Predicting Loan Status - Data Transformation and Modeling with Logistic Regression

Rita Miller

04/03/2024

Load Packages

```
library(tidyverse)
library(e1071)
library(usmap)
```

Load Data

```
dataset <- read.csv("loans50k.csv")
```

Executive Summary

We created a statistical model to assist financial institutions in the decision-making process to underwrite customers in the loan approval process. This included a classification threshold (probablity of a customer to pay their loan) to be used to predict loan status and make the most profits. We built a logistic regression model to predict good loans (fully paid off) and bad loans (charged off/defaulted). The figure below shows that as threshold increased, the count of disbursed loans decreased. Also, as threshold increased, profits rose and then decreased. In essence, the stricter a bank is on bad loans, the less profits they would make. Nevertheless, there is a tradeoff between accuracies to predict good versus bad loans. If we increase the threshold value, we are taking away the chance of disrbursing more loans from the bank, hence limiting the amount of profits with low risk. However, at a low threshold value, the bank may disburse more loans, but the risk for defaults would be high, but the bank could make more profits. The results also show that compared to not using our model, a bank could expect to make $1,299,446 in profits. Alternately, the maximum profits from using our model is projected at $4,120,756. Recommendations Perform separate models to explore good and bad loans. We could review specific variables like employment in an economic downturn. Questions could be more specific to the risk for defaulting on a loan, or adding questions that are specific to the industry of the borrower.We hope you will strongly consider using our model to help to increase your profit margins.

Introduction

The relationships among several fields of data were explored with a goal to predict customers who were likely to default on their loans. We set out to find key predictors which may have a major impact on those loans. We started with a data frame with 50,000 observations, 32 variables with some numeric, others categorical and some with missing values. We then established the inclusion criteria and created a new status to include loans that were fully paid (good loans), charged off and defaulted (bad loans) in our data. Once the inclusion criteria were established, the data was filtered down to 34,655 observations and 31 variables. Some variables like LoanID and employment were excluded, since they appeared to be inapplicable to the goal. The data was further segmented into good and bad loans revealing a total of 27,074 in good status and 7,581 loans in bad status. R programming was used to prepare, clean, explore, and transform the data. Finally, we will create a model and describe its most important features and how it changes the overall profit for the bank.

Preparing, Cleaning, Inclusion/Exclusion Criteria

We kept loans that were fully paid, charged off and defaulted in our data.

We excluded the -loanID variable because it was only an identifier and appeared inapplicable to the goal to predict customers who were more likely to default on their loans.

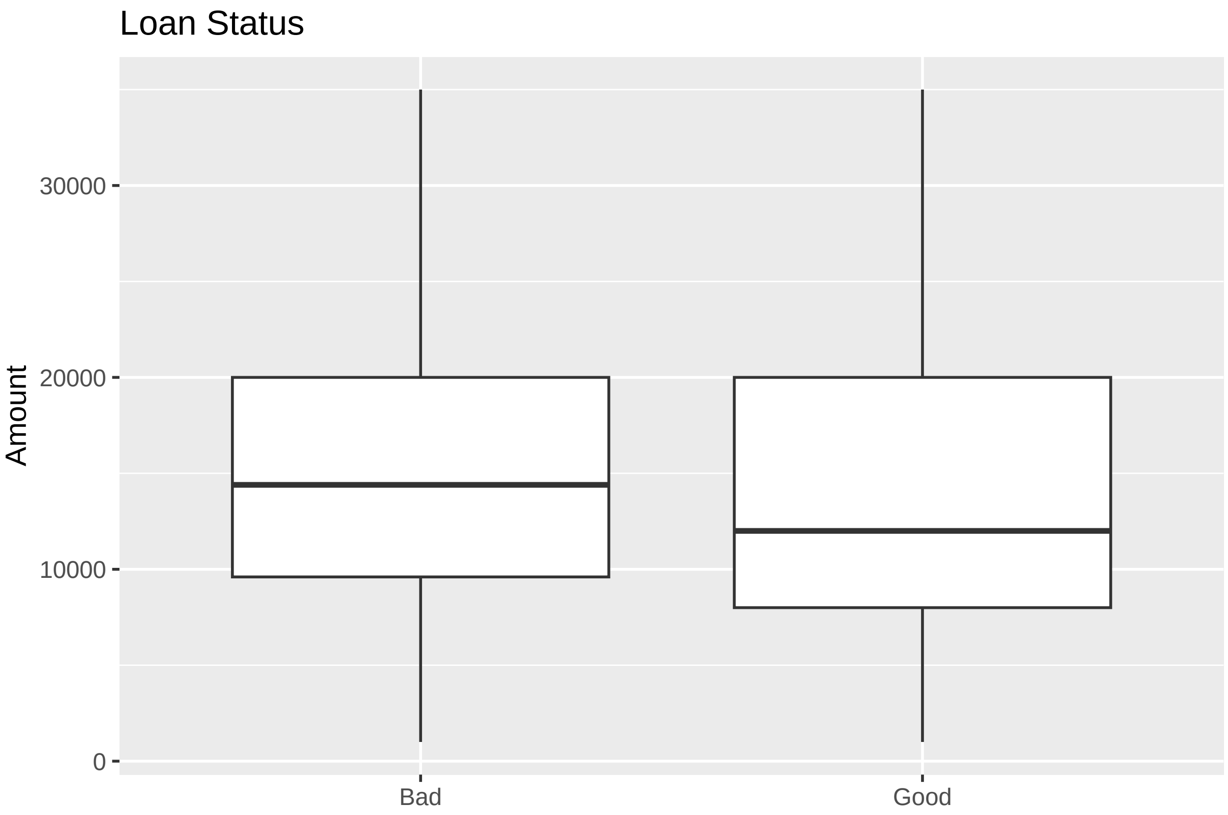
Feature engineering and NA values

We used feature engineering to consolidate the categories of the variable called "reason" into a single category called "other," because some categories were too sparse. Secondly, we integrated the "states" to the 5 regions of the U.S., because not all states were labeled.

