Predicting Loan Defaults with Logistic Regression

Mohan Rajendran

March 2, 2019

Introduction

The purpose of this project is to predict which applicants are likely to default on their loans using Logistic Regression. The dataset includes 30 variables for 50,000 loans. This dataset will be cleaned and any missing values will be imputed before creating the predictive model. The mdoel will help the banks to predict the outcome of the loan therby reducing the risk associated with loan defaults.

Data Preparation

As part of data preparation we will prepare response variable, clean the data, perform feature engineering and impute missing values.

Response Variable

The response variable is created based out of status variable. Any loans with status as Fully Paid are considered 'Good' and the ones with status as Charged off or Default are considered 'Bad'. Any loans with status other than the three mentioned above are removed from the data.

```
Loans_df <- Loans_df %>% filter(status %in% c('Charged Off', 'Fully Paid',
'Default'))
Loans_df <- Loans_df %>%
  mutate(responseVar = as.factor(ifelse(status == 'Fully Paid', 'Good',
'Bad')))
```

Data Cleaning

We identified some of the variables to be either irrelevant (employment) or information that is not known prior to loan provision(totalPaid). Though loadID variable can be ignored we want to keep this for time being so as to facilitate easy row manipulation.We also identified a row with loanID 656728 which is invalid and removed it from the data.

Feature Engineering

Now we can proceed to perform feature engineering on two of the variables 'status' and 'reason'. FOr the purpose of this project, we merged the status 'Default' with 'Charged off'. Similarly for the reason variable we merged 'car' and 'house' to form a new category 'Asset_Purchase' and 'renewable_energy' and 'wedding' to category 'other'. Though

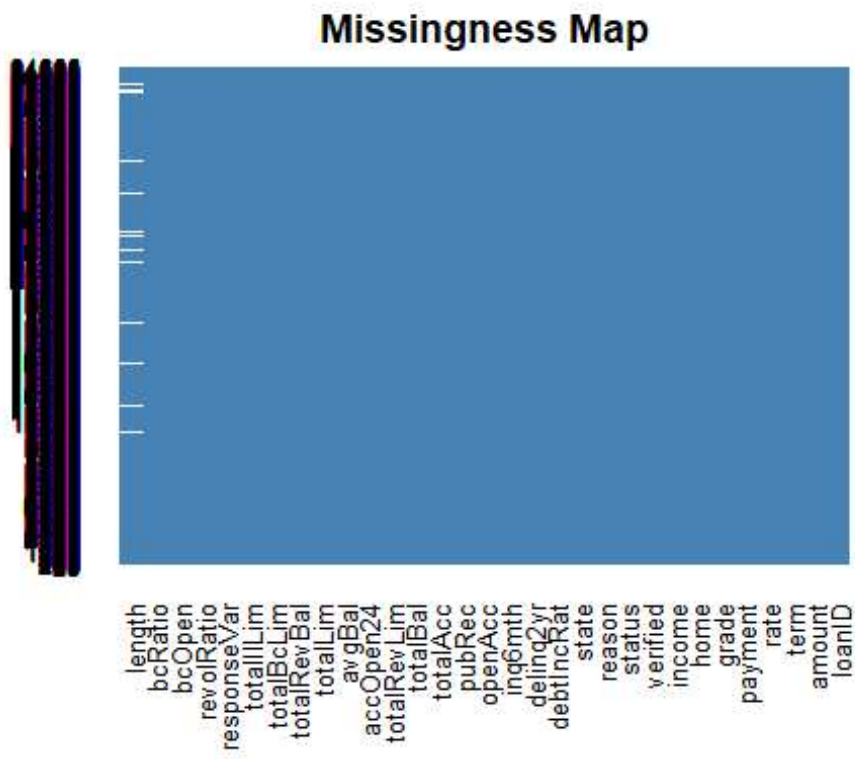
vacation and emergency can be merged with other, for time being we prefer not to since we believe them to have a influence on loan repayment capability.

```
Loans_df <- Loans_df %>%
  mutate(status = ifelse(status == 'Default', 'Charged Off', status))

Loans_df <- Loans_df %>%
  mutate(reason = ifelse(reason == 'car' | reason == 'house',
'Asset_Purchase',
                         ifelse(reason == 'renewable_energy' | reason ==
'wedding', 'other', reason)))

Loans_df <- Loans_df %>%
  mutate_if(sapply(Loans_df, is.character), as.factor)
```

