Modeling and Prediction for loan default

Shekhar Karmarar

10/10/2019

Executive Summary

Reducing bad loans and maximizing profit is the goal of any financial institution. This study is conducted for Wealth Bank to help the process of loan sanction. This study will suggest ways to reduce defaults and increase profit. It will help to benefit both top line and bottom line of the bank.

I received previous customer loan records. I analyzed those records to find the factors which makes any particular loan good or bad. It gives a number from 0 to 100 to each case depending on all information of that customer, which tells how likely the loan will be default. With the help of this analysis, I devised some methods which can be implemented by loan officers at the time of sanctioning a loan. These methods will help the bank to decide whether to approve the loan or not.

From review and analysis of all records, it is clear that some factors make default more likely while other factors make repayment of loan more likely. I combined all factors to calculate likelihood of default for each loan application. The model will suggest clearly to approve or deny the loan depending on the likelihood derived for particular customer. This process will make judging the loan application fast, easy and less prone to errors.

By applying this model at the time of loan approval, the average profit made by bank for each loan application is projected to increase from $ 258 to $ 581, which is significant gain in profit. This will be achieved without sacrificing large number of loan applications. The implementation will make better use of investment made by the bank to reduce defaults and increase profit.

Current Profit	Profit after Model
$ 258	$ 581

I will recommend to implement this study and use it at the time of approval of loan. This will increase profit and reduce the default rate. I will also recommend giving continuous feedback so that depending on actual result of loan outcome, this model can be improved.

There are certain limitations to application of any model, and this study is also not an exception. As market conditions change, we have to make improvements to this model. It will be possible to keep track of changes and update the model if we get regular feedback from all related persons.

It should be noted that the criteria used for approval is not hard and fast, but a compromise number where it is supposed to optimize the lending process. With further study and feedback, we should be open to change the criteria and update the model.

I thank Wealth bank for the trust they have shown and allowed us to serve.

1.Introduction:

The dataset 'Loans50k' contains information about loans made by financial institute to applicants for various purposes with duration up to five years. The task is to perform logistic regression analysis to predict which applicants are likely to default on their loans. The dataset contains total 50,000 observations with 32 variables. With multiple variables affecting the probability of loan default, it was essential to narrow down to few significant variables and create a model for prediction.

2. Preparing and cleaning the data

Load essential library and then load data

```
require(dplyr)
require(ggplot2)
require(gridExtra)
require(leaps)
require(ISLR)
require(pROC)
require(ROCit)
require(ROCR)

loans <- read.csv('loans50k.csv')
```

There is 'status' variable which shows the result about what is happening or happened to the loan.

```
## # A tibble: 8 x 2
##   status             count
##   <fct>              <int>
## 1 ""                     1
## 2 Charged Off         7579
## 3 Current            14532
## 4 Default                2
## 5 Fully Paid         27074
## 6 In Grace Period      261
## 7 Late (16-30 days)    102
## 8 Late (31-120 days)   449
```

As we are not interested in status other than 'Charged Off/ Default' and 'Fully Paid', I created new dataset with only default and fully paid status. There are only two entries for default. I merged those with Charged Off and created dataset with only two status.
Now we have our new dataset with only two status ready for further processing. merged those with Charged Off and created dataset with only two status.

```
## # A tibble: 2 x 2
##   status      count
##   <chr>       <int>
## 1 Charged Off  7581
## 2 Fully Paid  27074
```

The ratio of bad to good loans is around 1:3.5. This ratio is important for graph analysis of various variables down the road.

I created one new variable 'response' with two factors: Yes and No. This variable will serve as response variable of logistic regression. Yes means there is default of loan.

```
##    No   Yes
## 27074  7581
```

I created new variable 'profit' which is 'totalPaid' - 'amount'. This variable will help to predict profit prediction from model.
This is summary of profit showing profit or loss as a whole on all accounts.

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -35000.0    239.6   1172.0    191.1   2467.3  23697.4
```

There are some variables which are either redundant or not useful for analysis.
I will remove these variables from dataset.

1.**rate** (It is part of monthly payment amount and not as separate predictor)
2.**grade**(I am not sure how it is decided, and my analysis is going to determine the same thing)
3.**employment**(job title are so many, and do not provide any help for analysis)
4.**length**(time continuously employed in past is not related to current capacity to repay.)
5.**reason**(they are not useful and more of description nature.)
6.**state**(it is only demographic variable. Some states may have more default but place of residence should not be part of bias.)
7.**totalPaid** (This is invalid as we do not know at loan approval)
8.**totalIlLim** and totalBcLim(these are redundant with totalLim)

There are some discrete variables which can be **aggregated in fewer categories.**

1. delinq2yr : reduced to four categories

2. inq6mth : reduced to four categories

3. pubRec : reduced to three categories

```
## # A tibble: 4 x 2
##   delinq2yr count
##       <dbl> <int>
## 1         0 27600
## 2         1  4678
## 3         2  1396
## 4         3   981
```

```
## # A tibble: 4 x 2
##   inq6mth count
##     <dbl> <int>
## 1       0 19237
## 2       1  9712
## 3       2  3637
## 4       3  2069
```

```
## # A tibble: 3 x 2
##   pubRec count
##    <dbl> <int>
## 1      0 28076
## 2      1  5422
## 3      2  1157
```

Checking all columns for **NA values**. The count of NA are small and I will impute them with median values.

