Project Part 3: Predicting Loan Defaults with Logistic Regression

Prepared by Phil Callahan

Executive Summary

This report provides an analysis and proposed predictive model for improving the existing process of selecting loan applicants for this bank. Methods of analysis include historical data cleaning, exploring, and transformation as well as logistic regression for model selection and predictive analysis. Thirty-two variables such as home ownership and debt-to-income ratio were assigned predictive power and kept in the model if they were found to improve its overall predictive ability. The final model was trained on 80% of the bank's provided historical data, then validated on the remaining 20% for accuracy and profit optimization. All calculations and code can be found in the included supplemental files.

When compared with the bank's current loan selection methods through historical data (50,000 individual loans in total), results of this report's final predictive model show a profit improvement of nearly 125% over existing methodology (see Figure 1 below).

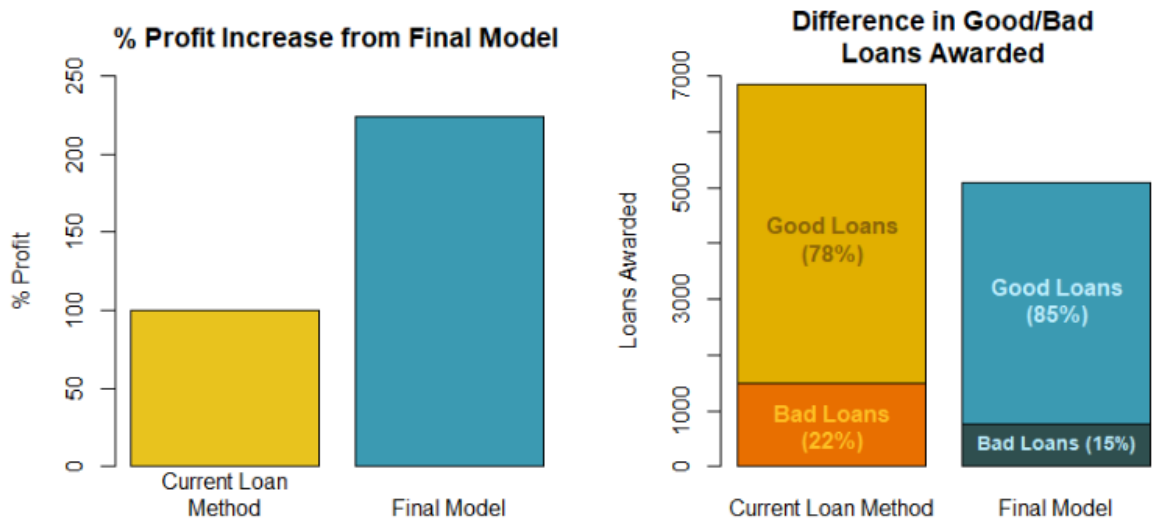


Figure 1 – This report's proposed model more than doubles profit (224%) over existing loan selection methodology thanks, in part, to the redistribution of "Good" vs "Bad" loans awarded.

The resulting model was able to increase profit by both awarding fewer unprofitable "Bad" loans (cutting them from 22% to 15%) and awarding a higher percentage of profitable "Good" loans (increasing them from 78% to 85%) than the current loan selection methodology. Increasing the efficiency of lending also had an overall effect of decreasing the total amount of loans awarded overall, thereby decreasing labor and overhead cost as well, though that metric was not factored into this report (see second graph in Figure 1).

It is recommended that this bank incorporate use of this report's predictive model for its future lending practices immediately to increase efficiency and profit margins.

Section 2 - Introduction

The problem to solve: Given a dataset containing information on 50,000 loans, create a predictive algorithm to determine which applicants are likely to default on their loan payments.

The given dataset has 32 variables to use as predictors. Using the programming language R, the task will be to create the best predictive model with the best predictor variables using logistic regression. The data will first be cleaned to prepare for model creation. The model will be created only from loans with a status of Charged Off, Current, or Fully Paid (removing all loans in the status of Default, In Grace Period, or Late). Secondly, predictors thought to have low predictive qualities will be removed from the data set. Next, categorical variables with too many categories will be consolidated to fewer groupings. After that, rows with blank values will be removed if sufficiently small (acceptable percentage so as not to affect the overall predictability of the data. Finally, heavily skewed data will be transformed to attempt to make more normal.