Missing Value Imputation

Upon analyzing the summary of the Loan_df dataframe we identified four variables length, bcRatio, bcOpen, revolRatio to have missing values/NA in them.This is confirmed by the below missmap plot.

```
missmap(Loans_df, col=c('white', 'steelblue'), legend=FALSE, y.cex = 0.8,
x.cex = 0.8,margins = c(5, 5))
```



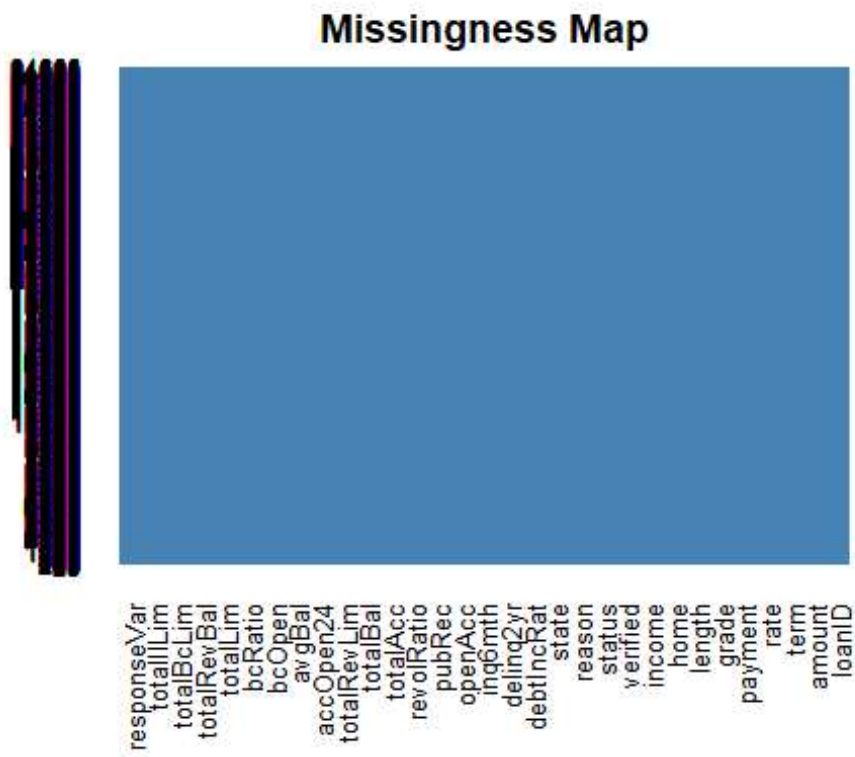
To impute the missing values, MICE imputation method is used. For the quantitative variables pmm(predictive mean matching) method is used while for the categorical

variable 'length' polytomous regression method is used. Loan ID variable is not included in the predictor variable list.