```
## [1] "revolRatio" "bcOpen"     "bcRatio"
```

```
## [1]  15 360 384
```

```
loans_use <- loans_use %>%
   mutate(revolRatioNA  = ifelse(is.na(revolRatio),median(loans_use$revolRatio, na.rm = T) , revolRatio
   bcOpenNA = ifelse(is.na(bcOpen),median(loans_use$bcOpen, na.rm = T) , bcOpen),
```

```
  bcRatioNA = ifelse(is.na(bcRatio),median(loans_use$bcRatio, na.rm = T) , bcRatio)
  )
```

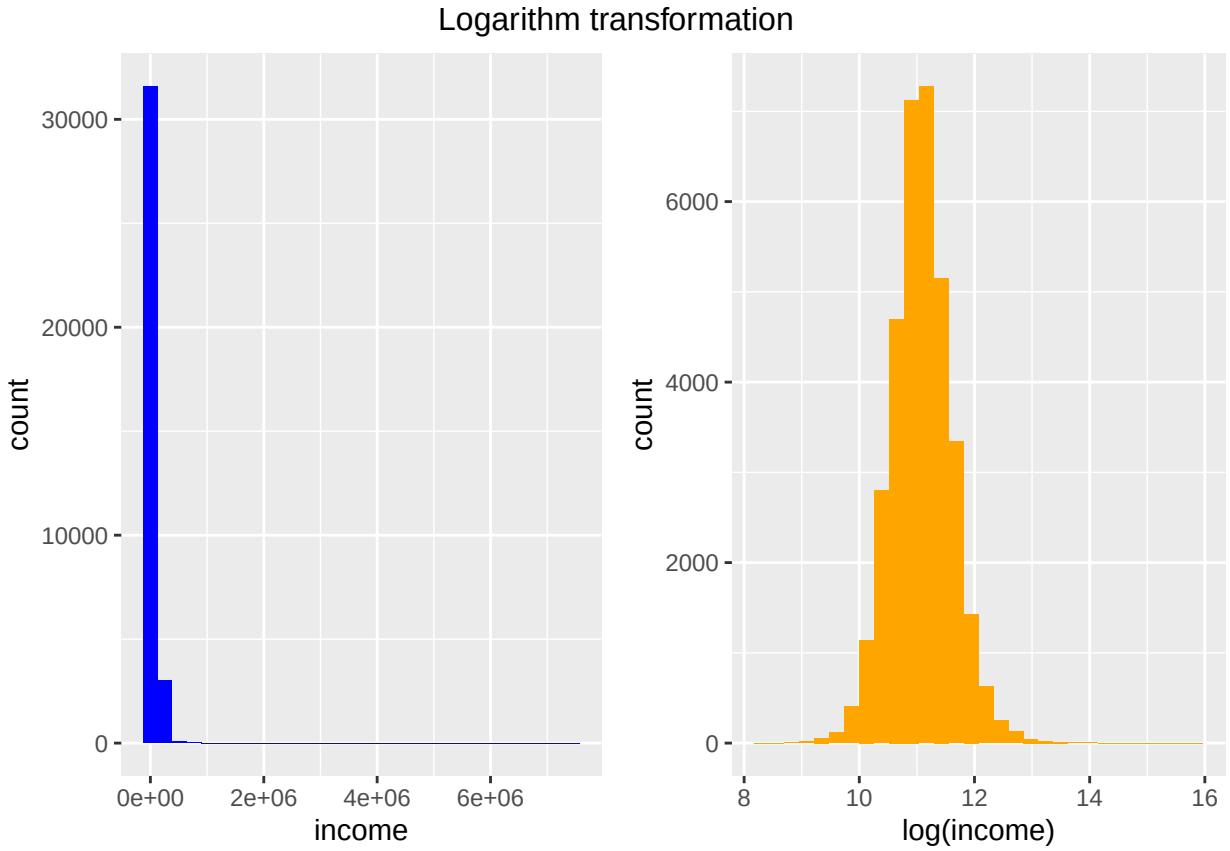
3.Exploring and transforming the data

I will use following tools for exploring data.
1.**Summary statistics**
2.**Histogram of quantitative variables**
3.**bar chart of discrete variables**
4.**box plot**

This will give me idea about which distribution of variables and their significance.

Now let us explore some variables to see how they affect good and bad loans.
First I will see distribution of some variables which are highly skewed.

There are six variables which are highly right skewed.
Using log transformation will make it more towards normal distributions.



I will use log transformation for these variables :
