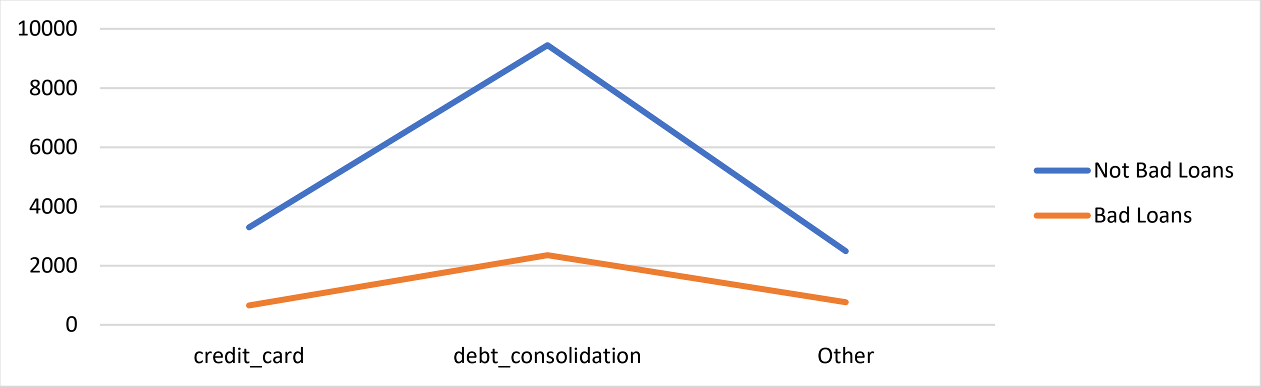
Grade Categories

Count of bad_loan	**Column Labels**			
Row Labels	**Not Bad Loans**	**Bad Loans**	**Grand Total**	
A	1926	119	2045	5.8%
B	5046	727	5773	12.6%
C	4186	1104	5290	20.9%
D	2428	916	3344	27.4%
E	1048	511	1559	32.8%
F and Below	592	391	983	39.8%
Grand Total	**15226**	**3768**	**18994**	



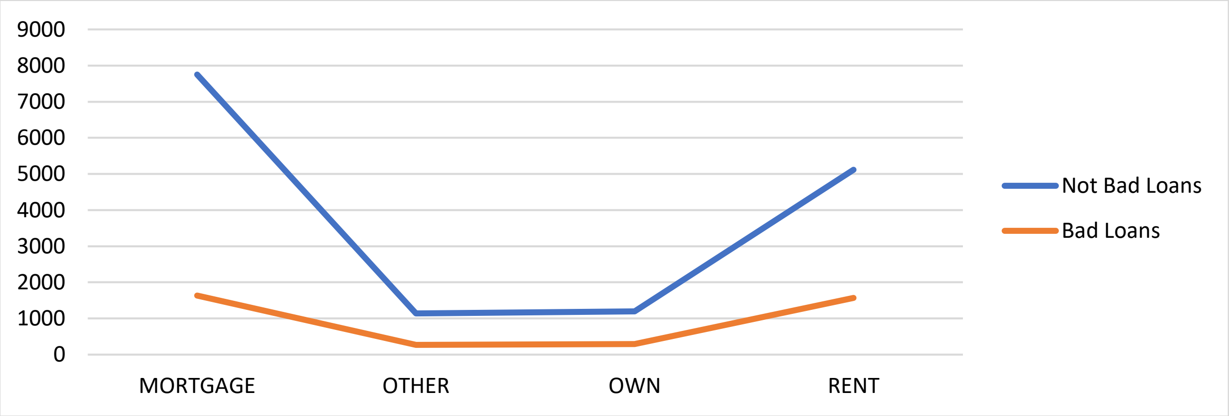
Purpose Categories

Count of bad_loan	**Column Labels**			
Row Labels	**Not Bad Loans**	**Bad Loans**	**Grand Total**	
credit_card	3290	655	3945	16.6%
debt_consolidation	9448	2353	11801	19.9%
Other	2488	760	3248	23.4%
Grand Total	**15226**	**3768**	**18994**	