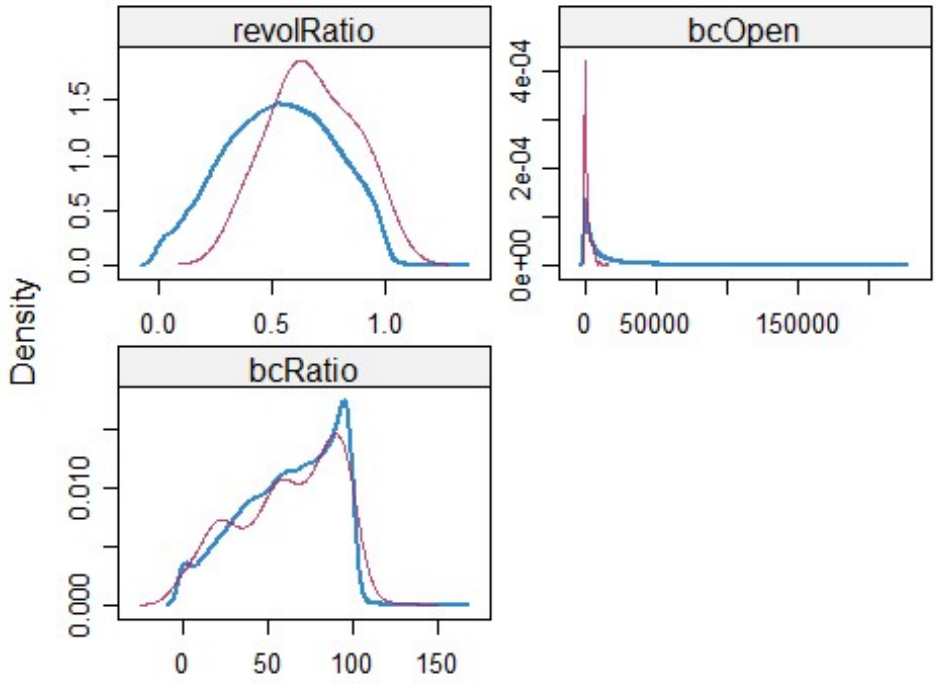
```
init = mice(Loans_df, maxit=0)
meth = init$method
predM = init$predictorMatrix


meth[c("bcRatio")]="pmm"
meth[c("bcOpen")]="pmm"
meth[c("revolRatio")]="pmm"
meth[c("length")]="polyreg"

predM[, c("loanID")]=0

set.seed(1984)
ImpPMMMethod <- mice(Loans_df, method = meth, predictorMatrix=predM, m=1,
maxit=5)

##
##  iter imp variable
##   1   1  length  revolRatio  bcOpen  bcRatio
##   2   1  length  revolRatio  bcOpen  bcRatio
##   3   1  length  revolRatio  bcOpen  bcRatio
##   4   1  length  revolRatio  bcOpen  bcRatio
##   5   1  length  revolRatio  bcOpen  bcRatio

Loans_Imputed_df <- complete(ImpPMMMethod,1)
```

The density plot and mismap plot is plotted again to verify whether all the missing values are imputed and are inline with existing values

```
missmap(Loans_Imputed_df, col=c('white', 'steelblue'), legend=FALSE, y.cex =
0.8, x.cex = 0.8,margins = c(5, 5))
```




```
densityplot(ImpPMMMethod)
```