1. income
2. totalBal
3. totalRevBal
4. totalLim
5. avgBal
6. bcOpen

The variables I am interested in are showing different distribution for good and bad loans, and I want to use

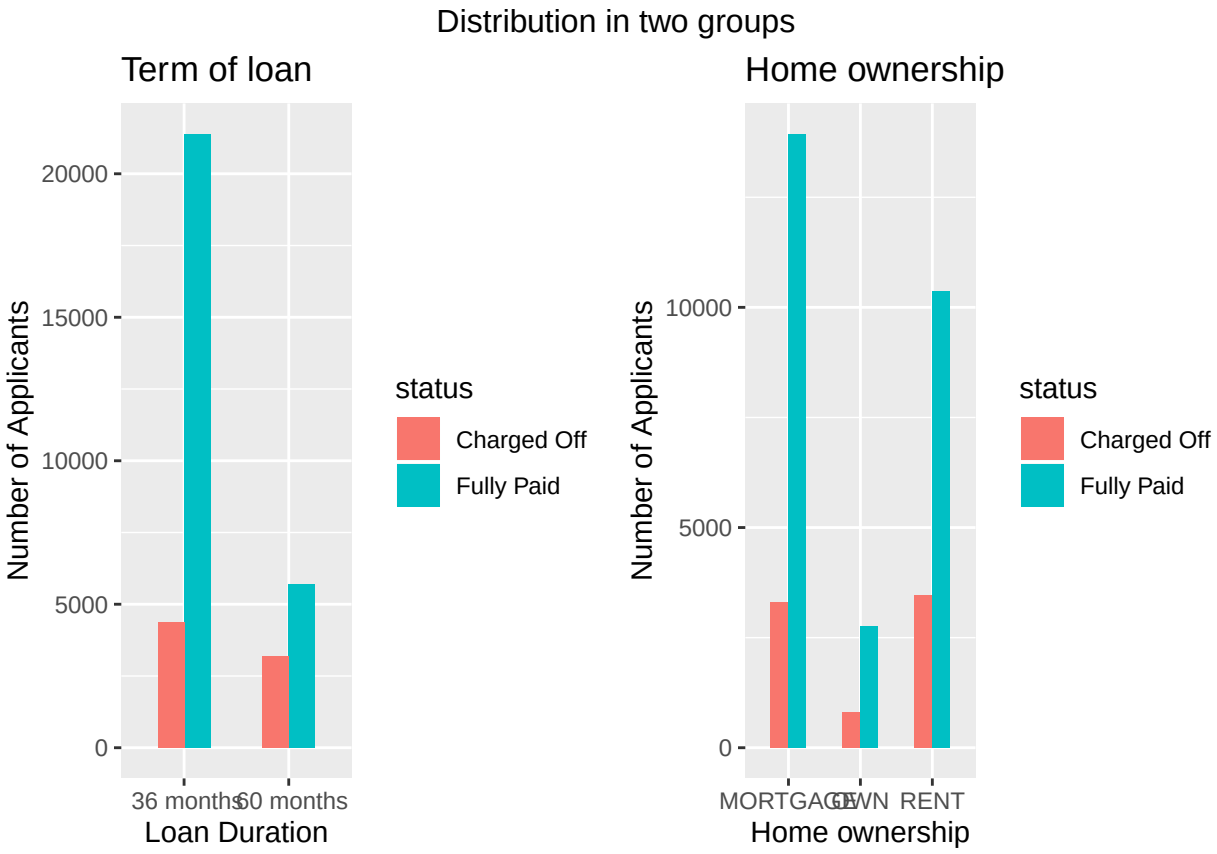
them in analysis.

1.Term

60 months term has more default cases.

2.home

Applicants with rented home are more likely to default.



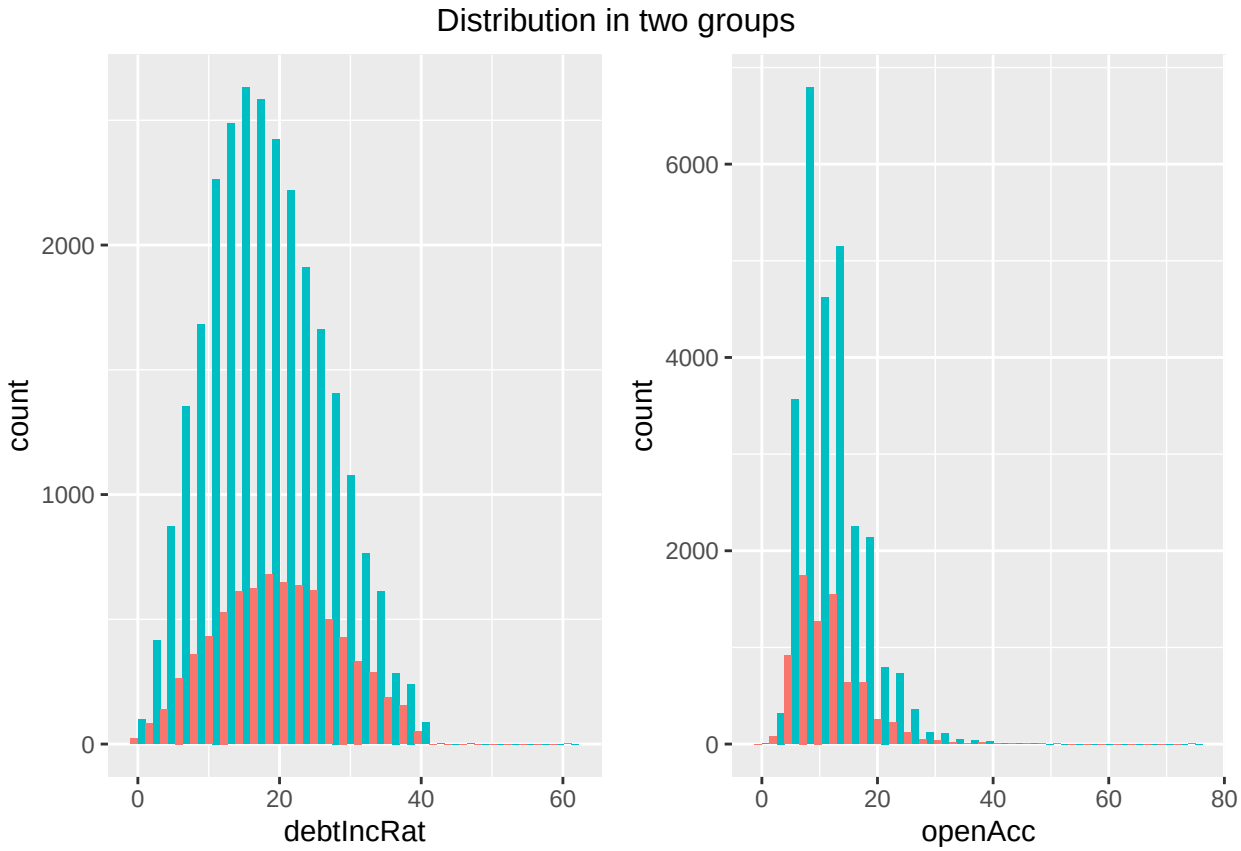
3.Debt to Income ratio

When the ratio is low, loan is good.

There are some variables related to credit card use and limit, which I think does not show different distribution for good and bad loans. I am not using these variables for analysis. I will show the distribution of these variables on good and bad loans.

4.Open credit cards

The graph is almost similar and mean and sd are not much different.
I am not using this variable for analysis.



Analysis of variables by use of summary statistics

This is example of variable , 'openAcc', which does not show difference of summary statistics in two groups of good and bad loans. I am not using this variable for analysis.

