Project: Predicting Loan Defaults with Logistic Regression

Kyle Krakoski

4/28/2021

Section 1: Executive Summary

This report provides an analysis of what characteristics of bank loan applicants would be most indicative of loan defaults, a model to most accurately predict if a loan will be good or bad, and a model to maximize profit from loans.

Before the calculations to find the most significant predictor characteristics were performed, the analysis first eliminated any characteristics that could be considered illegal, unethical, or impractical to use. Additionally, to keep the model simple, only the ten most significant predictors were used. A full summary of predictor choices and calculations used can be found in Sections 3, 4, and 5.

To determine the results of the model, "threshold" values between 0 and 0.95 were tested. These thresholds determined if a loan was classified as "Good" or "Bad". If the model output a value below the threshold value, the loan was considered "Bad", and if the model outputted a value above the threshold value it was considered "Good." Different threshold values were tested to find the most accurate model and most profitable model. A full summary of how threshold values were used and the calculations can be found in Sections 6 and 7.

The variables I chose to pre-eliminate, and why, are in Section 3. For model accuracy, without the model, loan officers were 78.46% accurate in determining "Good" loans. A graph of all accuracy values by threshold value can be found in Section 6. For profit, a graph of percent profit increase after using the model can be found in Section 7.

The ten most significant predictors were amount, term, payment, grade, debtIncRat, delinq2yr, inq6mth, totalAcc, accOpen24, and totalLim. For model accuracy, the most accurate model was at a threshold value of 0.5. The accuracy at this threshold was 79.17%, Additionally, the profit with this model was 70.83% greater than if no model were used. For maximum profit, the best profit model was at a threshold value of 0.7. The profit using this model was 117.18% greater than if no model were used. However, the accuracy was only 74.61%, 3.86% lower than if no model were used.

A couple decisions need to be made. First, to choose the model based on accuracy or profit. Second, decide if you'd like to have a more complex model by adding predictors that were cut. Finally, one limitation of this model is the number of predictors used. More predictors might result in a better profit model but complicate it. Additionally, another limitation is the most optimal threshold value might not be at a 0.05 interval but testing smaller intervals would take exponentially longer.

Section 2: Introduction

The goal of this project is to find what characteristics of applicants would be most indicative of loan defaults. This means this paper will be focusing mostly on loan defaults (and not what makes a good loan or maximizing good loans, for example). The first step is to clean the data in the loans50k.csv file. This involves eliminating the variables that would obviously not affect if a loan is defaulted on and variables that would be unethical to use to make a decision. Next is to change the loan status to either "Good" ("Fully Paid" loans) or "Bad" ("Charged Off" or "Default" loans). Ongoing loans ("Current", "Late …", and "In Grace Period" loans) were removed from analysis. After that I analyzed each possible predictor variable and consolidated categories within them or redefinied them as I saw fit. The last step for data cleaning was to eliminate rows with empty or NA values or impute those values. If the number of cells in a column with these values was statistically insignificant (I used a threshold of 1% of the total number of rows), I removed rows with those values. If the number of cells was over 1% of the total number of rows I imputed the values. The specifics of each step will be described in Section 3.

Once the data was cleaned the distributions of the predictor variables were analyzed and graphed. I analyzed the distributions of the variables for all loans and then for just "Good" loans and just "Bad" loans. Since we're focused on finding predictors for loan defaults, I'm most concerned about the distributions of "Bad" loans compared to the distributions of "Good" loans and all loans. For categorical variables, tables and bar graphs were used. For numerical variables density plots were used. I specifically chose to use density plots over histograms to better see the changes in proportions of "Bad" loans compared to all loans or "Good" loans. A large shift would mean that variable is a better predictor of "Bad" loans. In order for regression to be more accurate, normal distribution is desired. To get distributions as close to normal as I could, some variable values were transformed. For right-skewed variable distributions natural logs or roots of the variable values were taken to shift the distributions. For left-skewed distributions powers of the variable values were taken to shift the distributions. The specific transformations and conclusions will be discussed in Section 4.

Section 3: Preparing and Cleaning the Data

The variables I chose to remove were loanID, employment, employment length (length), home status, state, and total amount paid to the bank (totalPaid). I eliminated loanID because it's just used as an identifier and obviously does not affect loan default chance. I eliminated employment because there's far too many different values to make any meaningful analysis on. I eliminated employment length (length) because there could any number of reasons someone could be unemployed for a length of time (school, layoffs, job transitioning, etc.) and income is far more important. Additionally, I believe sudden unemployment would be more of an issue once the loan is already granted and not of our concern. I eliminated the "home" variable because whether someone rents or mortgages a house depends a lot on area they live in, which would be discriminatory to make a decision about. I do believe someone with an "Owned" home status would be more likely to payoff their loan(not having a monthly rent or mortgage to pay), but since we're just concerned about “Bad” loans I did not feel it was necessary to include. I removed the “state” variable for similar reasons- it would be unethical (and also impractical) to make a decision based on the state an applicant would live in. Finally, the variable of total amount paid to the bank (totalPaid) was eliminated due to the variable not being determined until after a loan was already granted.