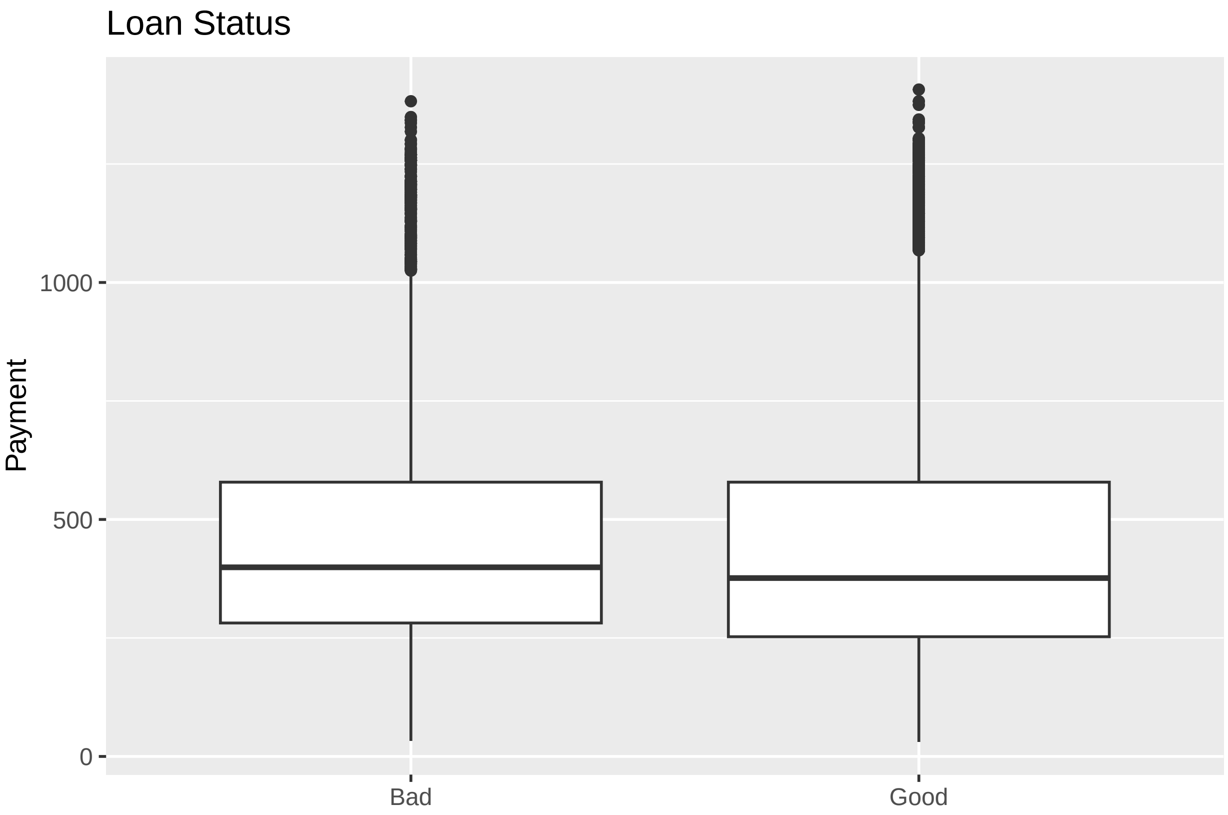
NA's occurred in variables "bcOpen," "bcRatio" and "revolRatio," so we replaced those values by the mean. When we disregard cases with any missing variables, we may lose useful information that the non-missing values may convey. Therefore, we may impute reasonable values (those that will not skew the results of analyses very much) for the missing values and that is the reason for replacing Na's with the mean.