```
## # A tibble: 2 x 3
##   status       mean    sd
##   <chr>       <dbl> <dbl>
## 1 Charged Off  12.1  5.80
## 2 Fully Paid   11.7  5.36
```

The data is now ready for logistic regression analysis.

4. The logistic model

Preparation of data for modeling

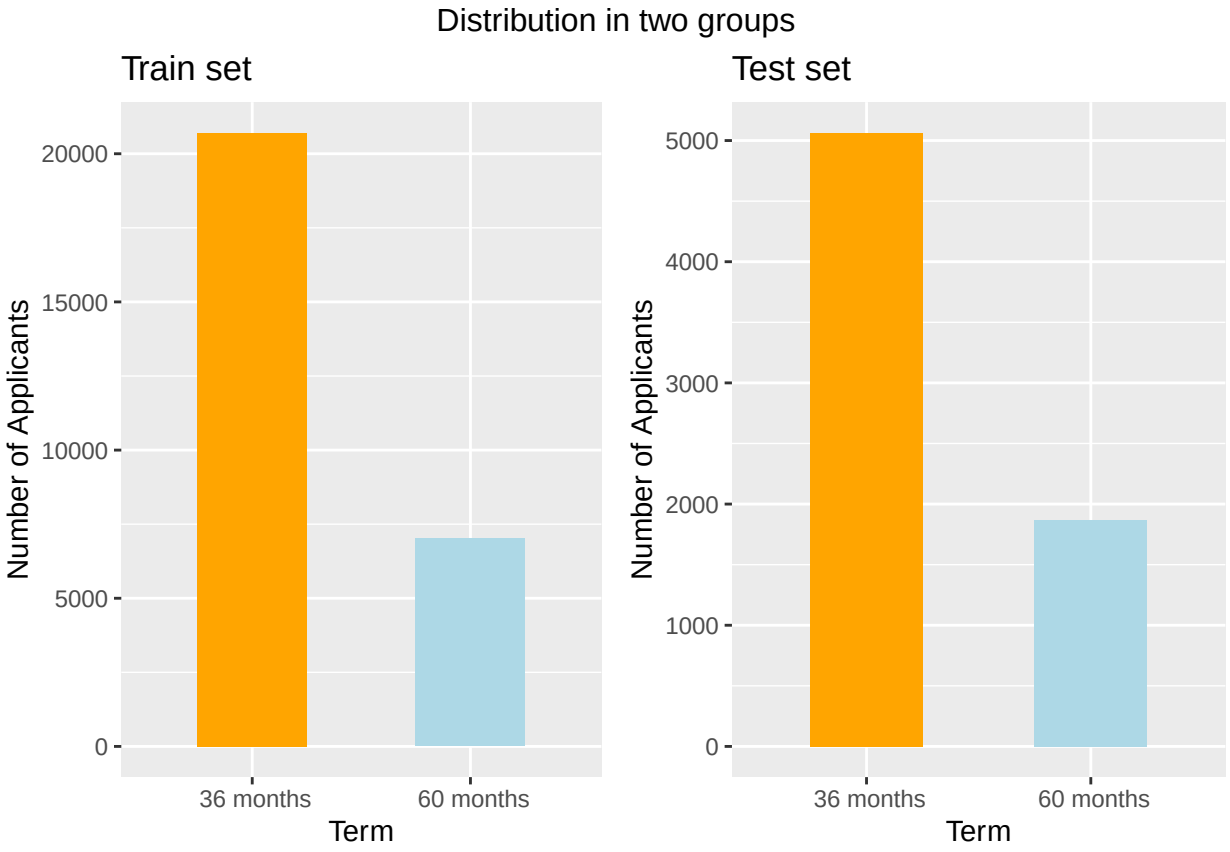
I have added new fields for variables log to data to be used in model.

I created new data with only those variabls which will go into model for analysis.

I divided dataset into two parts chosen randomly. One part will be 80% to create a model and 20% part which will be used to test the results.