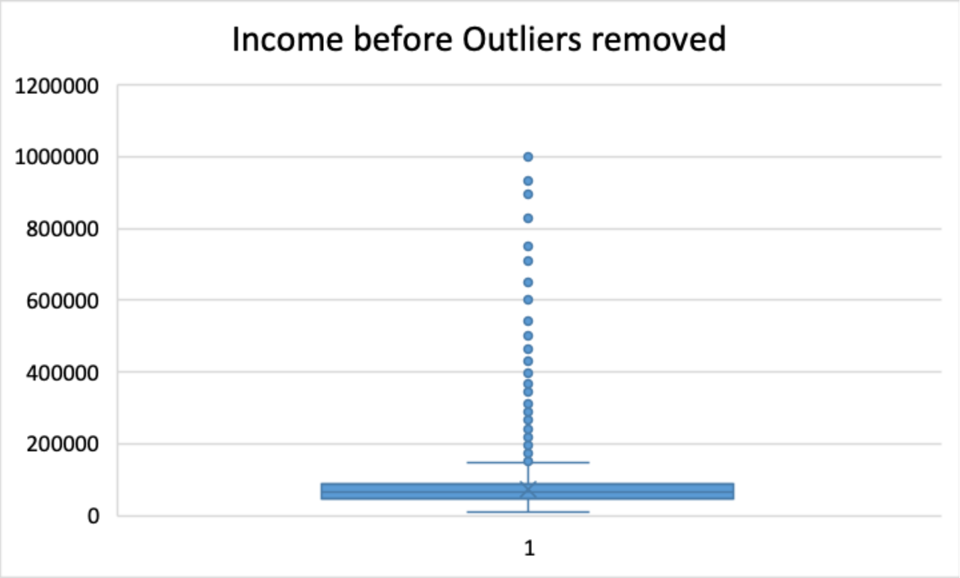


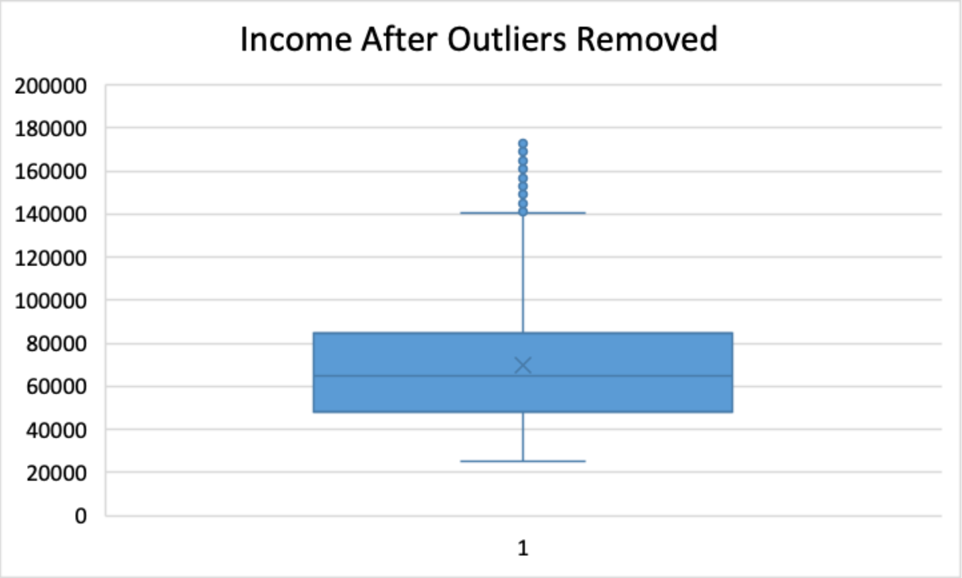
Home Ownership Category

Count of bad_loan	Column Labels			
Row Labels	Not Bad Loans	Bad Loans	Grand Total	
MORTGAGE	7751	1634	9385	17.4%
OTHER	1140	269	1409	19.1%
OWN	1198	289	1487	19.4%
RENT	5119	1567	6686	23.4%
Grand Total	15208	3759	18967	