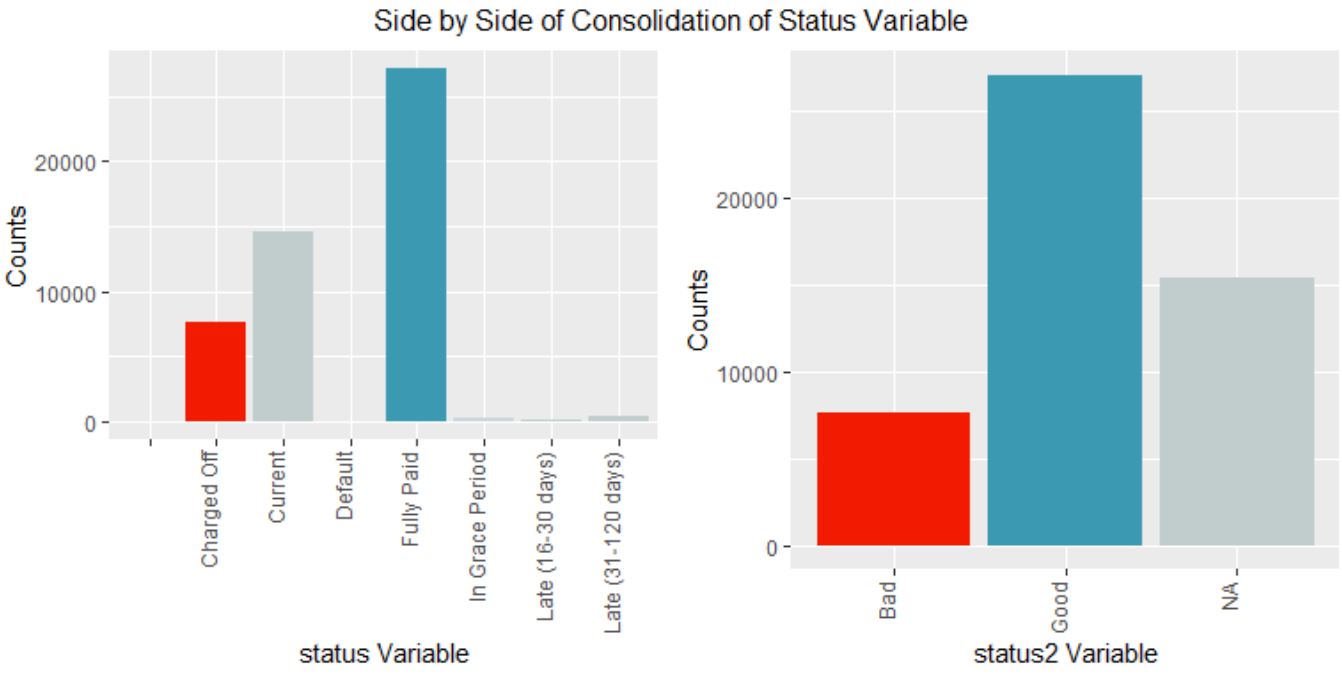
After the dataset is cleaned, a suitable model will be created through forward step methodology with the available predictor variables from a training subset of 80% of the dataset that is left. The remaining 20% will serve as a validation subset for the model. From this validation, the best threshold for profit and accuracy will be accepted as the final model and presented to the client.

Section 3 - Preparing and Cleaning the data

Several things had to be done to clean and prepare the data set for analysis. Firstly, a response variable, *status2,* was developed consisting of the two levels “Good” or “Bad” from the variable *status*. All loans classified as “Fully Paid” were assigned as “Good”, all loans classified as “charged off” or “default” assigned as “Bad”, and the rest of the classifications were removed using the following code chunk in R.

```
#we'll use mutate to create the dataframe we want for the predictors
loan_predictors <- loans %>%
  select(-totalPaid, -loanID, -employment, -state, -bcRatio) %>% #removes
uneeded columns
  #conditionals for assignment to "Good", "Bad", NA
  mutate(status2= ifelse(status == "Fully Paid", "Good",
                  ifelse(status == "Charged Off" | status == "Default",
                         "Bad", NA)))%>%
  select(1:status, status2, everything()) #puts status2 col next to old sta-
tus col for easy comparison
```

This results in the following consolidation of data points:

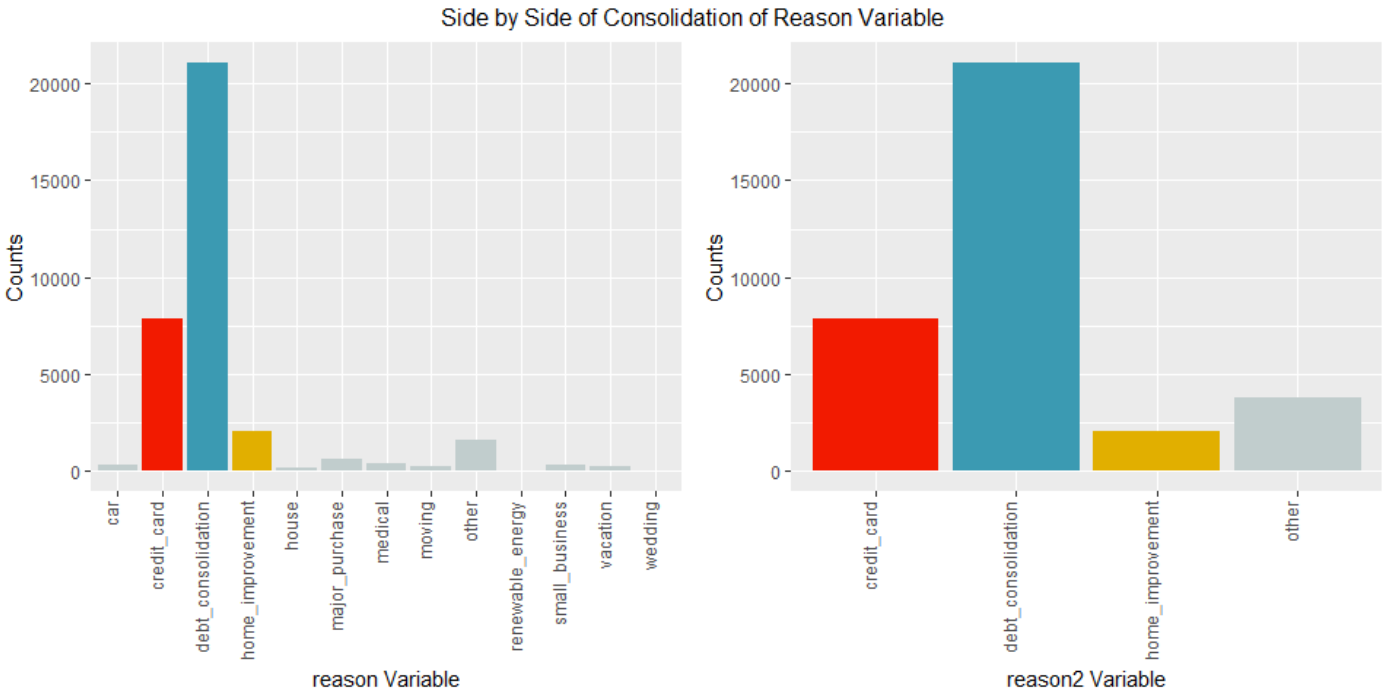


Note also that the variables (columns) loanID, employment, and bcRatio were removed with the previous code (See Table 1 for justifications).