```
size  <-  floor(0.8 * nrow(dfmodel))
set.seed(1000)
train_ind  <-  sample(seq_len(nrow(dfmodel)),size = size)
train_set  <- dfmodel[train_ind,]
test_set <- dfmodel[-train_ind,]
```

I will see whether data is split randomly. I will graph distribution of term variable in both sets. This confirms that our data is split randomly into two groups by **80:20 split**.



As this is logistic regression, I will create a model with glm function.

First I used forward step to get best model.

```
loans.full <- glm(response ~ . , data = train_set, family = 'binomial')
loans.null <- glm(response ~ 1 , data = train_set, family = 'binomial')
forward.step <- step(loans.null, scope = list(lower = loans.null, upper = loans.full), direction = 'for
summary(forward.step)
```

Then I used backward step to get best model.

```
backward.step <- step(loans.full, direction = 'backward')
summary(backward.step)
```

I have compared AIC values for both and they are almost same. Both are 26335.

I will choose this model for final analysis.

glm(formula = response ~ income_log + home + verified + debtIncRat + delinq2yr + amount + term + payment + inq6mth + openAcc + revolRatioNA + totalAcc + accOpen24 + bcOpen_log + totalLim_log + totalBal_log, family = "binomial", data = train_set)

```
model_final <- glm(formula = response ~ income_log + home + verified + debtIncRat +
    delinq2yr + amount + term + payment + inq6mth + openAcc +
     revolRatioNA + totalAcc + accOpen24 + bcOpen_log +
     totalLim_log +  totalBal_log,
    family = "binomial", data = train_set)
```

```r
model_summary <- summary(model_final)
model_summary
```

```
##
## Call:
## glm(formula = response ~ income_log + home + verified + debtIncRat +
##     delinq2yr + amount + term + payment + inq6mth + openAcc +
##     revolRatioNA + totalAcc + accOpen24 + bcOpen_log + totalLim_log +
##     totalBal_log, family = "binomial", data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9190  -0.7086  -0.5352  -0.3422   2.7549
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              2.8107152  0.4625549   6.077 1.23e-09 ***
## income_log              -0.2153825  0.0466822  -4.614 3.95e-06 ***
## homeOWN                  0.0431915  0.0561824   0.769 0.442028
## homeRENT                 0.1540788  0.0416016   3.704 0.000212 ***
## verifiedSource Verified  0.0868596  0.0404156   2.149 0.031622 *
## verifiedVerified         0.1665470  0.0437540   3.806 0.000141 ***
## debtIncRat               0.0270390  0.0022156  12.204  < 2e-16 ***
## delinq2yr                0.1761749  0.0227061   7.759 8.56e-15 ***
## amount                  -0.0001186  0.0000117 -10.139  < 2e-16 ***
## term 60 months           1.6518423  0.0662798  24.922  < 2e-16 ***
## payment                  0.0043663  0.0003572  12.222  < 2e-16 ***
## inq6mth                  0.1233167  0.0175335   7.033 2.02e-12 ***
## openAcc                  0.0168449  0.0042435   3.970 7.20e-05 ***
## revolRatioNA             0.2308514  0.0900156   2.565 0.010330 *
## totalAcc                -0.0165930  0.0018767  -8.841  < 2e-16 ***
## accOpen24                0.0956077  0.0059608  16.039  < 2e-16 ***
## bcOpen_log              -0.0579405  0.0095437  -6.071 1.27e-09 ***
## totalLim_log            -0.5076170  0.0753621  -6.736 1.63e-11 ***
## totalBal_log             0.2443370  0.0599808   4.074 4.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29195  on 27723  degrees of freedom
## Residual deviance: 26347  on 27705  degrees of freedom
## AIC: 26385
##
## Number of Fisher Scoring iterations: 4
```

The AIC value of final model is 26385.

As this model is finalized, I will generate responses depending on prediction of test data, to see how effective is this model.