Exploring and Transforming the Data

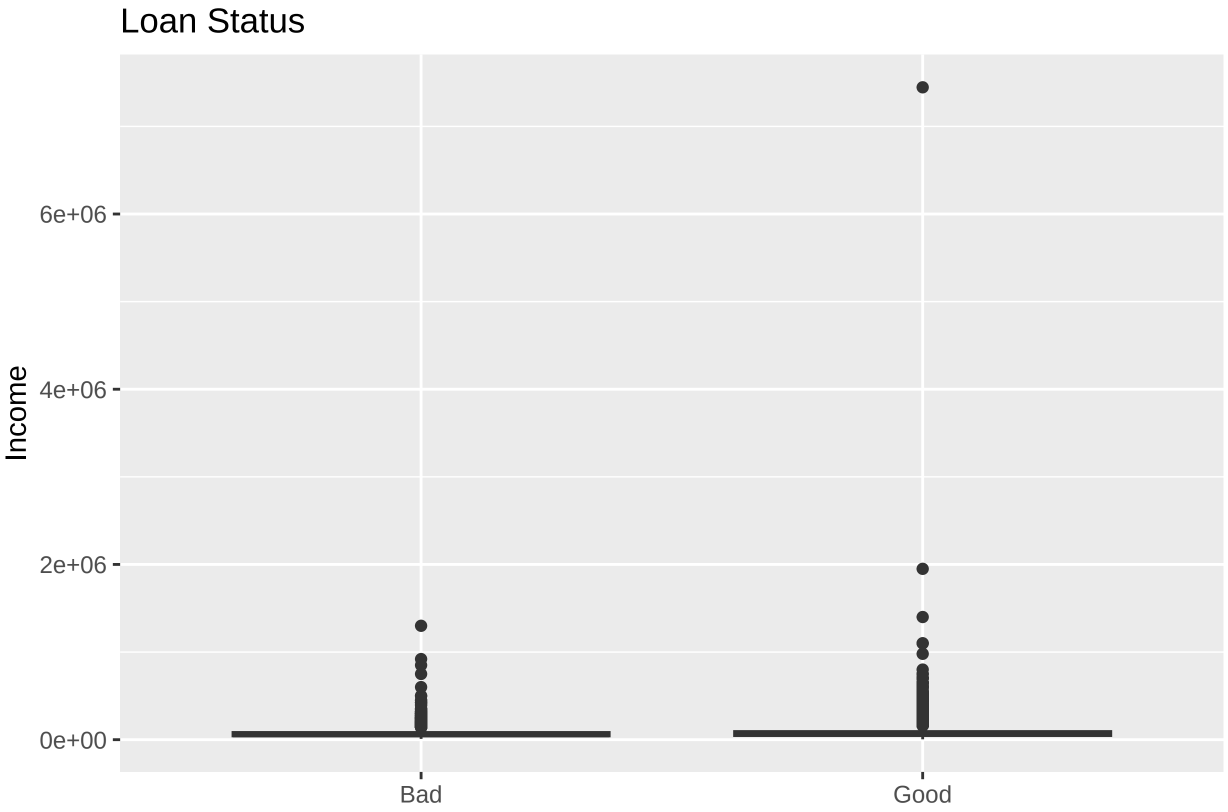
Next, we checked the assumptions of the data. The following charts revealed the distributions of some of the quantitative predictor variables to see if they were distributed differently for good and bad loans.



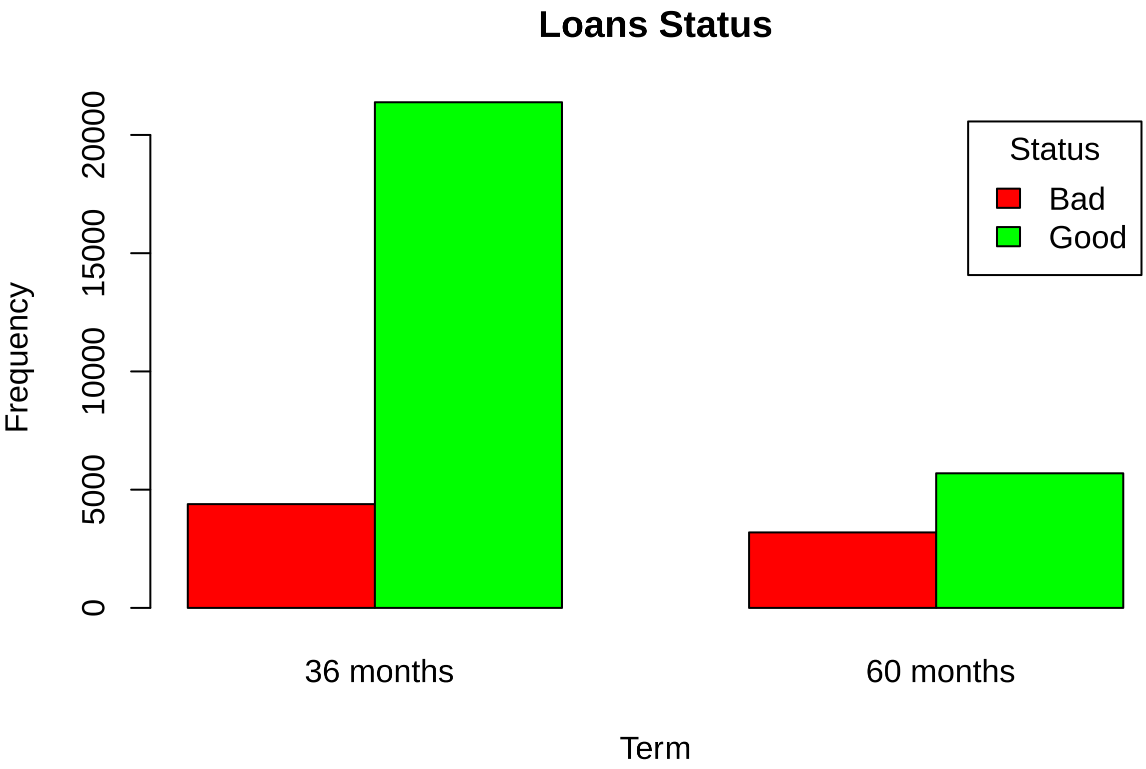
The distribution of amount for good and bad loans were moderately skewed to the right with no apparent outliers. Will replace the quantitative predictor variables with transformed values using logarithms (log(x+1) to prevent log(0).