Variable (column) Removed	**Justification**
loanID	This is a unique identifier that's highly unlikely to have any predictive attributes.
employment	The categories are too granular to be effective for overall prediction. For example, even if one occupation was a good predictor, say Financial Analyst, there are only 48 occurrences.
bcRatio	This variable was removed because the categories that derive the ratio are still in the dataset (totalBal and totalLim) causing unnecessary redundancy. Additionally this category had a relatively high occurrence of NAs so wasn't worth keeping in the data frame.

Table 1 - Removed variables and justifications.

Once the variables thought to have low predictive qualities had been removed and the data frame had been pruned to only Good and Bad loans, remaining features were engineered to be more useful. The variable *reason* had 13 categorical values which were consolidated to four into *reason2* which combined the smallest values into the category "*other*" (see plot below).



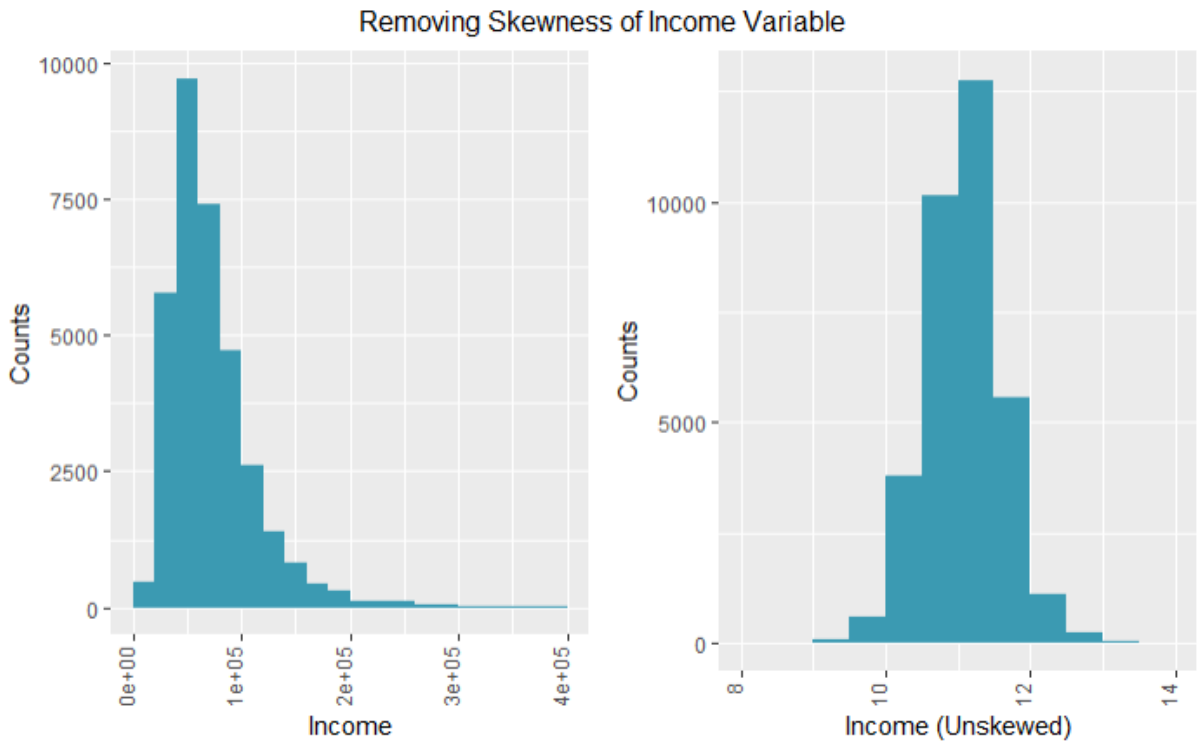
After the feature consolidation, there were still aberrant NAs to be dealt with before the dataset was ready for analysis. To do this, the summary() function was run to determine that the variables revolRatio and bcOpen still had 15 and 360 NAs, respectively. These rows were simply removed from the dataset since 347 (some rows contained NAs in both categories) is only about 1% of the entire dataset leaving it with a still robust 34293 data points.

Section 4 - Exploring and Transforming the data

With a clean dataset, the next step was to explore it, transform any skewed variables, then attempt to find any potentially interesting correlations to help predict loan status.

One of the predictors chosen for transformation was the *income* variable. Since it was positively skewed, it was transformed with a log() function (see the results in the side-by-side plots below).

Prepared by Phil Callahan



See below for other variables chosen for transformation (14 positively skewed variables in total). NOTE: Nine of the fourteen variables (denoted with " * " below) resulted in -Inf values which proved problematic during the modeling phase. This was overcome by using a log(x+1) transformation. Fourteen variables may seem like an unnecessarily large amount to keep, but when establishing predictive models, the more options (variables) available, the better.