Box Plots





GLM R Script

```
setwd("G:/IBA/Courses resources/Analytical Approach to Marketing Decisions/Project3")
data <- read.csv(file.choose(), header=TRUE)
colnames(data)
data<- data[,-c(1,2)]
colnames(data)

set.seed(123)
```

```
data <- data[sample(1:nrow(data)),]
rownames(data) <- 1:nrow(data)

data$grade <- as.factor(data$grade)
data$short_emp <- as.factor(data$short_emp)
data$home_ownership <- as.factor(data$home_ownership)
data$purpose <- as.factor(data$purpose)
data$term.Months. <- as.factor(data$term.Months.)
data$last_delinq_none <- as.factor(data$last_delinq_none)
data$last_major_derog_none <- as.factor(data$last_major_derog_none)
data$Above.Avg.Income <- as.factor(data$Above.Avg.Income)
data$Below.Avg.DTI <- as.factor(data$Below.Avg.DTI)
data$bad_loan <- as.factor(data$bad_loan)

train <- data[1:12726, ]
test <- data[12727:18994, ]

weights <- ifelse(train$bad_loan == "0", 1, 4)

GLM <- glm(bad_loan~.,data = train,family = 'binomial',weights = weights)
Stepwise <- step(GLM)
TrainPred <- Stepwise$fitted.values
TestPred <- predict(Stepwise,test,type = "response")
Pred <- c(TrainPred,TestPred)
write.csv(Pred,file="Pred2.csv")

library(PRROC)
# Train
ROC <- roc.curve(scores.class0 = TrainPred,weights.class0 =
as.numeric(as.character(train$bad_loan)),curve = T)
print(ROC)
plot(ROC)
```

```
PRCURVE <- pr.curve(scores.class0 = TrainPred,weights.class0 =
as.numeric(as.character(train$bad_loan)),curve = T)
print(PRCURVE)
plot(PRCURVE)
#Test
ROC <- roc.curve(scores.class0 = TestPred,weights.class0 =
as.numeric(as.character(test$bad_loan)),curve = T)
print(ROC)
plot(ROC)
PRCURVE <- pr.curve(scores.class0 = TestPred,weights.class0 =
as.numeric(as.character(test$bad_loan)),curve = T)
print(PRCURVE)
plot(PRCURVE)

# Cross Validation

i = 10
rand <- runif(n = 18994)
for (i in 1:10)
test_indices <- rand <= i / 10 & rand > (i - 1) / 10
train_cv <- data[!test_indices,]
test_cv <- data[test_indices,]
weights_cv <- ifelse(train_cv$bad_loan == "0", 1, 4)
model_cv <-
glm(bad_loan ~ .,data = train_cv,family = "binomial",weights = weights_cv)
STEP_cv <- step(model_cv)
pred_cv <-predict(STEP_cv, newdata = test_cv, type = "response")
roc_cv <- roc.curve(scores.class0 = pred_cv,weights.class0 =
as.numeric(as.character(test_cv$bad_loan)),curve = T)
print(roc_cv)
```