```r
probs <- predict(model_final, test_set, type='response')
loans2 <- cbind(test_set, probs)
loans2$probs2 <- ifelse(loans2$probs > 0.5, '1','0')
```

```
loans2_table <- table(loans2$response, loans2$probs2)

addmargins((loans2_table))
```

```
##
##          0    1  Sum
##   No  5260  181 5441
##   Yes 1293  197 1490
##   Sum 6553  378 6931
```

```
nrow(test_set[test_set$response == 'Yes',]) # Default cases
```

```
## [1] 1490
```

```
nrow(test_set[test_set$response == 'No',]) # Good loans
```

```
## [1] 5441
```

Now I will analyze **confusion matrix**. In test data, actual result is 1490 were defaulted and 5441 paid loan fully. In the prediction of this model, 378 defaulted and 6553 paid fully.
True positive prediction is 197 out of 1490 actual defaults. The percentage is 0.13 or **13%.** At this threshold of 0.5, the model is very very poor in predicting the true default rate. I have to adjust the threshold level to make it more accurate to predict true default rate.

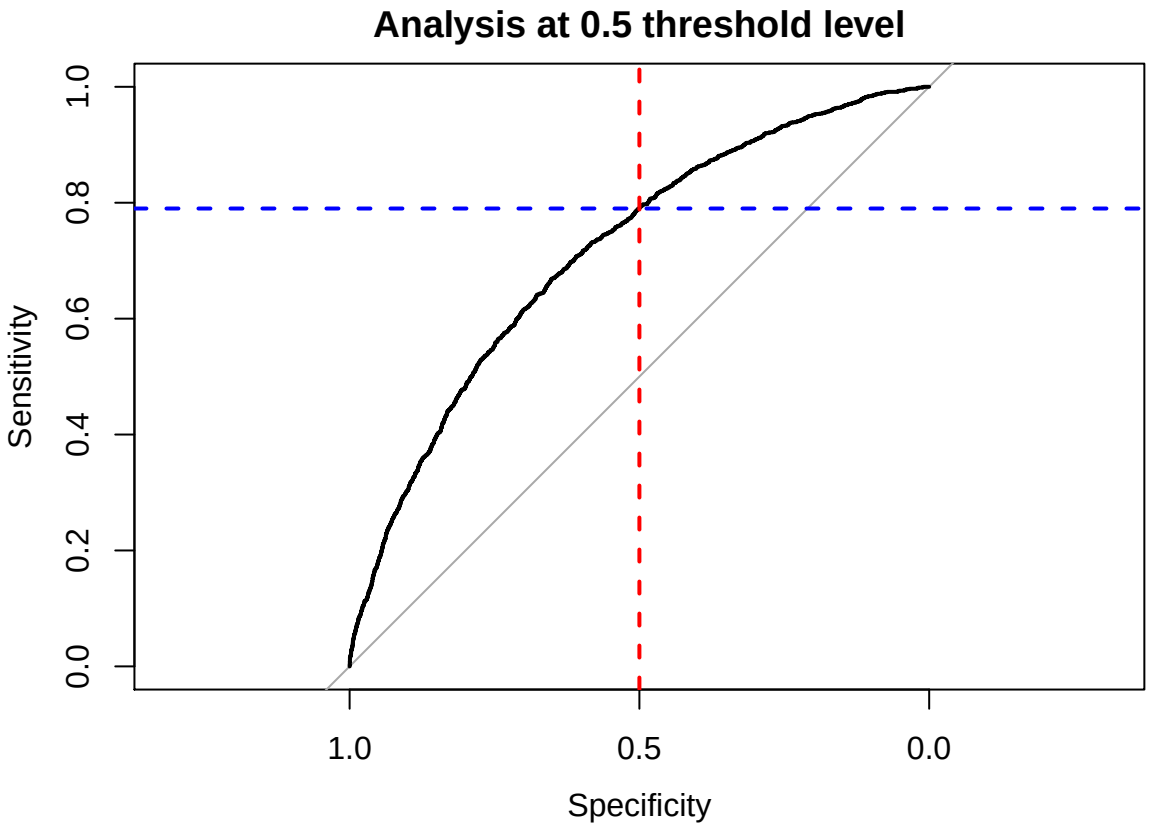
If I analyze total correct outcome, that is combination of true positive of default and fully paid, I get (5260+197)/ 6931 = 0.78 or **78 %.** This is overall better accuracy of prediction.

The good loan prediction with the threshold of 0.5 is 5260/5441 is 0.96 or **96%** which is very good. I can see that this threshold predicts good loans much better but I sacrificed the prediction of bad loans.

5. Optimizing the threshold for accuracy

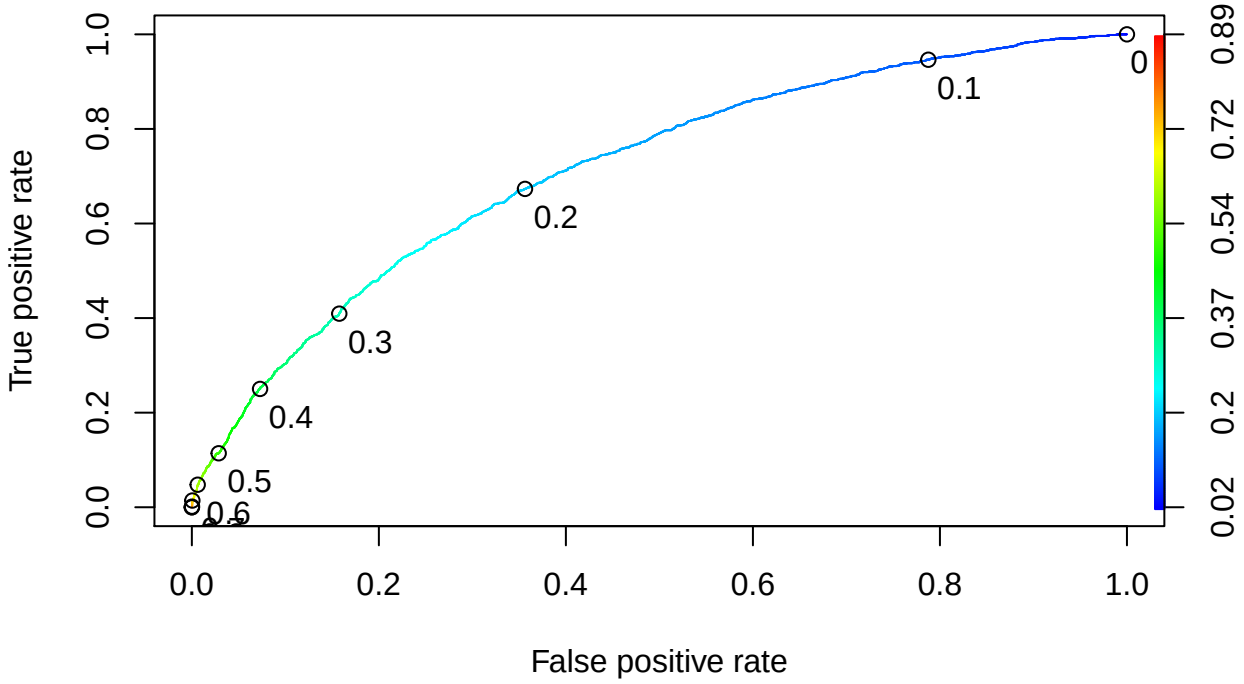
It is necessary to study how change in threshold from 0.5 changes accuracy.