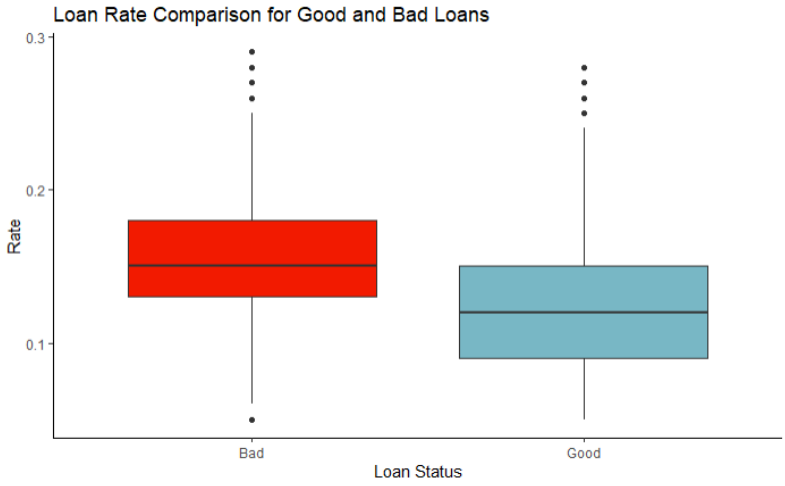
Skewed variable	***New, unskewed column added***
delinq2yr	delinq2yr_unskewed*
inq6mth	inq6mth_unskewed
openAcc	openAcc_unskewed
pubRec	pubRec_unskewed*
totalAcc	totalAcc_unskewed
totalBal	totalBal_unskewed*
totalRevLim	totalRevLim_unskewed

Skewed variable	***New, unskewed column added***
accOpen24	accOpen24_unskewed*
avgBal	avgBal_unskewed*
bcOpen	bcOpen_unskewed*
totalLim	totalLim_unskewed
totalRevBal	totalRevBal_unskewed*
totalBcLim	totalBcLim_unskewed*
totalIlLim	totalIlLim_unskewed*

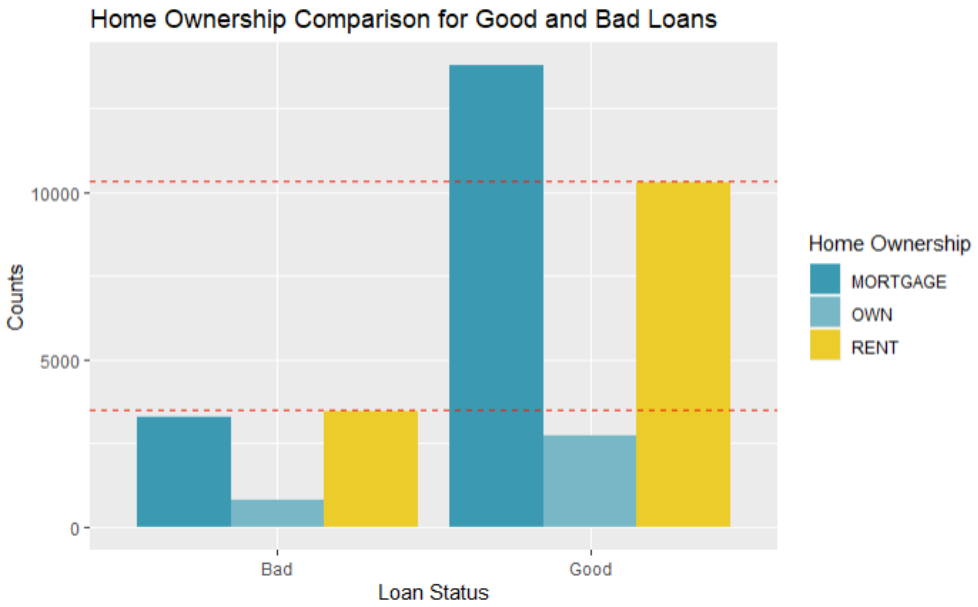
** denotes log(x+1) transformation*

After skewed variables were transformed, data exploration could be performed to find potentially interesting relationships. After looking for potential correlations between variables, e.g. *income* vs. *grade*, *income* vs. *status2*, etc., it was striking how little income influenced loan status. In fact, for most of the relationships, income tended NOT to influence loan status as dramatically as expected overall.

Some interesting relationships were seen however, with other variables' influence on status. For instance, the loan rate comparison between Good and Bad loans yielded a slight difference as well as the mean amount of credit checks within the past six months (0.8256294 for Bad loans vs. 0.661415 for Good loans). These are likely not strong predictors of correlation, however. For example, the rate comparison may just be a causal relationship from whatever criteria the lender already used to select the rate (e.g. credit score).



Perhaps the most compelling relationship found was between loan status and home ownership. Disregarding the obvious count differential (the sample sizes are much different), notice how much higher mortgage counts are than renting for Good loans vs Bad loans. In the Bad loans the mortgage counts are slightly lower than renters. This is an indicator of a potential correlation between having a mortgage and being a good borrowing candidate.



Section 5 – The Logistic Model

After the data was cleaned and explored, the dataset was ready to begin logistic model creation. From the categorical variable *status2* dummy variables were created and stored as column *status 3*. In this new column, 1 represented “Good” loans and 0 represented “Bad” loans. This was the basis for the model creation.