Stepwise GLM Summary

```
Call:
```

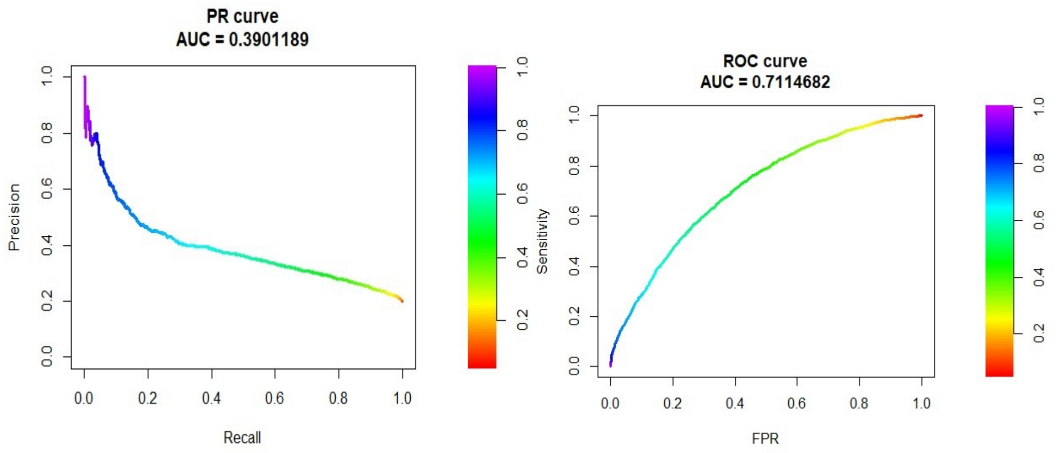
```
glm(formula = bad_loan ~ grade + annual_inc + short_emp + home_ownership +
    dti + purpose + term.Months. + last_delinq_none + last_major_derog_none +
    revol_util + total_rec_late_fee + Debt + Above.Avg.Income,
    family = "binomial", data = train, weights = weights)

Deviance Residuals:
RPART2 #decision tree
1) root 12726 10152 0 (0.5008850 0.4991150)
    2) grade=A,B 5224  2228 0 (0.6768673 0.3231327)
      4) annual_inc>=51202 3709  1248 0 (0.7313240 0.2686760) *
      5) annual_inc< 51202 1515   980 0 (0.5644444 0.4355556)
       10) dti< 17.3 744   360 0 (0.6449704 0.3550296) *
       11) dti>=17.3 771   616 1 (0.4983819 0.5016181)
         22) grade=A 135    60 0 (0.6666667 0.3333333) *
         23) grade=B 636   496 1 (0.4696970 0.5303030)
           46) short_emp=0 534   424 0 (0.5023474 0.4976526)
             92) emp_length_num< 4.5 172    88 0 (0.6302521 0.3697479) *
             93) emp_length_num>=4.5 362   278 1 (0.4527687 0.5472313) *
           47) short_emp=1 102    68 1 (0.3333333 0.6666667) *
    3) grade=C,D,E,F and Below 7502  5521 1 (0.4106359 0.5893641)
      6) grade=C 3579  2813 1 (0.4786456 0.5213544)
       12) annual_inc>=79650 1098   636 0 (0.5961905 0.4038095)
         24) Debt< 15278.13 580   240 0 (0.6842105 0.3157895) *
         25) Debt>=15278.13 518   396 0 (0.5141104 0.4858896)
           50) od_ratio< 0.5063618 252   148 0 (0.5922865 0.4077135) *
           51) od_ratio>=0.5063618 266   204 1 (0.4513274 0.5486726)
            102) last_delinq_none=0 132    88 0 (0.5555556 0.4444444) *
            103) last_delinq_none=1 134    94 1 (0.3700787 0.6299213) *
       13) annual_inc< 79650 2481  1874 1 (0.4356113 0.5643887)
         26) term.Months.=36 1818  1429 1 (0.4787270 0.5212730)
           52) dti< 15.83 716   472 0 (0.5588785 0.4411215) *
           53) dti>=15.83 1102   831 1 (0.4339426 0.5660574) *
         27) term.Months.=60 663   445 1 (0.3378891 0.6621109) *
      7) grade=D,E,F and Below 3923  2708 1 (0.3578224 0.6421776)
       14) annual_inc>=62550 2000  1477 1 (0.4138414 0.5861586)
         28) grade=D 1077   853 1 (0.4877073 0.5122927)
           56) dti< 20.655 786   576 0 (0.5270936 0.4729064)
            112) last_delinq_none=0 429   260 0 (0.5833333 0.4166667) *
            113) last_delinq_none=1 357   278 1 (0.4680135 0.5319865) *
           57) dti>=20.655 291   211 1 (0.3973635 0.6026365) *
         29) grade=E,F and Below 923   624 1 (0.3428571 0.6571429) *
       15) annual_inc< 62550 1923  1231 1 (0.3078270 0.6921730) *
>
```

```
    Min       1Q   Median       3Q      Max
-3.4797  -1.2276  -0.9489  -0.5949   4.7342
```

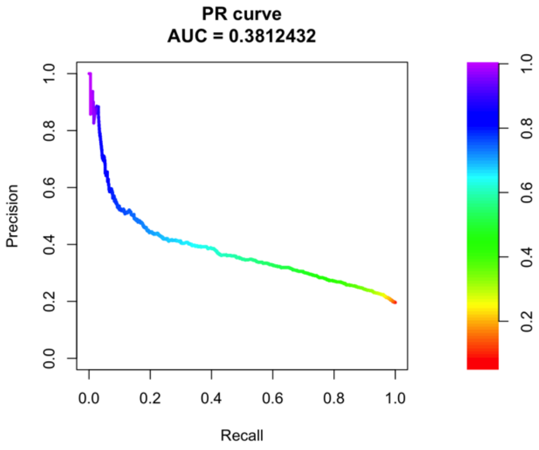
```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -1.883e+00  1.454e-01 -12.951  < 2e-16 ***
gradeB                    6.633e-01  7.258e-02   9.139  < 2e-16 ***
gradeC                    1.129e+00  7.406e-02  15.241  < 2e-16 ***
gradeD                    1.405e+00  7.730e-02  18.180  < 2e-16 ***
gradeE                    1.566e+00  8.859e-02  17.677  < 2e-16 ***
gradeF and Below          1.750e+00  1.000e-01  17.499  < 2e-16 ***
annual_inc               -8.570e-06  1.447e-06  -5.924 3.15e-09 ***
short_emp1                2.823e-01  4.771e-02   5.918 3.27e-09 ***
home_ownershipOTHER       1.125e-01  5.995e-02   1.877  0.06053 .
home_ownershipOWN        -4.640e-02  5.891e-02  -0.788  0.43092
home_ownershipRENT        2.749e-01  3.450e-02   7.969 1.60e-15 ***
dti                       2.407e-02  5.152e-03   4.672 2.98e-06 ***
purposedebt_consolidation  6.900e-02  3.971e-02   1.738  0.08225 .
purposeOther              3.824e-01  5.212e-02   7.337 2.19e-13 ***
term.Months.60            3.914e-01  3.922e-02   9.979  < 2e-16 ***
last_delinq_none1         7.089e-02  3.080e-02   2.301  0.02136 *
last_major_derog_none1   -1.643e-01  1.052e-01  -1.562  0.11828
```