```
model_test <- glm(formula = response ~ income_log + home + verified + debtIncRat +
    delinq2yr + amount + term + payment + inq6mth + openAcc +
     revolRatioNA + totalAcc + accOpen24 + bcOpen_log +
     totalLim_log +  totalBal_log,
    family = "binomial", data = test_set)

pr=predict(model_test,type=c("response"))

test_set$pr <- pr

g <- roc(response ~ pr, data = test_set)
plot(g, main = 'Analysis at 0.5 threshold level')
abline(v = 0.5, col="red", lwd=2, lty=2)
abline(h = 0.79, col="blue", lwd=2, lty=2)
```

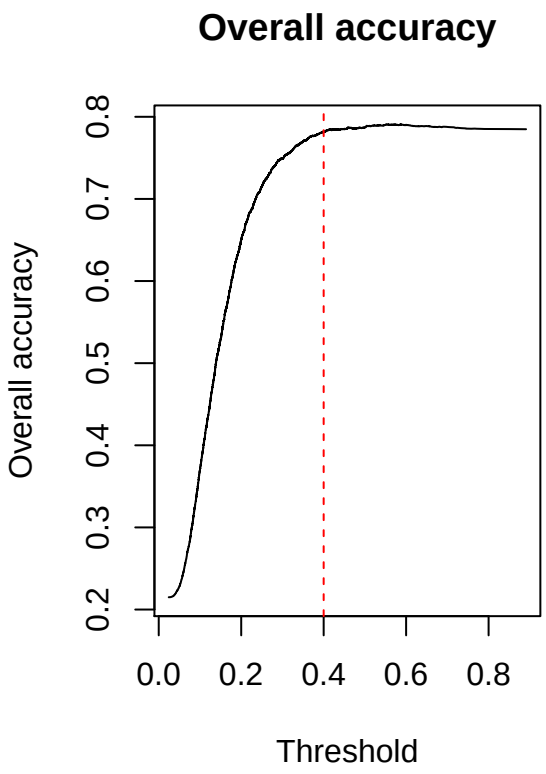


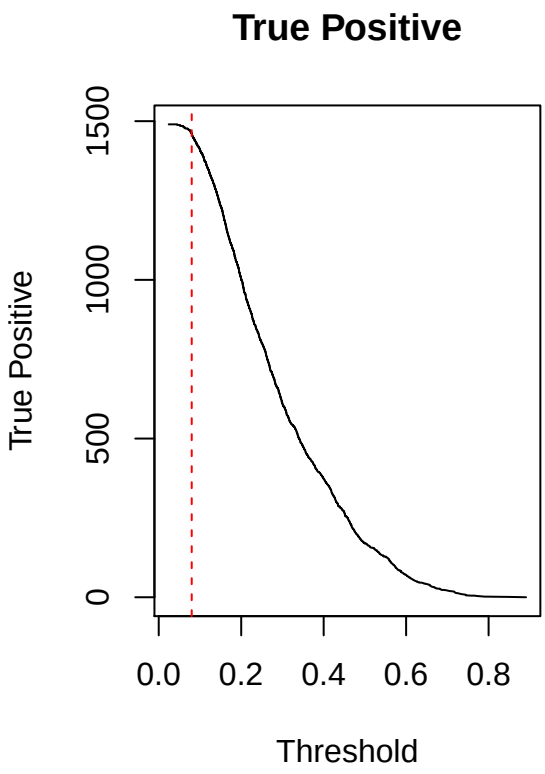
Analysis of the graph : The graph is sensitivity vs specificity of the model. The black curved line is how model will behave. At the threshold level of 0.5, the specificity is 0.5 and sensitivity is 0.78. Blue and black dashed lines show these marks. Upto this point, sensitivity increases with reduction in specificity. Beyond this point, there is no much gain in sensitivity but we lose specificity very fast. There is always trade off between two.