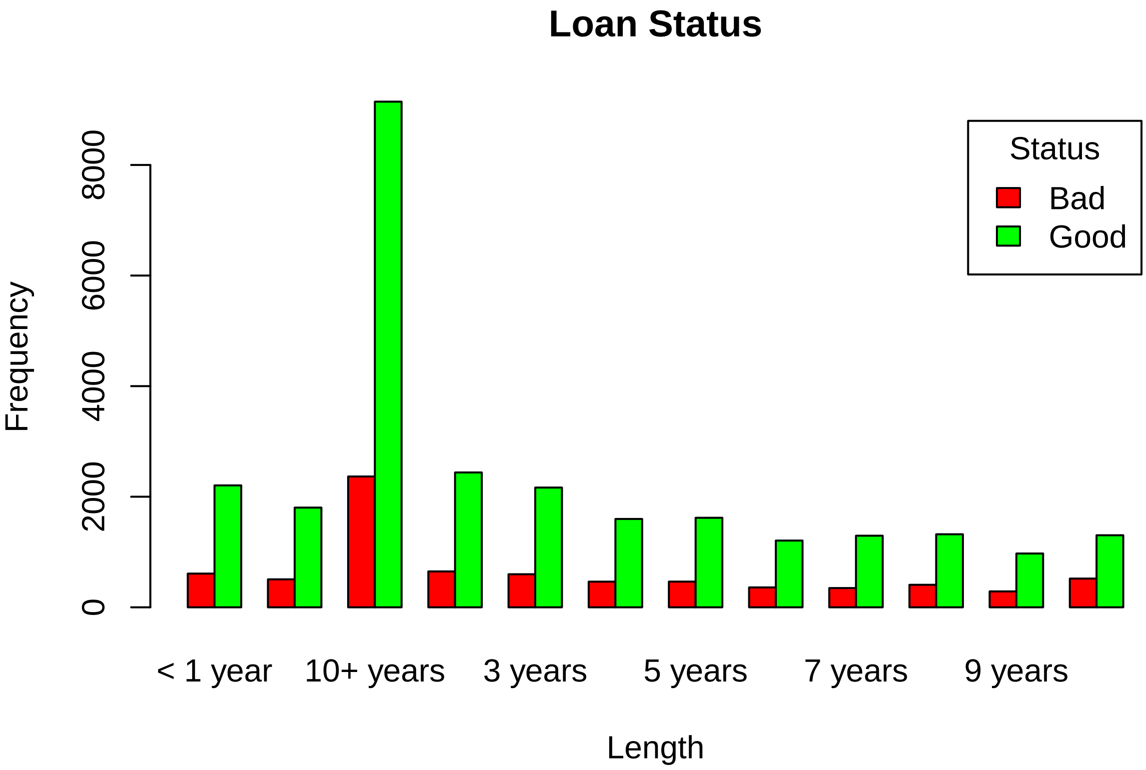
The distribution of payment for good and bad loans were moderately skewed to the right with some outliers apparent.



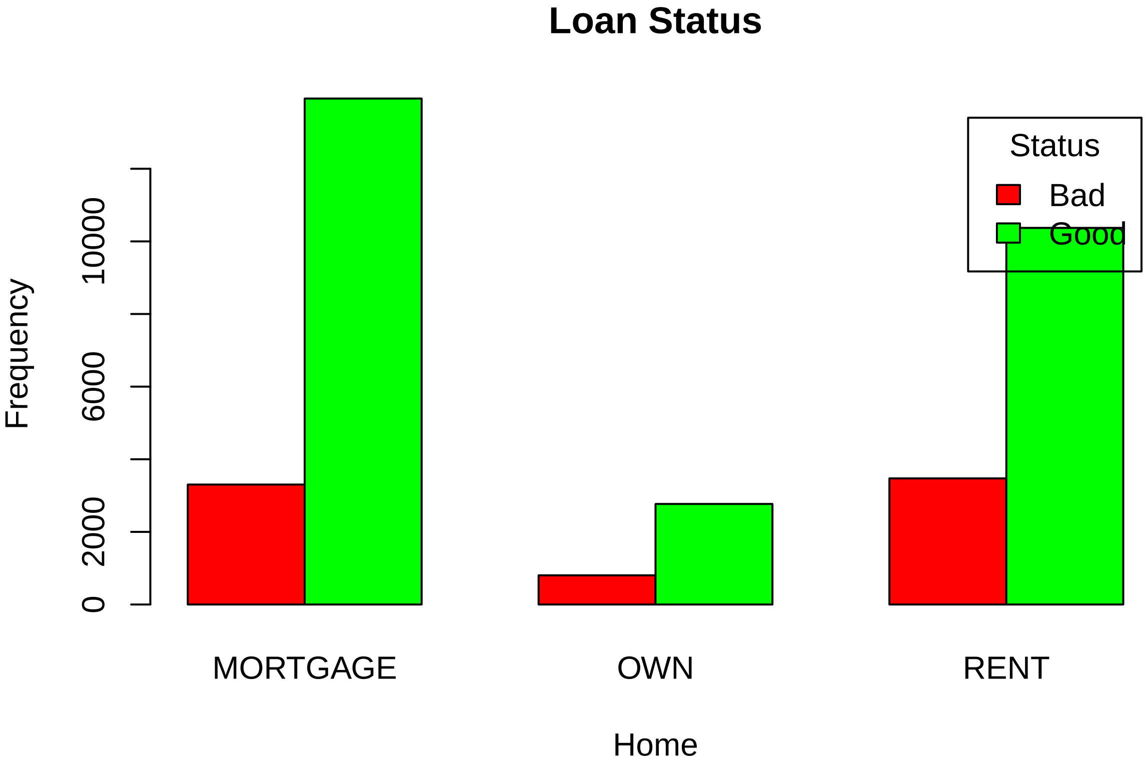
The distribution of income for good and bad loans were strongly skewed to the right with outliers.