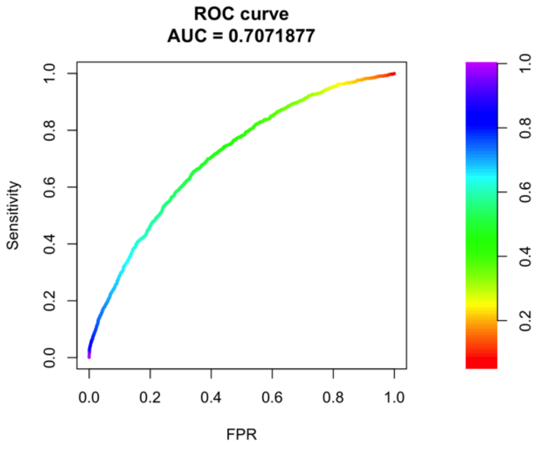
```
revol_util              4.767e-03  7.273e-04   6.554 5.59e-11 ***
total_rec_late_fee        1.033e-01  8.234e-03  12.551  < 2e-16 ***
Debt                    1.395e-05  7.320e-06   1.906  0.05670 .
Above.Avg.Income1         1.419e-01  5.236e-02   2.710  0.00672 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28197  on 12725  degrees of freedom
Residual deviance: 25198  on 12705  degrees of freedom
AIC: 25240

Number of Fisher Scoring iterations: 5
```