From this dataset, 80% of the loans (27,434 rows) were randomly sampled using R’s base sample() function to be used as the training data frame. The remaining 20% (6,858 rows) were then stored in another data frame for validation of the model later. The model was fitted using the following code chunk on the 80% “training” data frame utilizing forward step methodology:

```
#fit model with all predictors
loan.out.all <- glm(status3~., data=df80training, family=binomial)
summary(loan.out.all) #summary to look at data

#fit null model
loan.null <- lm(status3~1, data=df80training) #when stepping forward this
creates the minimal null model (zero predictors)

#forward stepwise procedure
step(loan.null,scope=list(lower=loan.null,upper=loan.out.all),direction="forw
ard")
```

The resulting model used the following 22 predictors to predict *status3*: grade of loan (*grade*), loan term (*term*), ratio of monthly non-mortgage debt payment to monthly income (*debtIncRat*), total credit limits (*totalLim_unskewed*), how many accounts were opened in the past 24 months (*accOpen24*), state, home, length, revolRatio, delinq2yr, totalAcc, payment, amount, sum of credit limits from all credit lines (*totalRevLim_unskewed*), number of open credit lines (*openAcc*), unskewed number of credit checks in the past 6 months (*inq6mth_unskewed*), total credit balance except mortgages (*totalRevBal_unskewed*), total current balance of all credit accounts (*totalBal_unskewed*), how many accounts were opened in the past 24 months (*accOpen24_unskewed*), total unused credit on credit cards (*bcOpen_unskewed*), total of credit limits for installment accounts (totalIlLim_unskewed), number of credit checks in the past 6 months (*inq6mth*).

While this report will utilize classification tables with the 20% of loans set aside as the “test” dataset to evaluate the predictive ability of the final model, the Hosmer-Lemeshow Goodness of Fit test (HL test) was used to evaluate the fit. The HL test can be thought of as having a null hypothesis stating that the model fits the data and an alternative hypothesis stating it doesn’t. The initial model above has an HL p-value of 0.325 meaning there is insufficient evidence to say that this model does not fit the data. Overfitting was not a concern with this particular dataset since the training/testing paradigm mitigates the effect.

Since the HL test confirmed adequate fit, it was decided to go no further on model development at this point and test performance. Pending satisfactory performance, this would remain the final model chosen.

Contingency tables of the bank's current methodology versus the final logistic regression model can be seen below. A threshold of 0.5 was used for the final model since that is its maximum accuracy threshold.

```
Initial Model Contingency Table

       Bad Good  Sum
  0      0 1500 1500
  1      0 5358 5358
  Sum    0 6858 6858
[1] "Proportion correctly
predicted = 0.781277340332458"
```

Old vs. Proposed

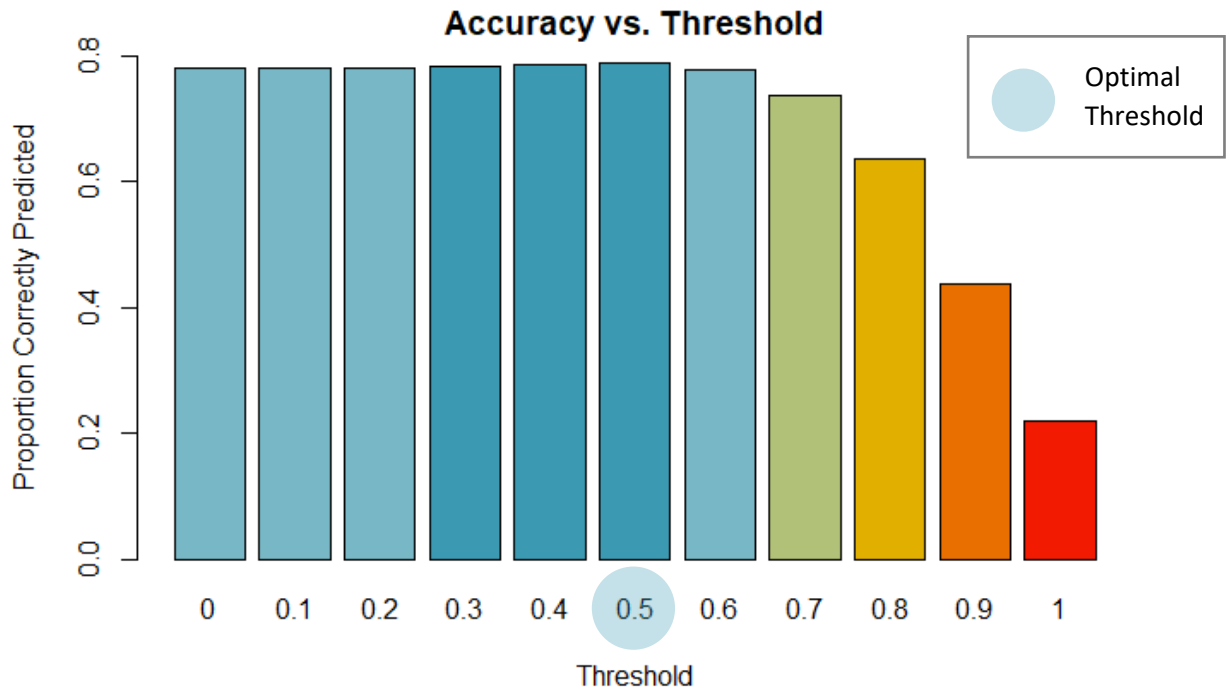
```
Final Model Contingency Table

      Bad Good  Sum
  0   209 1291 1500
  1   161 5197 5358
  Sum 370 6488 6858
[1] "Proportion correctly
predicted = 0.78827646544182"
```