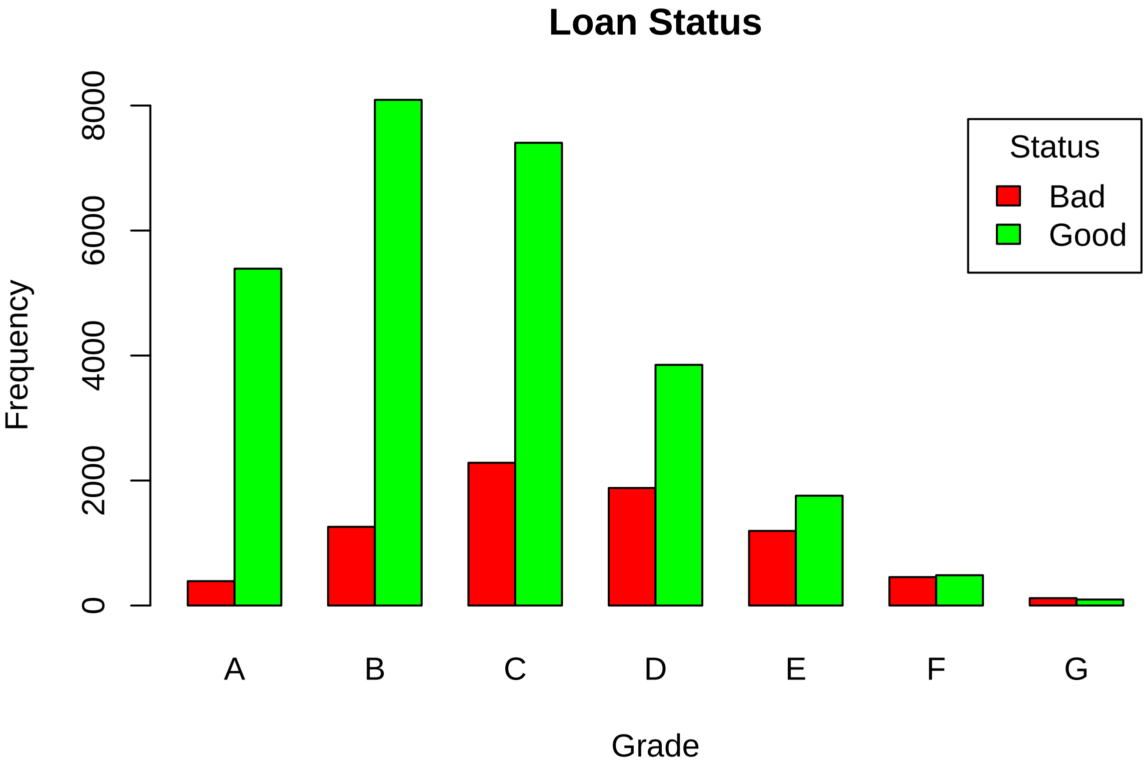
The distribution of the term for the majority of loans were in good status at 36 months.



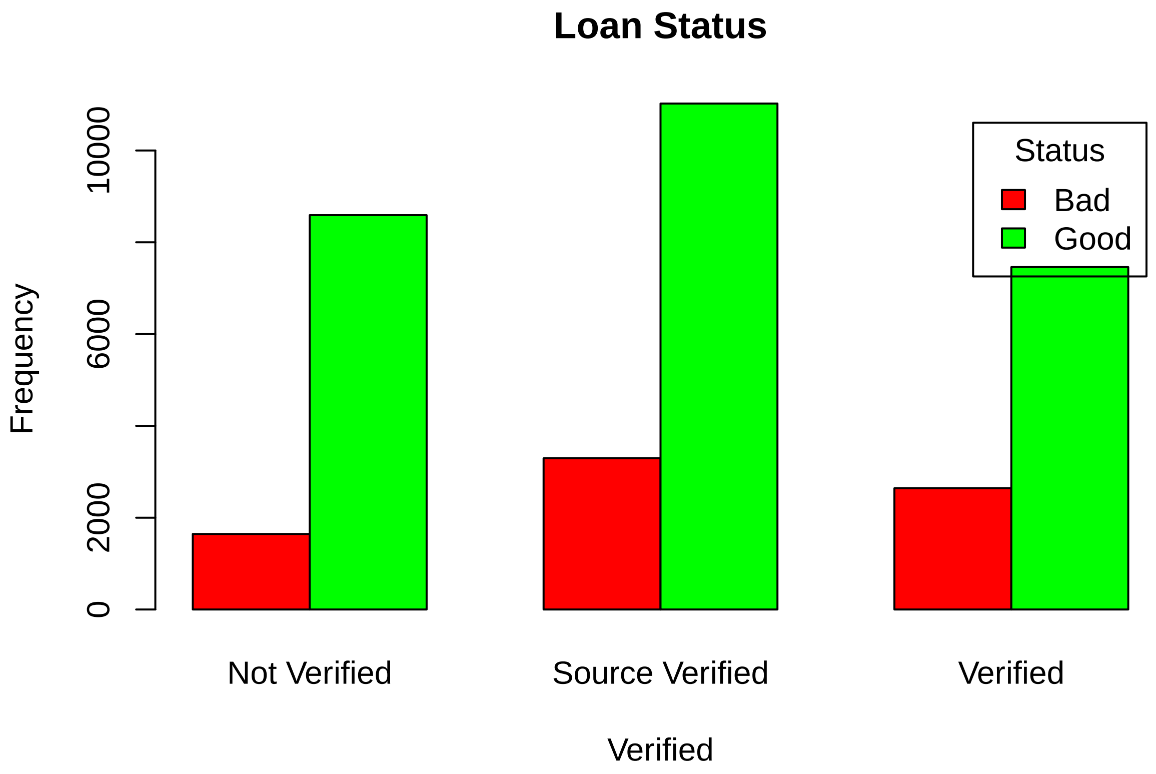
The distribution of the length of loans were approximately 2 years and the majority appeared to be in good status.