The process for the next step, changing loan statuses to “Good” or “Bad” and removing the ones not fitting these categories, was described in the intro.

Next, for the “grade” variable, I changed the letter grades to numbers in order to graph a density plot of the data.

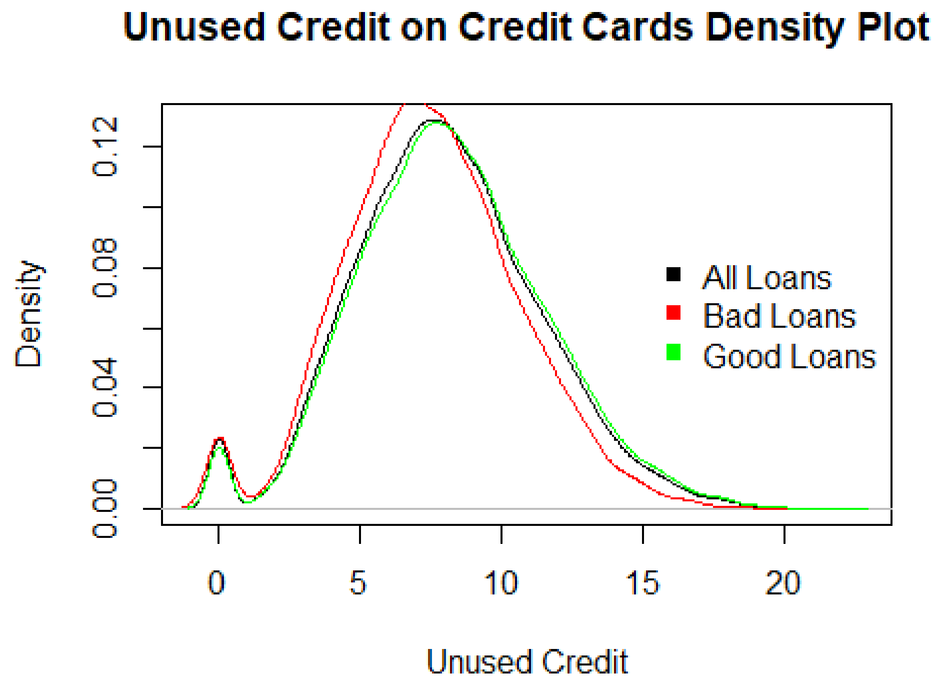
I consolidated some of the categories with low counts (under 200) in the “Reason” variable into the “other” category. I also consolidated “Source Verified” and “Verified” categories together in the “verified” variable.

The final step was to eliminate or impute empty and NA cells. Using my rules explained in the Introduction, the “revolRatio” NA cells had their rows eliminated while the “bcOpen” and “bcRatio” variables had their cells imputed.

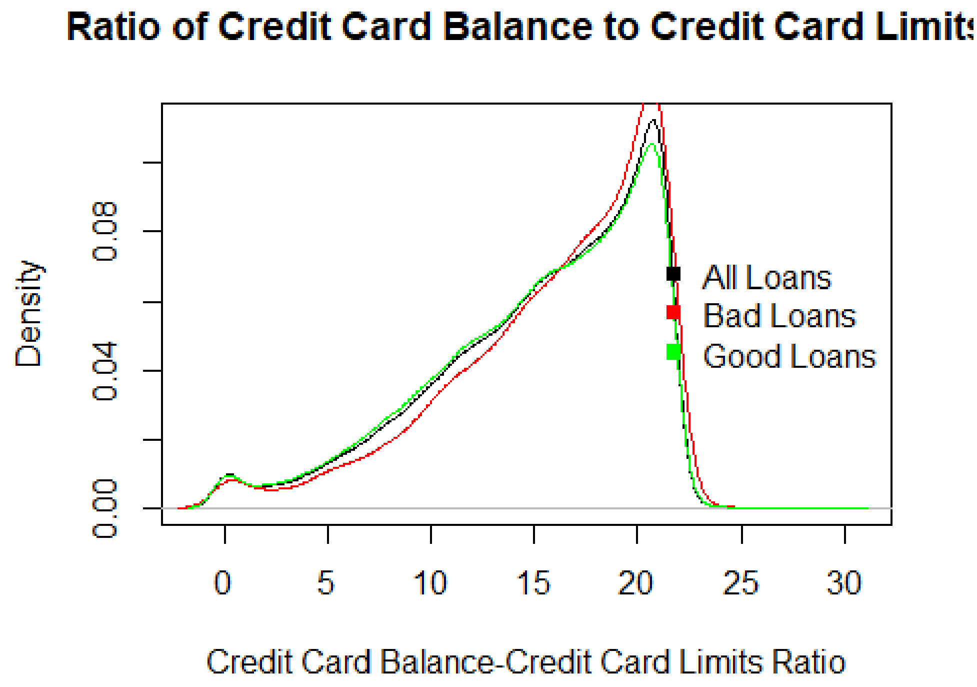
After all the data cleaning, I was left with 27 variables out of 32, and 34,640 loans out of 50,000.