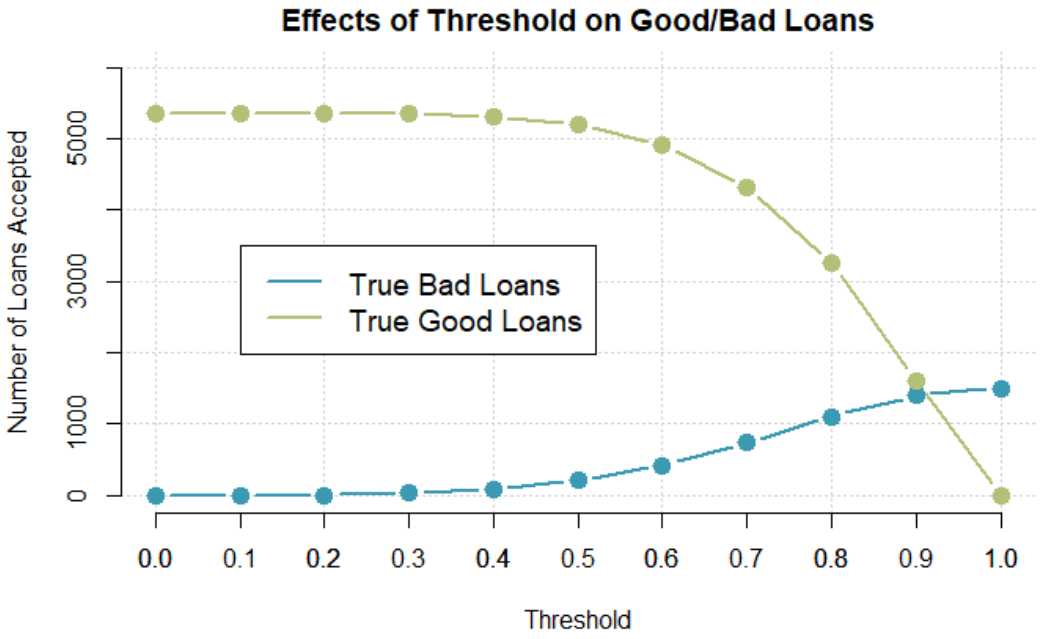
As can be seen from the above output, the overall accuracy of the final model is approximately 78.83%. As will be seen later, and in the executive summary, this is an effective model for predicting if a loan will be repaid. The criteria for effectiveness of the model will be accuracy and overall profit.

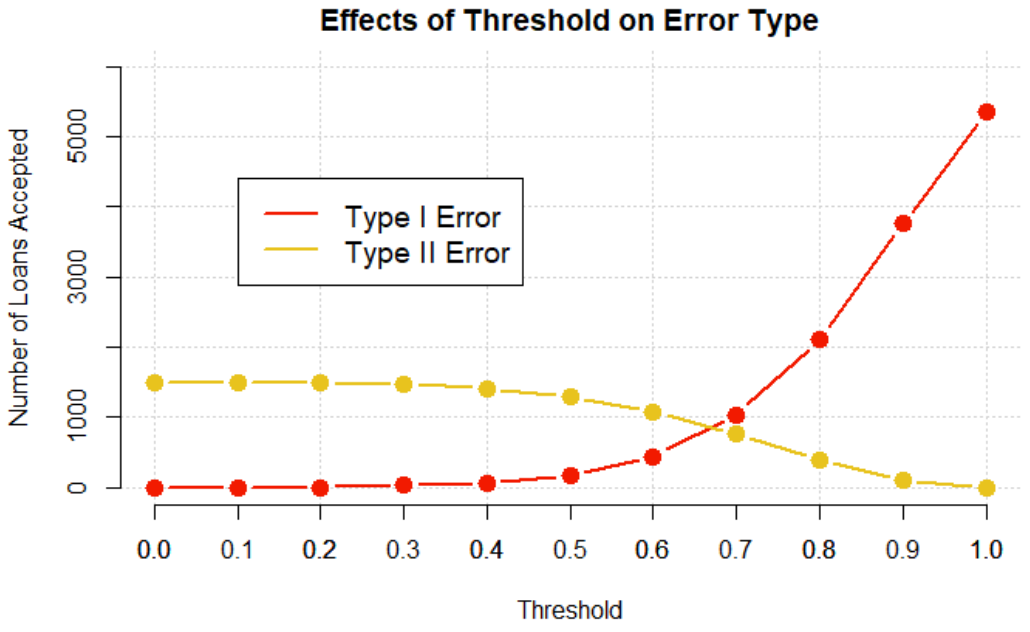
Section 6 – Optimizing the Threshold for Accuracy

Modifying the threshold to establish optimal accuracy, the model looked like this:



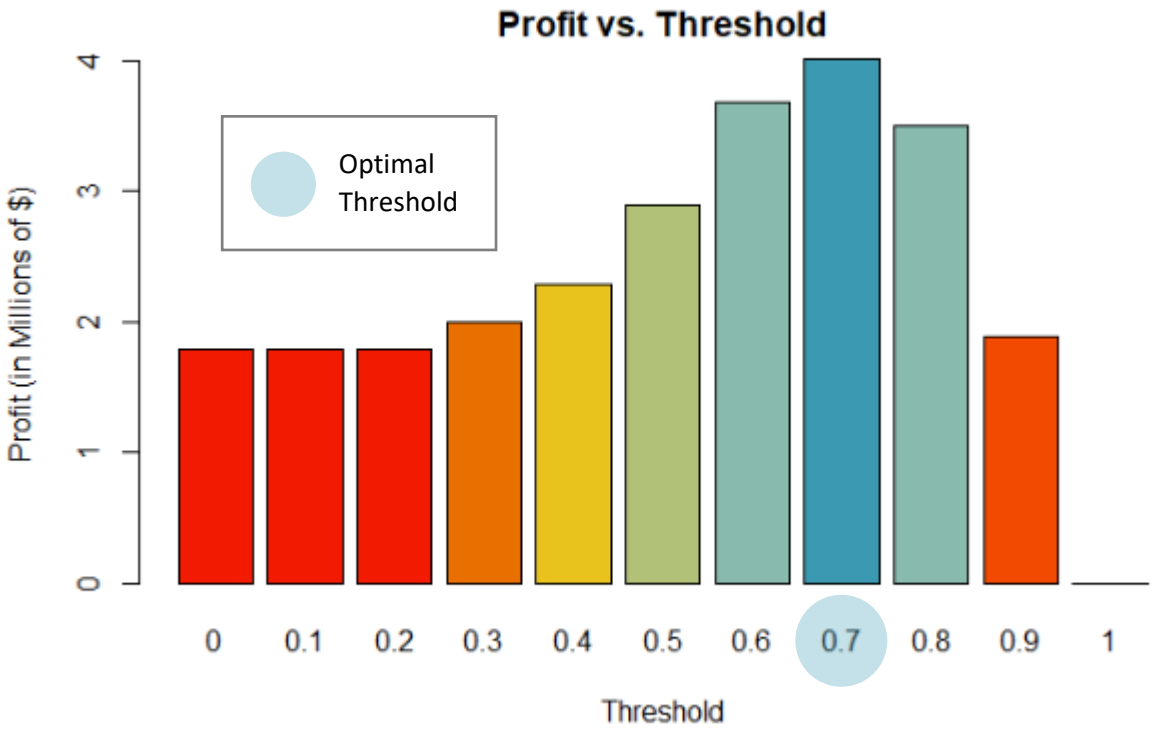
As can be seen from the previous graph, the optimal threshold for accuracy is 0.5, though the threshold varied little from zero to 0.6 compared with 0.7 to 1. This tells us that there is a tradeoff between correctly predicting good and bad loans (see below graphs). The model is more flexible having Type II errors versus having Type I errors. Accepting bad loans as good has very little effect on the accuracy, whereas rejecting good loans quickly affects the accuracy negatively (as can be seen on the right side of the graphs). If the final model were to use the threshold of 0.5 (optimal accuracy) the profit for the potential client (bank) would be $2,893,150 for the test data.





Section 7 – Optimizing the Threshold for Profit

Repeating this threshold analysis for profit, the result was a slightly higher threshold. A new data frame was created consisting of the probabilities from the training data and the profit per loan. This data frame was then further filtered by a threshold, then the profit column was tallied, resulting in an overall projected client profit for the model. The threshold vs. profit graph below shows the results of this analysis.



As can be seen in the previous graph, the threshold for maximum profit of the model is 0.7. This maximum threshold would yield a total profit for the potential client of $4,011,922 from the test data. This is a total profit increase of $1,118,772 over the optimal threshold for accuracy ($4,011,922 - $2,893,150 = $1,118,772) for the test data.

The current profit for the potential client is $1,789,901 (found by tallying profit from raw test data frame i.e. threshold of zero). **The profit increase for the potential client (bank) of deploying this model would be $2,222,021** ($4,011,922 - $1,789,901 = $2,222,021).*

While this model's profit increase is impressive, the profit of a perfect model, that is, a model that perfectly accepts only "good" loans, would be $12,526,418, or an increase of $10,736,517. This can be found by summing the total profit made by only loans with a status of *Good* from the test data frame.

* For the 20% subset of the whole dataset

As can be seen from the below contingency table at the model's best profit threshold (0.7), the overall accuracy is approximately 74% with approximately 49% (734/1500=0.489) of bad loans predicted correctly and approximately 81% of good loans predicted correctly (4320/5358=.806).

```
  Profit Contingency Table
       Bad Good Sum
  0    734  766 1500
  1   1038 4320 5358
  Sum 1772 5086 6858
[1] "Proportion correctly predicted = 0.73694954797317"
```

*Best **profit** threshold contingency table:*

Threshold = 0.7

The profit threshold is higher at 0.7 than the accuracy threshold at 0.5. The difference can be explained by the unprofitable (*bad*) loans in this range having more impact (more strongly negative profit). The higher profit threshold is accepting less unprofitable loans; hence the negative profit impact is lessened. Conversely, it could be that the profit model is not accepting as many weakly profitable loans (rather, categorizing fewer weakly profitable loans as *good*).

```
  Accuracy Contingency Table
       Bad Good  Sum
  0    209 1291 1500
  1    161 5197 5358
  Sum  370 6488 6858
[1] "Proportion correctly predicted = 0.78827646544182"
```

*Best **accuracy** threshold contingency table:*

Threshold = 0.5

It's important to point out at this point that all the previous calculations were performed a on a subset of the original data equaling 20% of the cleaned data as a whole. Any static profit increases or totals calculated would be applied to the rest of the 80% of the data. For example, the profit increase of $2,222,021 would be multiplied by five (100% / 20% = 5) to get the total profit for the entire dataset ($2,222,021 x 5 = $11,110,105). **In other words, the model would yield a 124% increase in profit for the potential client.**

This is computed as follows:

$$\frac{profit\ from\ proposed\ model\ test\ data\ set}{bank's\ actual\ profit\ from\ test\ data\ set} \times 100 = \begin{matrix} Percent\ profit\ of\ proposed\ model \\ (versus\ actual\ profit) \end{matrix}$$

$$\frac{\$4{,}011{,}922}{\$1{,}789{,}901} \times 100 = 224.14\%$$

Section 8 – Results Summary

The final logistic model used to predict good loans for the potential client (bank) used the following 22 predictor variables: grade of loan (*grade*), loan term (*term*), ratio of monthly non-mortgage debt payment to monthly income (*debtIncRat*), total credit limits (*totalLim_unskewed*), how many accounts were opened in the past 24 months (*accOpen24*), state, home, length, revolRatio, delinq2yr, totalAcc, payment, amount, sum of credit limits from all credit lines (*totalRevLim_unskewed*), number of open credit lines (*openAcc*), unskewed number of credit checks in the past 6 months (*inq6mth_unskewed*), total credit balance except mortgages (*totalRevBal_unskewed*), total current balance of all credit accounts (*totalBal_unskewed*), how many accounts were opened in the past 24 months (*accOpen24_unskewed*), total unused credit on credit cards (*bcOpen_unskewed*), total of credit limits for installment accounts (totalIlLim_unskewed), number of credit checks in the past 6 months (*inq6mth*).