Variable Transformations

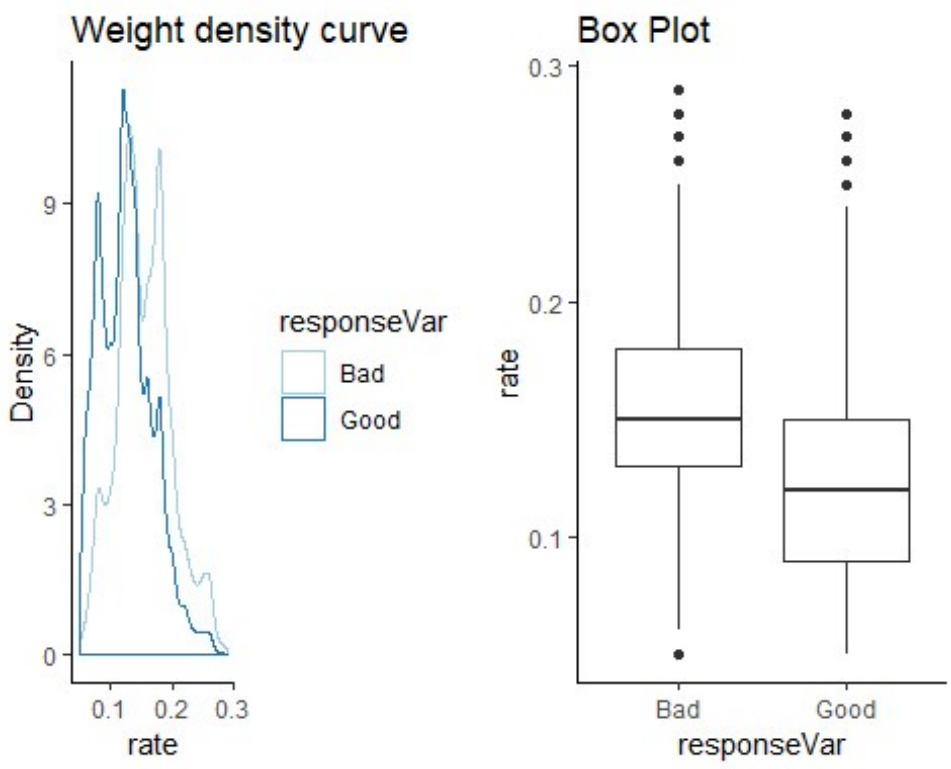
As a final step we will perform some transformation on the variables to eliminate the skewing in the data which is one of the prerequisite for the Logistic Regression. Since its possible for the value to be 0 in most of the variables we decided to perform cube root transformation. By exploratory data analysisfunction created for the project(below), we identified most of the income, balance and Limit variables to be skewed and hence transformation is performed on these variables.

```
CubeRootTransformation <- function(df, colNamesVector){
  for (c in colNamesVector){
   df[, c] <-  df[, c]^(1/3)
  }
  return(df)
}

tName <- c('payment', 'income', 'totalBal', 'totalRevLim', 'avgBal',
'bcOpen', 'totalLim', 'totalRevBal', 'totalBcLim', 'totalIlLim')

Loans_Imputed_df <- CubeRootTransformation(Loans_Imputed_df, tName)
```

Exploratory Data Analysis

The density plot and box plot are plotted for the quantitative variables and bar plots are computed for categorical variables against response variable. Upon analysing the plots its found rate has a huge impact on the outcome of the loans. Loans that are fully paid off appear to have lower rate.

```
PlotQuantitative(Loans_Imputed_df, 'rate', 'responseVar')
```



Logistic Model

We begin our Logistic Regression by splitting our data into Train and Test data in the 80:20 ratio. We also removed loanID, status from the train dataset and also the state variable so as not to introduce any location based bias.

We then simulated the three following models Full Model with all the predicted variables, Backward Step wise Elmination approach and Forward Stepwise ELimination approach. Analysing the AIC we found there is no significant difference between the three and the same is confirmedd in the later part of analysis where the accuracy for various cut off values lined up exactly same for all three model. So from this point we will focus our analysis only on Full Model.