Section 4: Exploring and Transforming the Data

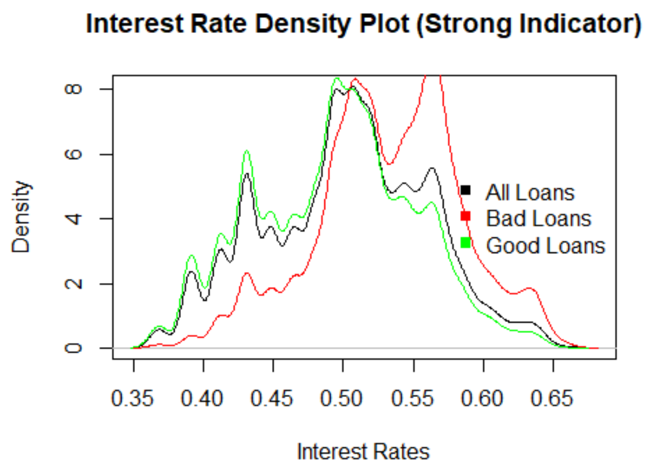
Most of the numerical data I had required some type of transformation to reach a normal (or nearer to normal) distribution. These transformations were either natural logs or roots to fix right-skewness, and powers to fix left-skewness. Some variables required both, as taking the log of a variable would push it too far into left-skewness. The variables requiring one of these transformations were: amount, rate, payment, grade, income, delinq2yr, openACC, pubRec, totalAcc, TotalBal, TotalRevLim, AccOpen24, avgBal, bcOpen, bcRatio, totallim, totalRevBal, totalBcLim, and totalIlLim.
An example of one of these transformed graphs that turned out well is the “Total Unused Credit on Credit Cards (bcOpen)” graphs:

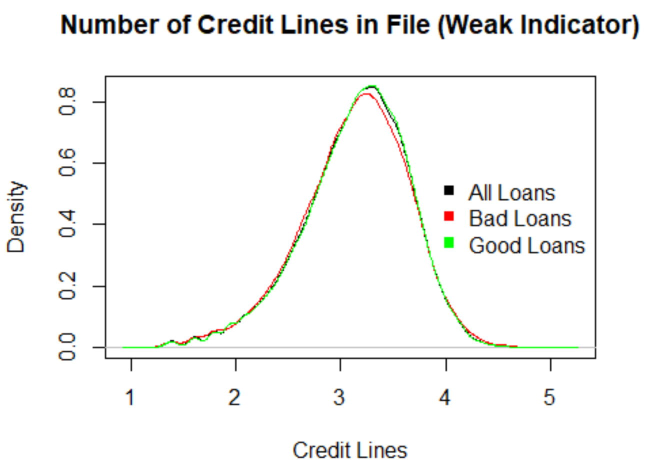


However, some of these variables were difficult to transform into a normal distribution, such as the "ratio of total credit card balance to total credit card limits (bcRatio)" variable:

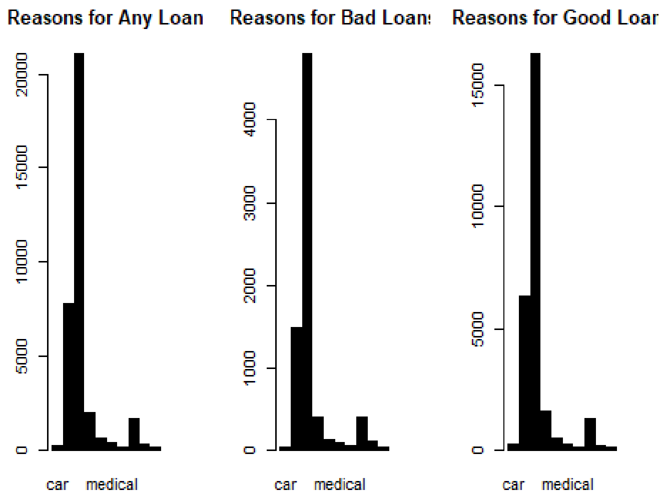


The only two numerical variables that didn't require transformations were the debtIncRat and revolratio variables. Once all the numerical variables were put into a normal distribution (or as close as I could get) I inspected how the "Bad" loan distrubutions differed from the "Good" loan and total loan distributions. A major shift in the "Bad" loan distribution indicated that it was likely that variable was a strong predictor of a loan default. A "Bad" loan distribution with an insignificant shift from the population distribution was an indicator that the variable probably wasn't a good predictor of a loan default and just mirrored population size. Both types of "Bad" loan distributions are graphed below:





For categorical variables I used a bar graph or just a table. In fact, the "reason" variable was the only variable I used a graph for:

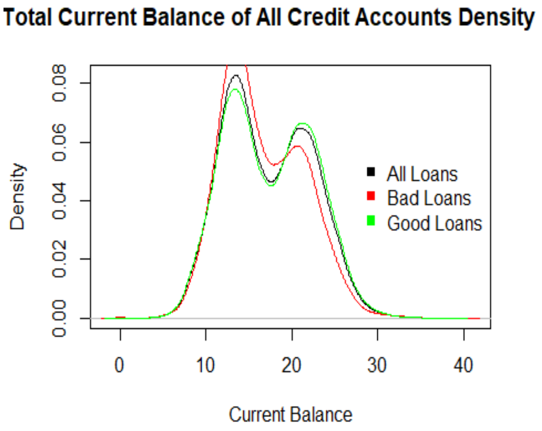


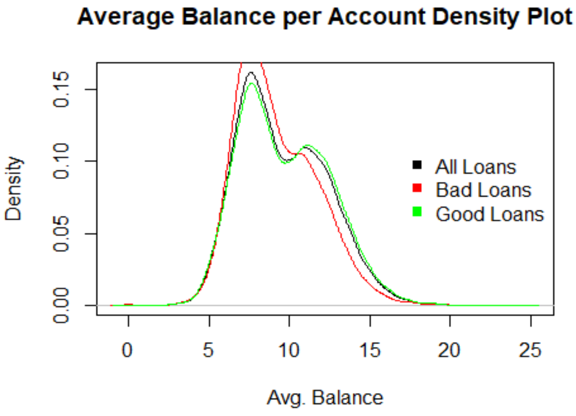
The distribution for the "Bad" loan bar graph didn't seem to change much from the population bar graph, so I'd consider it a weak predictor.

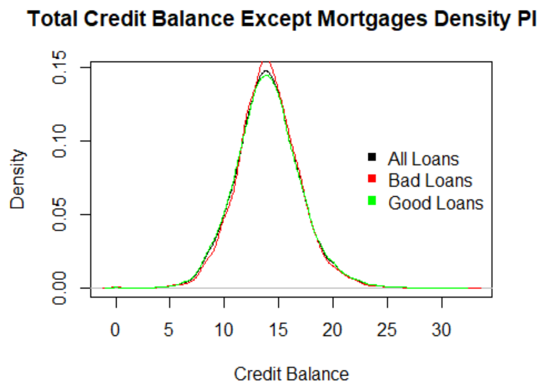
For the final variables, “term” and “verified”, I simply got the counts for population total, “Bad” loans, and “Good” loans of each category within each variable. Then, I found the percentages of the categories in the total population and compared them to the percentages in the “Bad” loans. For example, here is the table for the “term” variable:

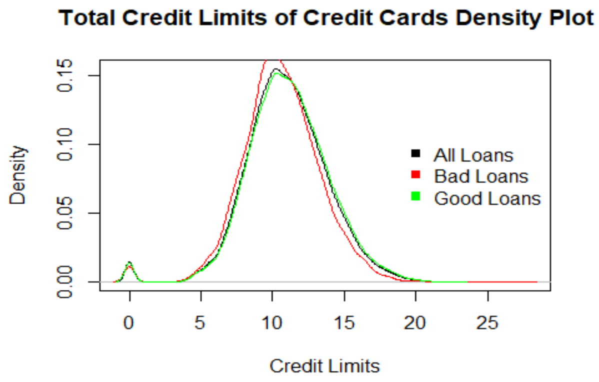
```
## [1] "Total Loans:"

##
##  36 months  60 months
##      25759       8881

## [1] "Bad Loans"

##
##  36 months  60 months
##       4387       3191

## [1] "Good Loans:"

##
##  36 months  60 months
##      21372       5690
```