The final model is approximately 74% accurate as well as having an overall profit increase for the potential client (bank) of 124.14% (or $2,222,021 total from the 20% test data). The threshold for maximum profit of the model is 0.7. As can be seen from the below contingency table, approximately 49% (734/1500=0.489) of bad loans predicted correctly and approximately 81% of good loans predicted correctly (4320/5358=.806).

```
  Profit Contingency Table
       Bad Good Sum
  0    734  766 1500
  1   1038 4320 5358
  Sum 1772 5086 6858
[1] "Proportion correctly predicted = 0.73694954797317"
```

*Best **profit** threshold contingency table:*

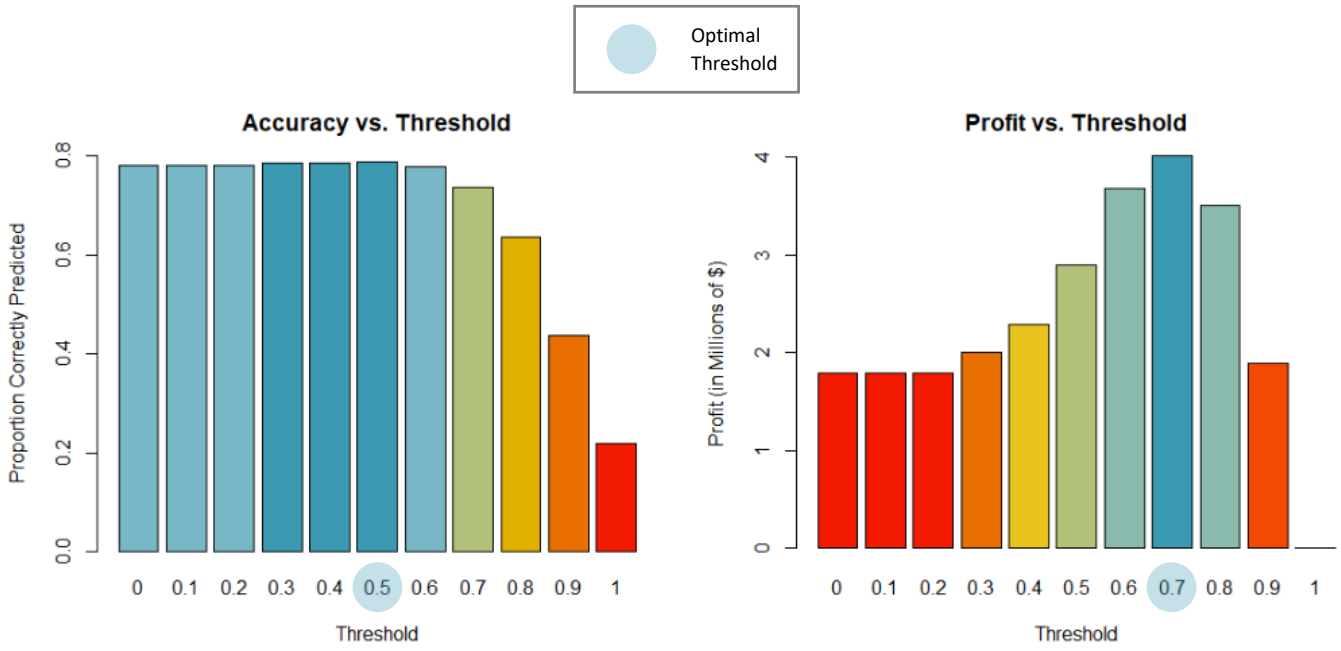
Threshold = 0.7

Future Improvement Potential

While this model will increase the profit of the potential client, there is plenty of room for improvement in future model design. As was computed in Section 7, a perfect model may be unattainable, but it would yield over $10 million more in profit from the test data (that would be $50 million when extrapolated to the other 80%!). This pool of money is tempting to try and convert to profit for the bank. For example, next steps could be trying methods such as backward step or exploring the regsubsets() function with a static number of predictors and using the best of them. Additional variable transformations could be attempted on the original skewed columns, then model fitting techniques could be reapplied to find possible increases in accuracy and profit. Though the accuracy and profit are adequate in the current model (124% profit increase), these improvements could increase one or both parameters.

Threshold analyses could be repeated with more granular values applied rather than increments of ten. Perhaps there are increases between the peaks. Further tools like McFadden's Pseudo-R-Squared Test for Logistic Regression could be used to help analyze the explanatory capabilities of the model. Utilizing tools like this as well as modifying the variables left in and removed, the model could likely be improved upon in the future. Higher order interactions as well as collinearity could be explored for combinations of variables.

Even without future improvements, the final model developed in this report could save the potential client (bank) a large sum of money immediately by preventing them from lending to applicants likely to default on their loans at the recommended threshold of 0.7 (see below graphs).



As mentioned earlier in the report, all the previous calculations were performed a on a subset of the original data equaling 20% of the cleaned data as a whole. Any static profit increases or totals calculated would be applied to the rest of the 80% of the data. For example, the profit increase of $2,222,021 would be multiplied by five (100% / 20% = 5) to get the total profit for the entire dataset ($2,222,021 x 5 = $11,110,105). **In other words, the model would yield a 124% increase in profit for the potential client (bank).**