Stepwise GLM ROC and PR Curves for Train Data

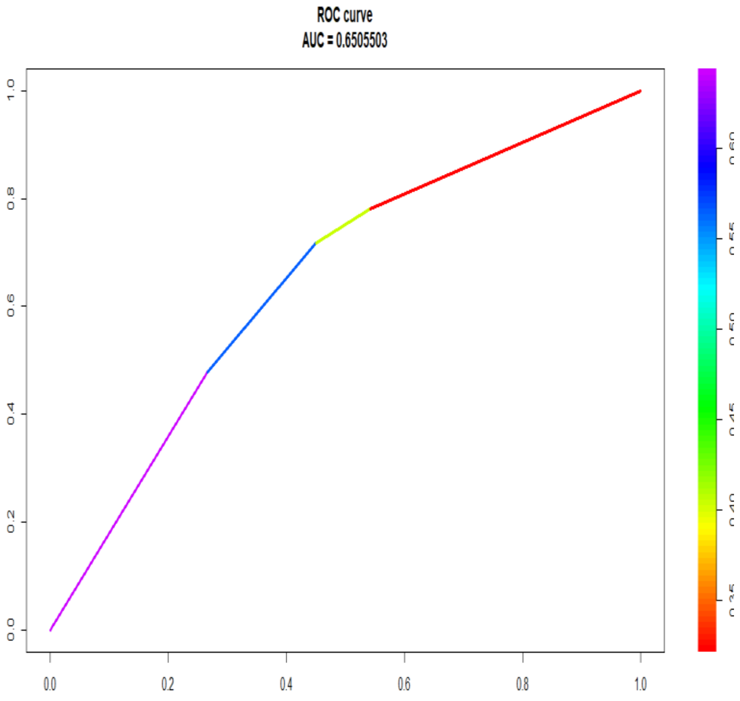


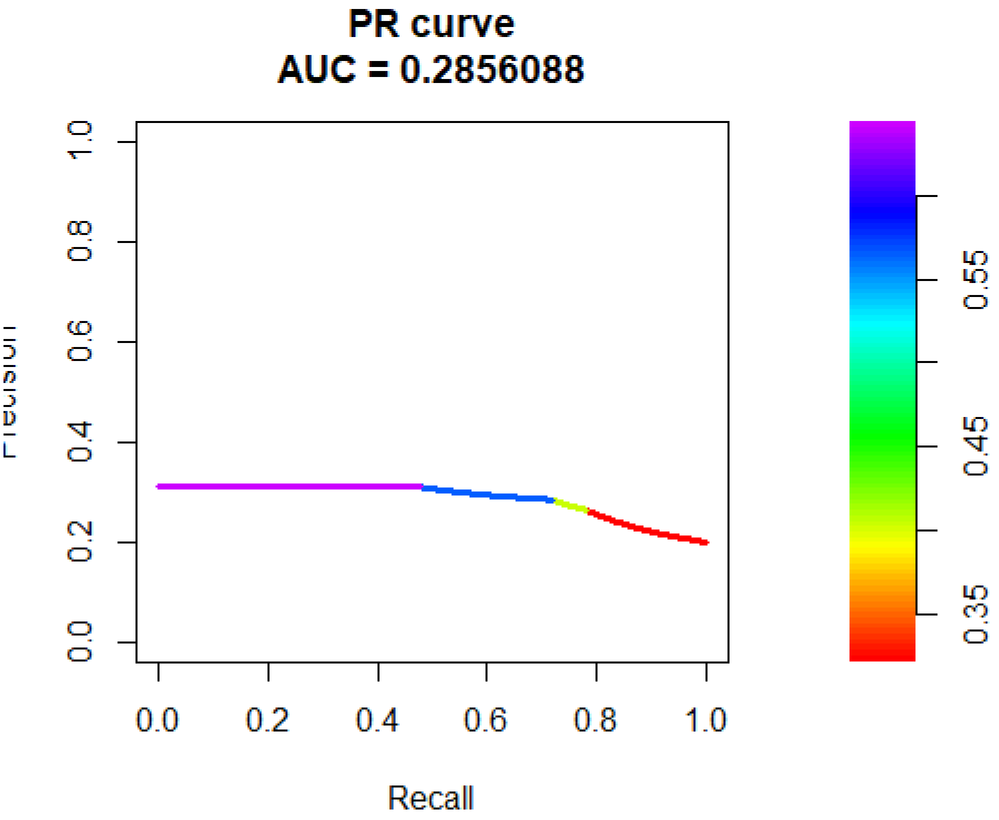
Stepwise GLM ROC and PR Curves for Test Data