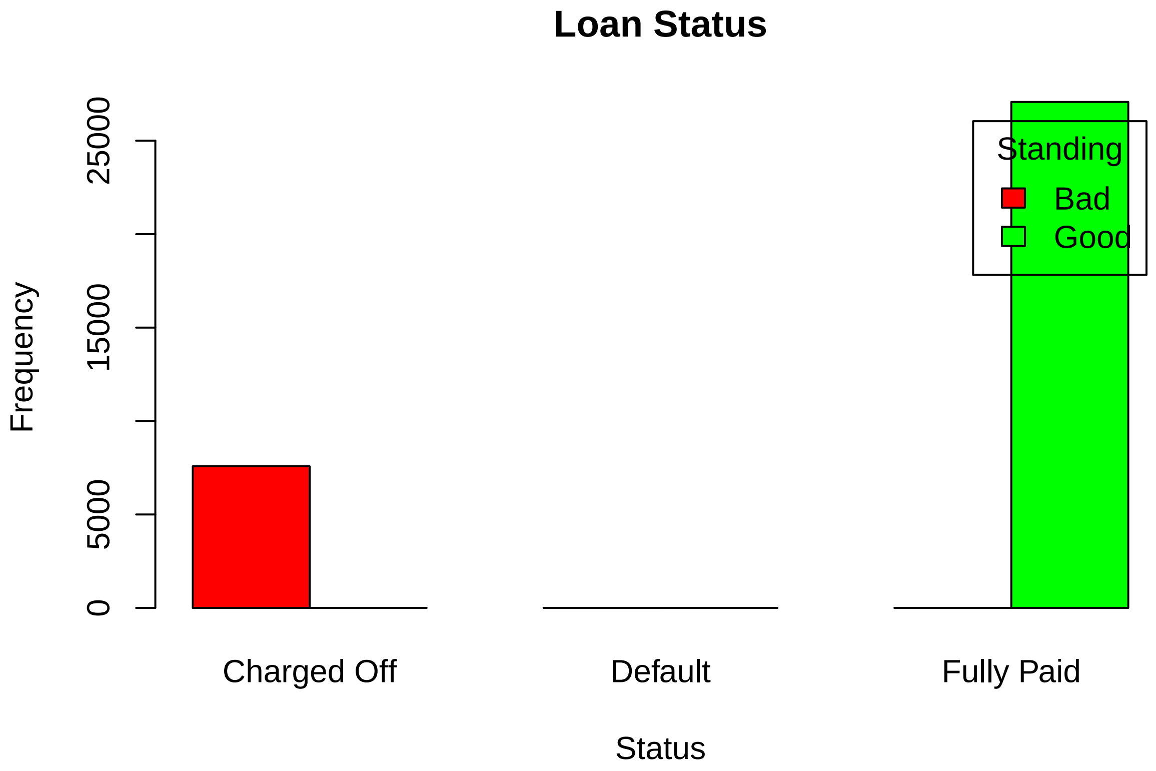
The distribution of home loans were mostly for mortgages and the majority appeared to be in good status.



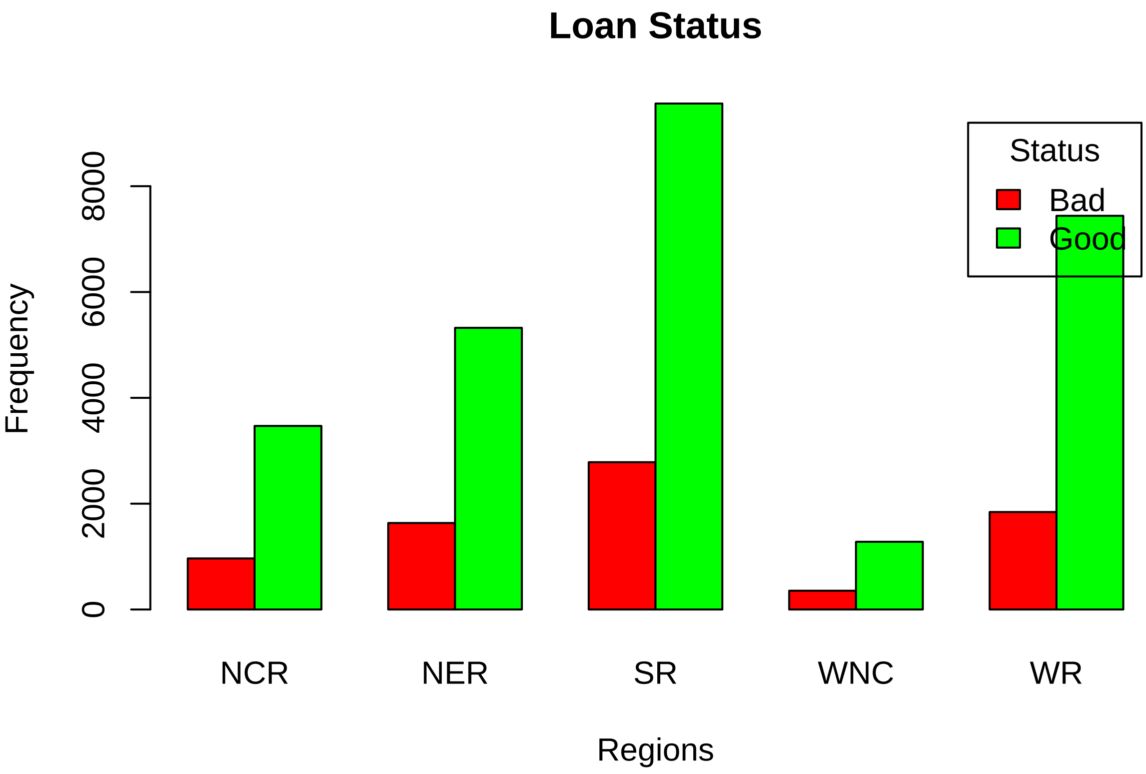
The distribution for the grade of most loans was a B and the majority appeared to be in good status.