```
ROCRpred = prediction(pr, test_set$response)
ROCRperf = performance(ROCRpred, "tpr", "fpr")
#Plot ROC curve
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```



Analysis of the graph : This is graph of true positive (default cases model will identify correctly) and false positive (model will label as default but actually they are not). In ideal world, the graph should go straigth high and stay at 100% level. It means the model should pick 100 % true positive and 0% false positive. In real world, the models are not ideal and there is tradeoff between two.





Analysing threshold level

When I want to decide the threshold, it depends on the goal. For overall accuracy of the model, threshold is around 0.4. At this level, true positive and true negative combinely have maximum value. As a model, this is the level I would prefer. If bank wants to maximize profit, then the threshold level is **0.08** as noted in coming section. According to true positive graph, **true positive** rate starts falling sharply after threshold value of 1. This is tradeoff for any threshold value. If bank adopts level of 0.08 for maximizing profit, the chances that the bank will deny loan to good customer increases. If bank wants to be inclusive in customer service, then rate of default increases affecting profit.
I will explain this to bank and it is up to the bank to decide at which threshold level it wants to implement the decision.

6. Optimizing threshold for profit

Profit made by bank is most important factor to decide approval of loan. This analysis will calculate average increase in profit per client before and after applying the model. I will change the classification threshold from 0.5 to the level of optimization of profit.
First get test data for profit calculation.

```
size  <-  floor(0.8 * nrow(loans_use))
set.seed(1000)
train_ind  <-  sample(seq_len(nrow(loans_use)),size = size)
train_profit  <- loans_use[train_ind,]
test_profit <- loans_use[-train_ind,]
summary(test_profit$profit)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

```
## -35000.0    262.3   1157.6    258.1   2507.5  20532.0
```

The average profit made by bank actually including all clients is $ 258.
Now I will calculate average profit excluding bad loans.

```
good_loans <- test_profit$profit[test_profit$response == 'No' ]
bad_loans <- test_profit$profit[test_profit$response == 'Yes' ]
summary(good_loans)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     1.16   833.08  1596.06  2343.71  3019.01 20532.05
```

If bank denies actual bad loans, the profit per client would be **$ 2343 which is maximum potential profit** we can expect by application of model.

I will apply model to test data to categorize loans into good and bad by model prediction. I will remove all bad loans in prediction and calculate the profit on good loans predicted by the model.

```
probs_profit <- predict(model_final, test_set, type='response')
loans2_profit <- cbind(test_profit, probs_profit)
loans2_profit$probs2 <- ifelse(loans2_profit$probs_profit > 0.5, '1','0')
#loans2_profit_table <- table(loans2_profit$response, loans2_profit$probs2)
#addmargins((loans2_profit_table))
good_predict <- loans2_profit$profit[loans2_profit$probs2 == '0']
summary(good_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -35000.0    330.0   1172.0    428.8   2478.6  20532.0
```

With use of model, I predicted bad loans with **threshold of 0.5.** After excluding bad loans by prediction, the average profit made by bank is **$428** which is better than **$258** when model is not used at all.
I will try to optimize the threshold so that the profit is optimized.

```
loans2_profit$probs2 <- ifelse(loans2_profit$probs_profit > 0.08, '1','0')
good_predict <- loans2_profit$profit[loans2_profit$probs2 == '0']
summary(good_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -27782.0    482.1    918.3    982.6   1570.8  11246.0
```

As it can be seen from analysis, if I make threshold 0.08, the profit will be maximized at **$ 982**. This is most efficient way to optimization of profit.

I will see confusion matrix at maximum profit threshold.

```
##
##          0    1  Sum
##   No   664 4777 5441
##   Yes   33 1457 1490
##   Sum  697 6234 6931
```

This is profit level at each threshold used to predict profit in test set.

Threshold Avarage Profit
0.9 — 257
0.8 — 260
0.7 — 292
0.6 — 349
0.5 — 428
0.4 — 581
0.3 — 744

0.2 — 903
0.1 — 938
0.08 — 982 **
0.05 — 462

At the threshold level of **0.08** where profit is maximized, the model predicts 1457 out of 1490 bad loans correctly, at 97% accuracy. This comes at a price of predicting good loans only 664 out of 5441 correctly at only 13% accuracy.

7. Results summary

I have used **backward step generalized linear model** for binary logistic regression to model and predict loan default and optimization of profit. The model classifies new applicant into good loan and bad loan depending on prediction of the model from various criteria. These criteria are chosen according to their significance of effect on response variable.

For **overall accuracy**, the value I suggest is **0.4** where the combined value of true positive and true negative is maximum of 77% . At this level, True positive rate is 26% which means bad loans will be labeled as bad loans. True negative rate is 91 % which means good loans will be labeled as good loans. At 0.4 level, the profit will be $ 581 compared to $ 258 currently without implementation of any model.

For **maximization of profit**, the threshold level will be **0.08** where true positive rate is 97% compromising true negative rate. In my opinion, 0.4 is chosen threshold for maximum overall accuracy. At this level, the bank will not discriminate against good loans for the sake of profit and will be more inclusive towards customers.