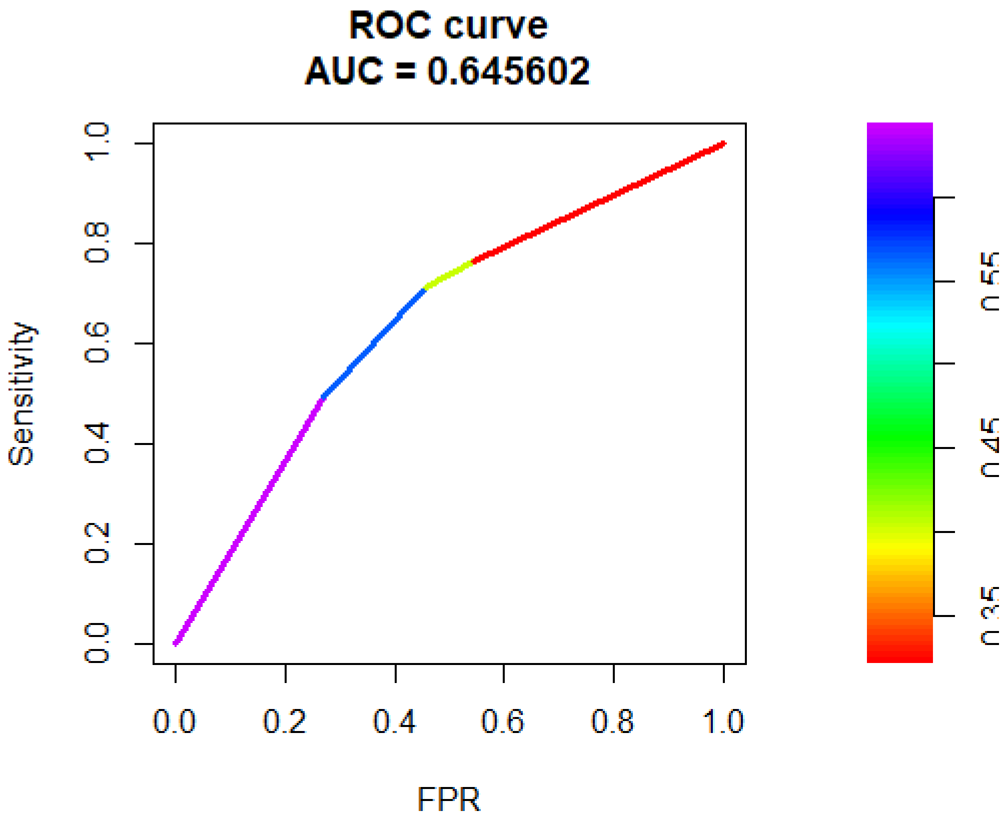


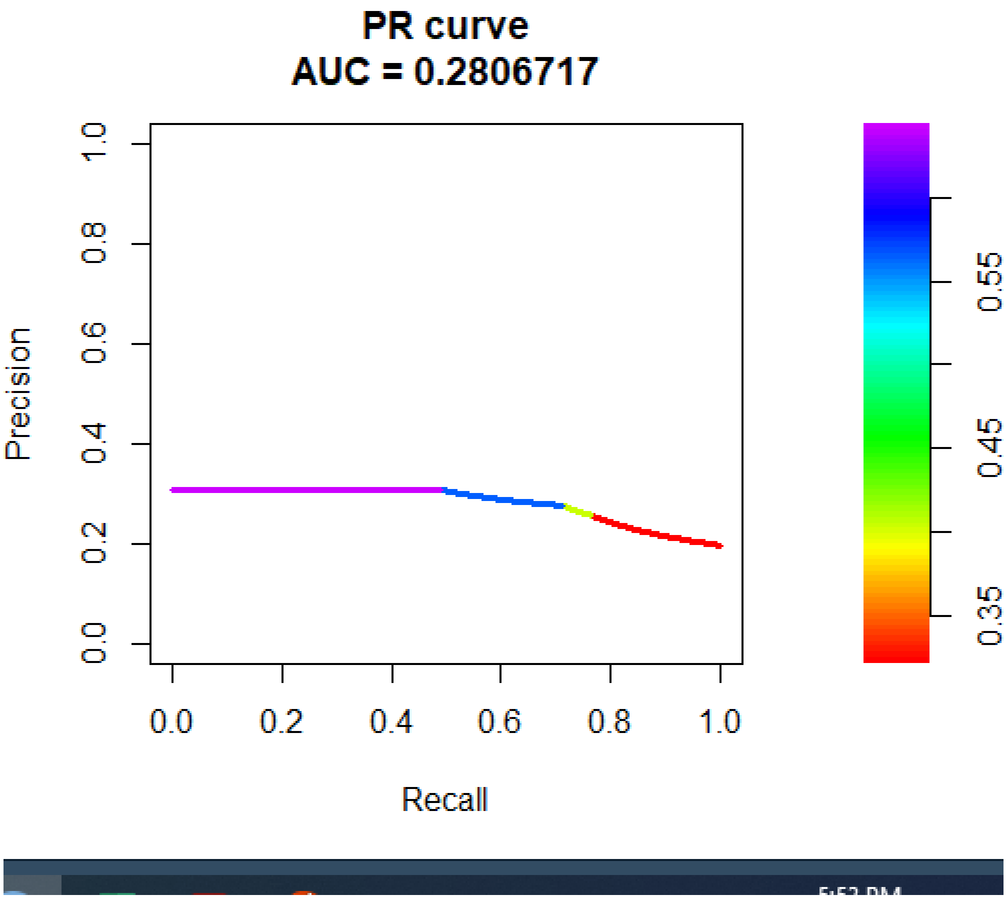
Decision Tree ROC and PR curves for Train Data




PR curve
AUC = 0.2856088
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.55
0.45
0.35

Dec Tree ROC and PR for test data




PR curve
AUC = 0.2806717
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.55
0.45
0.35