The distribution of verified for most loans was Source Verified and the majority appeared to be in good status.



The distribution of the status of loans were fully paid and the majority appeared to be in good standing.



The largest distribution of loans by region were located in the Southern Region of the U.S. and majority loans appears in good standing.

The Logistic model

```
i <- 1
pred_all <- c()
total_acc <- c()
total_acc_badloans <- c()
total_accgoodloans<- c()
for (i in 1:length(threshold)){
  print(i)
  pred_test <-ifelse(prediction > threshold[i],1,0)
  x<- as.matrix(table(pred_test, test.data$status_new))

  total <- x[3][1] + x[4][1]+x[2][1]+x[1][1]

  total_acc_temp = (x[1][1]+x[4][1])/total
  total_acc_badloans_temp =x[1][1]/(x[1][1]+x[2][1])
  total_accgoodloans_temp= x[4][1]/(x[4][1]+x[3][1])
  total_acc[i] <- total_acc_temp
  total_acc_badloans[i] <- total_acc_badloans_temp
  total_accgoodloans[i] <- total_accgoodloans_temp

  pred_all[[i]] <- pred_test


}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
```

```
accuracy <-as.data.frame(cbind(threshold,total_acc,total_acc_badloans,total_accgoodloans
colnames(accuracy)<- c("threshold","total_acc","total_acc_badloans","total_acc_goodloans

orig_dataset$totalPaid <- expm1(orig_dataset$totalPaid)
orig_dataset$amount <-expm1(orig_dataset$amount)

length(pred_all)
```

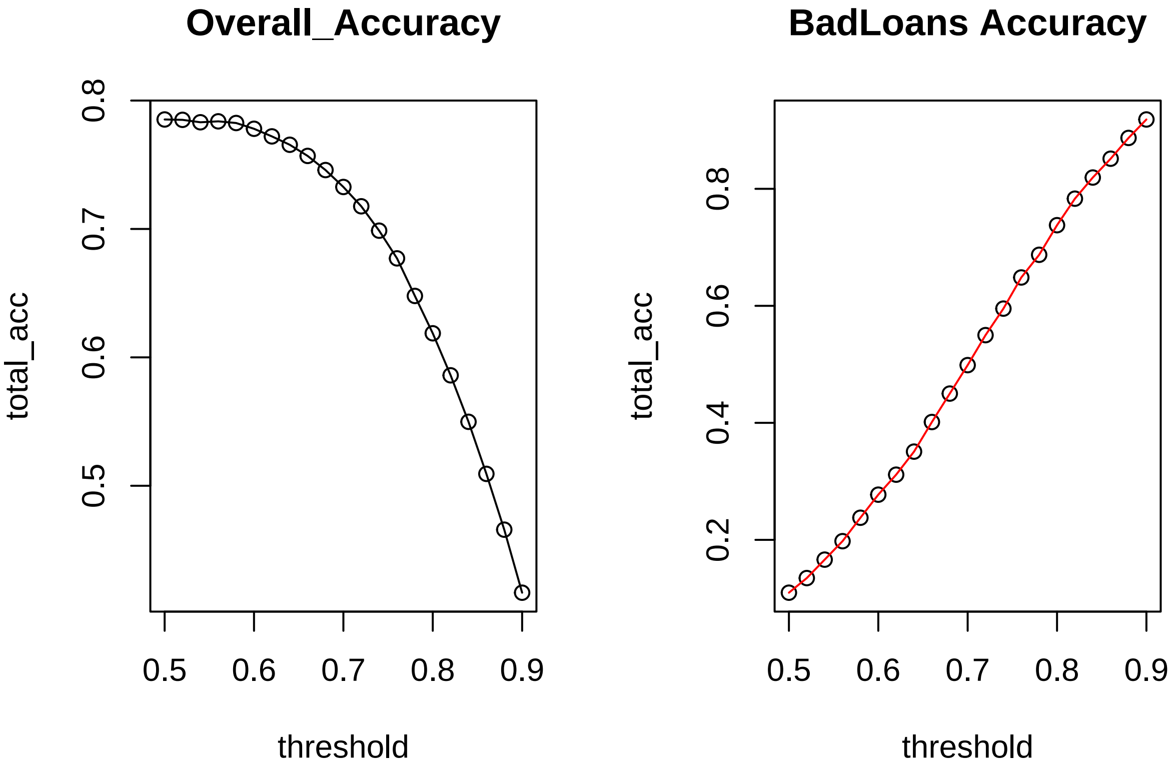
```
## [1] 21
```

```
profit <- c()
disbursed_loans <- c()
for (i in 1:length(pred_all)){
  #i <-1
  test2 <-orig_dataset[ind==2,]

  test2$threshold<- pred_all[[i]]
  test2$profit <- test2$totalPaid - test2$amount
  proft_temp <- test2[test2$threshold == 1,]#calculating on customers whose prob is gre
  l <- nrow(proft_temp)
  p <- sum(proft_temp$profit, na.rm= TRUE)
  profit[i] <- p
  disbursed_loans[i] <- l

}

accuracy$profit <- profit
accuracy$disbursed_loans <-disbursed_loans
```



Optimizing Threshold for Accuracy

We created two sets of data from the original dataset (training data & test data). The training dataframe (DF) contained 80% of the original sample, while the testing DF contained 20% to generate our model to predict statuses for each loan and analyze the performance (accuracy) of our model.