NOTE: WE DIDNT REMOVED ANY VARIABLES OTHER THAN THE ONES REMOVED PREVIOUSLY FROM THE PREDICTOR VARIABLE LIST. THOUGH WE CAN REMOVE CERTAIN VARIABLES BASED ON THE VIF VALUES THIS STEP IS OMITTED IN FULL MODEL SINCCE IN THE INSTRUCTIONS IT IS MENTIONED TO FIT MODEL WITH ALL PREDICTOR VARIABLES. DEPENDING ON THE COMMENTS WE CAN CORRECT THIS IF NEEDED IN FINAL SUBMISSION

```
Full_Model <- glm(responseVar ~ ., data = Loans_Train_df, family =
"binomial")
summary(Full_Model)

##
## Call:
```

```
## glm(formula = responseVar ~ ., family = "binomial", data = Loans_Train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7119   0.2982   0.5152   0.7250   1.9628
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.571e+00  3.309e-01  10.791  < 2e-16 ***
## amount                    2.506e-05  7.587e-06   3.302 0.000958 ***
## term60 months            -8.241e-01  5.216e-02 -15.802  < 2e-16 ***
## rate                     -2.751e-01  1.372e+00  -0.201 0.841059
## payment                  -2.331e-01  4.119e-02  -5.660 1.52e-08 ***
## gradeB                   -4.264e-01  8.306e-02  -5.133 2.85e-07 ***
## gradeC                   -8.040e-01  1.095e-01  -7.343 2.09e-13 ***
## gradeD                   -1.037e+00  1.500e-01  -6.915 4.68e-12 ***
## gradeE                   -1.130e+00  1.850e-01  -6.108 1.01e-09 ***
## gradeF                   -1.288e+00  2.453e-01  -5.250 1.52e-07 ***
## gradeG                   -1.258e+00  3.106e-01  -4.050 5.12e-05 ***
## length1 year             -7.612e-03  7.801e-02  -0.098 0.922267
## length10+ years           4.581e-02  6.038e-02   0.759 0.447975
## length2 years             4.783e-02  7.312e-02   0.654 0.512991
## length3 years             1.668e-02  7.542e-02   0.221 0.825012
## length4 years            -2.278e-02  8.091e-02  -0.281 0.778333
## length5 years             3.369e-02  8.086e-02   0.417 0.676957
## length6 years            -3.442e-02  8.728e-02  -0.394 0.693312
## length7 years             8.312e-02  8.800e-02   0.945 0.344855
## length8 years            -4.857e-02  8.405e-02  -0.578 0.563365
## length9 years            -3.001e-02  9.380e-02  -0.320 0.749062
## homeOWN                  -8.366e-02  5.667e-02  -1.476 0.139846
## homeRENT                 -1.976e-01  4.437e-02  -4.452 8.49e-06 ***
## income                    3.224e-03  3.848e-03   0.838 0.402057
## verifiedSource Verified  -6.714e-02  4.090e-02  -1.642 0.100682
## verifiedVerified         -1.033e-01  4.446e-02  -2.324 0.020107 *
## reasoncredit_card        -2.591e-01  1.603e-01  -1.616 0.105987
## reasondebt_consolidation -2.161e-01  1.568e-01  -1.378 0.168175
## reasonhome_improvement   -2.781e-01  1.692e-01  -1.644 0.100269
## reasonmajor_purchase     -3.453e-01  1.939e-01  -1.781 0.074984 .
## reasonmedical            -5.424e-01  2.086e-01  -2.601 0.009300 **
## reasonmoving             -7.391e-01  2.293e-01  -3.223 0.001270 **
## reasonother              -2.685e-01  1.703e-01  -1.576 0.114937
## reasonsmall_business     -7.544e-01  2.105e-01  -3.584 0.000339 ***
## reasonvacation           -4.479e-01  2.427e-01  -1.845 0.065015 .
## debtIncRat               -2.718e-02  2.747e-03  -9.896  < 2e-16 ***
## delinq2yr                -7.255e-02  1.702e-02  -4.262 2.03e-05 ***
## inq6mth                  -6.568e-02  1.608e-02  -4.085 4.41e-05 ***
## openAcc                  -1.521e-02  6.989e-03  -2.176 0.029540 *
## pubRec                   -5.591e-03  2.531e-02  -0.221 0.825178
## revolRatio               -3.405e-01  1.274e-01  -2.672 0.007539 **
## totalAcc                  1.198e-02  1.912e-03   6.264 3.75e-10 ***
```

```
## totalBal                 -6.611e-03  8.734e-03  -0.757 0.449053
## totalRevLim               1.484e-02  5.040e-03   2.944 0.003240 **
## accOpen24                -7.273e-02  6.196e-03 -11.740  < 2e-16 ***
## avgBal                    8.606e-03  1.149e-02   0.749 0.453948
## bcOpen                    9.350e-03  5.376e-03   1.739 0.082012 .
## bcRatio                   2.810e-03  1.288e-03   2.181 0.029157 *
## totalLim                  1.141e-02  7.745e-03   1.473 0.140756
## totalRevBal              -5.480e-03  4.439e-03  -1.234 0.217051
## totalBcLim                1.581e-03  4.936e-03   0.320 0.748824
## totalIlLim                9.192e-03  2.637e-03   3.486 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29198  on 27723  degrees of freedom
## Residual deviance: 26056  on 27672  degrees of freedom
## AIC: 26160
##
## Number of Fisher Scoring iterations: 5
```

```
vif(Full_Model)
```