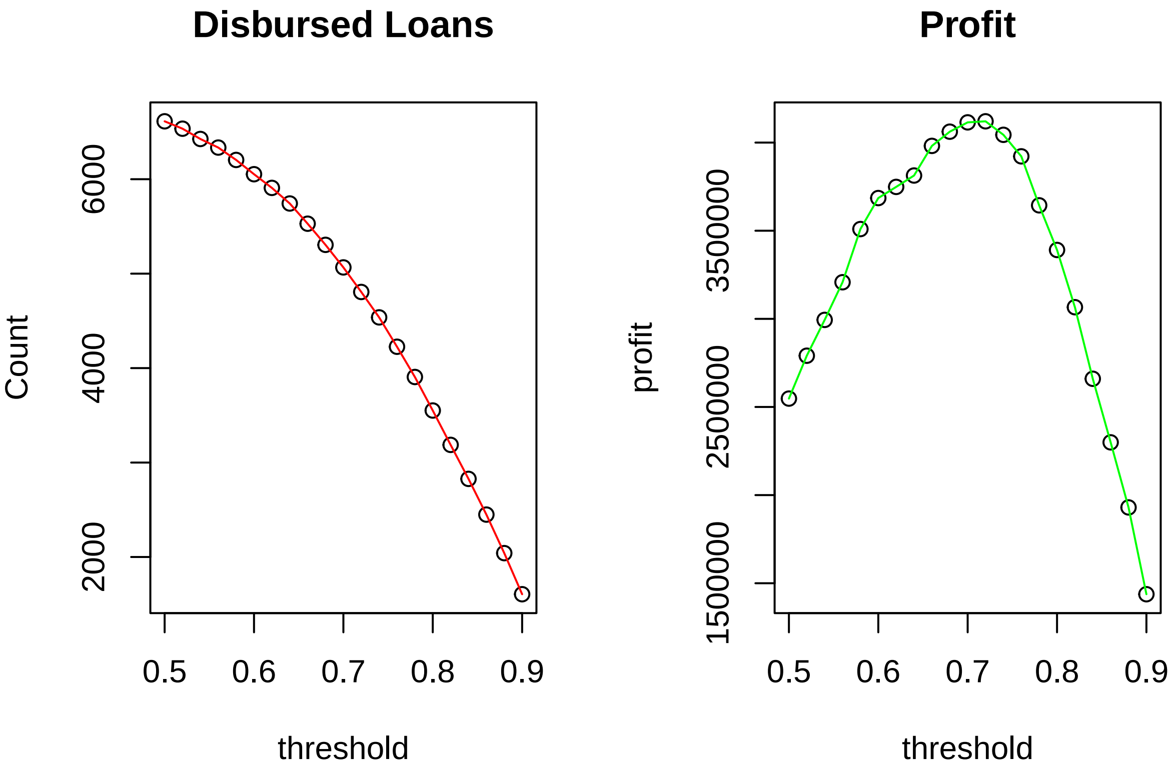
The dataset is not balanced since there are more good (27074) loans than bad (7581) loans. We then fit the model using our predictors on the training data in order to predict loan status. For the analysis, we oversampled to increase the number of bad loans to match the number of good loans, while keeping the original row number of bad loans in the training data. This led to a final training dataset of 27,747 total rows (6,058 bad loans and 21,689 good loans).

Predicting bad loans was the goal and that's what we did. At this point, we could move to automatic model selection to check for problems of collinearity using the variance inflation factor (VIF) on our full model. However, we suspect that some of the variables might produce a high VIF, but in this project, we are uninterested in whether the estimated regression coefficient increases if our predictors are correlated. Instead, we focused on loan status prediction - predicting customers who were likely to default on their loans. The results revealed that as threshold increased, the overall accuracy decreased. On the other hand, as threshold increased, the accuracy to predict bad loans increased.

Optimizing Threshold for Profit

The best profit threshold was 0.72 and the overall accuracy and percentages of correctly predicting good loans was 77% for good loans and 55% for bad loans, based on an accuracy of 72% (all the calculations and graphs are based on the test data).

As threshold increases (90%), total accuracy would only be 42%, the accuracy of predicting the total amount of bad loans would also increase (92%), and we see an inverse proportion in total accuracy of good loans as they would decrease (27%). However, the bank could make less of a profit (143,7690) and disburse fewer loans than it would at a lower threshold. The maximum profit threshold did not coincide with the maximum accuracy threshold. In other words, as the maximum accuracy threshold increased, for instance (0.90), profits decreased. Compared to not using our model, a bank could expect to make $1,299,446 in profits. Alternately, the maximum profits from using our model is projected at $4,120,756.



Results Summary

We created a statistical model to assist financial institutions in the decision making model to underwrite customers in the loan approval process. This included a classification threshold (probablity of a customer to pay their loan) to be used to predict loan status and make the most profits for a bank. We built a logistic regression model to predict good loans (fully paid off) and bad loans (charged off/defaulted). We also found a feasible classification threshold to be used to make the most profits for the bank. The model revealed the best or highest threshold for profits at 0.72. It predicted good loans with an accuracy of 77% and bad loans with an accuracy of 55%. The overall accuracy of the model was 71.8%.

Nonetheless, there is a tradeoff between accuracies to predict good and bad loans. If we increase the threshold value, we would be able to predict a higher percentage of bad loans, but would inversing predict a lower percentage of good loans/low risk loans. Ultimately, the bank would make less profits and disburse fewer loans. It all depends on whether a bank wants to be risk averse or risky with the potential to make more profits.