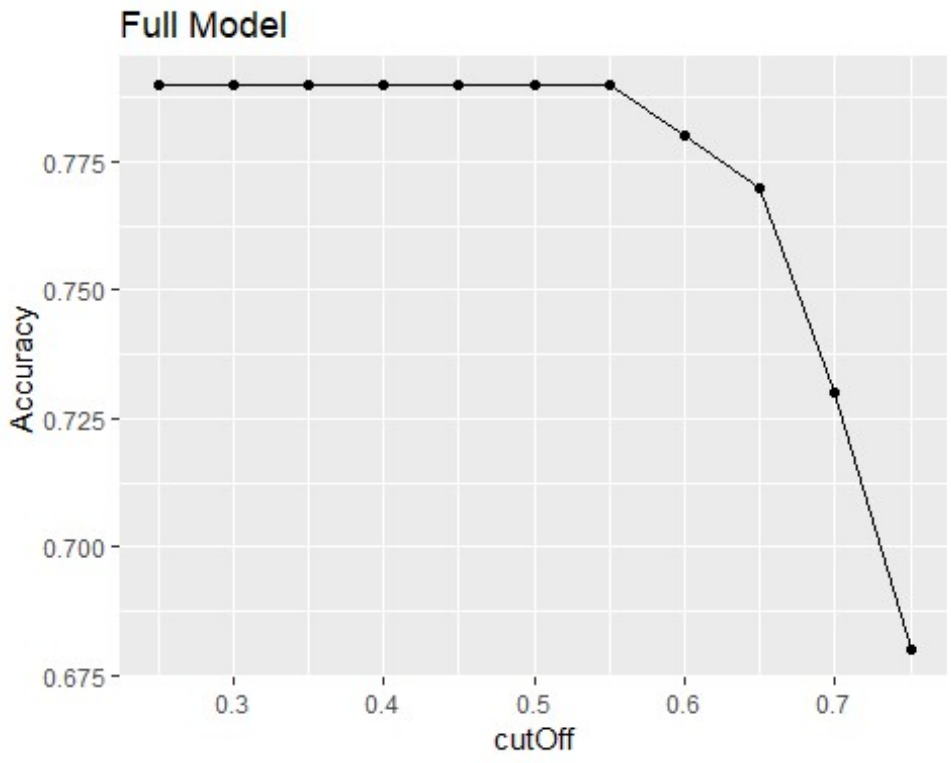
```
##                  GVIF Df GVIF^(1/(2*Df))
## amount      17.307837  1        4.160269
## term         2.607793  1        1.614866
## rate        14.422770  1        3.797732
## payment     13.996457  1        3.741184
## grade       15.073273  6        1.253672
## length       1.120770 10        1.005717
## home         1.896832  2        1.173565
## income       2.982570  1        1.727012
## verified     1.216919  2        1.050305
## reason       1.409624  9        1.019257
## debtIncRat   2.290561  1        1.513460
## delinq2yr    1.114972  1        1.055922
## inq6mth      1.170544  1        1.081917
## openAcc      6.315138  1        2.512994
## pubRec       1.091638  1        1.044815
## revolRatio   3.716129  1        1.927726
## totalAcc     2.303480  1        1.517722
## totalBal    89.941296  1        9.483739
## totalRevLim  5.847939  1        2.418251
## accOpen24    1.740793  1        1.319391
## avgBal      29.915540  1        5.469510
## bcOpen       7.639421  1        2.763950
## bcRatio      5.045316  1        2.246178
## totalLim    63.893820  1        7.993361
## totalRevBal  7.185114  1        2.680506
```

```
## totalBcLim   6.133638  1         2.476618
## totalIlLim   5.458038  1         2.336244
```

Threshold for Accuracy and Profit

The accuracy threshold plot is same for all the three models and we retained only the Full Model for analysis. The accuracy is highest at cutoff = 0.55.

```
plotfull
```

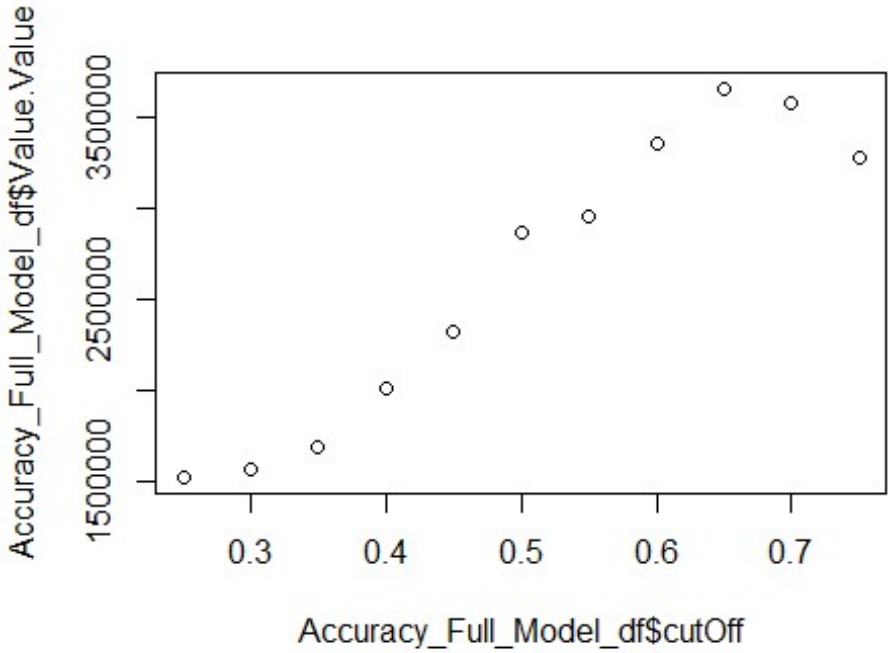



```
Accuracy_Full_Model_df
```

```
##    cutOff Accuracy Value.Value
## 1    0.25     0.79     1524924
## 2    0.30     0.79     1570214
## 3    0.35     0.79     1686748
## 4    0.40     0.79     2013898
## 5    0.45     0.79     2322976
## 6    0.50     0.79     2872413
## 7    0.55     0.79     2958175
## 8    0.60     0.78     3358008
## 9    0.65     0.77     3657314
## 10   0.70     0.73     3579288
## 11   0.75     0.68     3282813
```

Profit Threshold

```
plot(Accuracy_Full_Model_df$cutOff, Accuracy_Full_Model_df$Value.Value)
```



The maximum profit is derived when the cut off threshold equals 0.65. The profit for the perfect model or the test data as it is stands at 12596572 while the model at threshold of .65 gives a profit of 3657314. Thats a increase of 30% in profit.For the profit threshold of 0.65 the overall accuracy is 0.77

Results

The model predicts the Loans repayment outcome with a accuracy of 0.77 for maximum profit retention to the bank. The model though may not the perfect fit it does enhance the existing loan process.