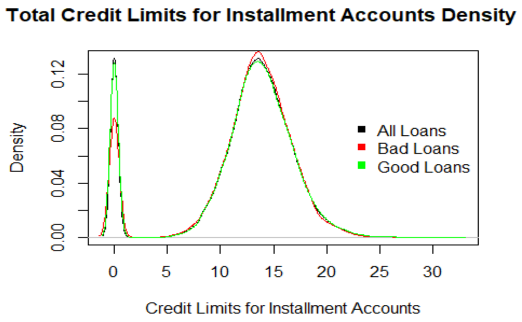
60-month loans made up about 26% of all loans. However, 60-month loans made up 42% of all bad loans and 21% of all good loans. Since the percentage of 60-month loans that made up all bad loans was higher than the percentage of 60-month loans in the total population, I’d say loan term is a strong predictor of a loan default.











Section 5: The Logistic Model

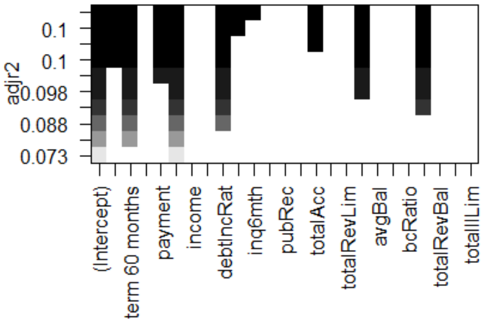
The first step for creating the logistic model was to replace the columns in the loan dataframe with the transformed values.

Then I set a random seed and created a training and testing dataset, and ran the logistic regression analysis.

After running the logistic regression model for the first time, reason was only partially a significant variable- only a few categories of "reason" were significant, less than half. To me, this isn't a very strong predictor. This also aligns with my findings in the exploratory step, where the distribution of "Bad" loans and "Good" loans didn't change much when analyzed under the "reason" variable. Therefore, I decided that "reason" wasn't a very useful predictor variable so I removed it from the training dataframe and reran the logistic regression model below.

```
##
## Call:
## glm(formula = status ~ ., family = "binomial", data =
train_trans_loans_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7767   0.3070   0.5224   0.7214   1.7309
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.518e+00  6.426e-01   3.919 8.90e-05 ***
## amount              2.494e-05  8.668e-06   2.877 0.004014 **
## term 60 months     -1.102e+00  1.358e-01  -8.114 4.90e-16 ***
## rate                9.101e-01  1.081e+00   0.842 0.399871
## payment            -1.119e-02  3.147e-03  -3.555 0.000378 ***
## grade              -2.362e+00  3.635e-01  -6.498 8.16e-11 ***
## income              4.884e-02  5.175e-02   0.944 0.345300
## verifiedVerified -8.283e-02  3.812e-02  -2.173 0.029773 *
## debtIncRat         -2.910e-02  2.759e-03 -10.547  < 2e-16 ***
## delinq2yr          -1.686e-01  3.432e-02  -4.912 9.03e-07 ***
## inq6mth            -1.231e-01  2.945e-02  -4.179 2.93e-05 ***
## openAcc          -2.250e-01  1.195e-01  -1.882 0.059782 .
## pubRec           -1.577e-02  3.860e-02  -0.408 0.682948
## revolRatio       -4.285e-01  1.246e-01  -3.439 0.000585 ***
## totalAcc          3.208e-01  4.713e-02   6.806 1.00e-11 ***
## totalBal         -1.807e-02  2.849e-02  -0.634 0.526029
## totalRevLim       1.159e-01  4.217e-02   2.747 0.006007 **
## accOpen24        -5.368e-01  4.686e-02 -11.456  < 2e-16 ***
## avgBal            2.255e-02  4.658e-02   0.484 0.628298
## bcOpen            1.968e-02  1.136e-02   1.732 0.083242 .
## bcRatio           1.762e-02  6.016e-03   2.929 0.003405 **
## totalLim          1.138e-02  4.263e-03   2.669 0.007618 **
## totalRevBal      -1.166e-02  1.212e-02  -0.962 0.335919
## totalBcLim       -3.330e-03  1.215e-02  -0.274 0.783987
## totalIlLim        1.358e-02  5.934e-03   2.289 0.022087 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29176  on 27711  degrees of freedom
## Residual deviance: 26124  on 27687  degrees of freedom
## AIC: 26174
##
## Number of Fisher Scoring iterations: 5
```

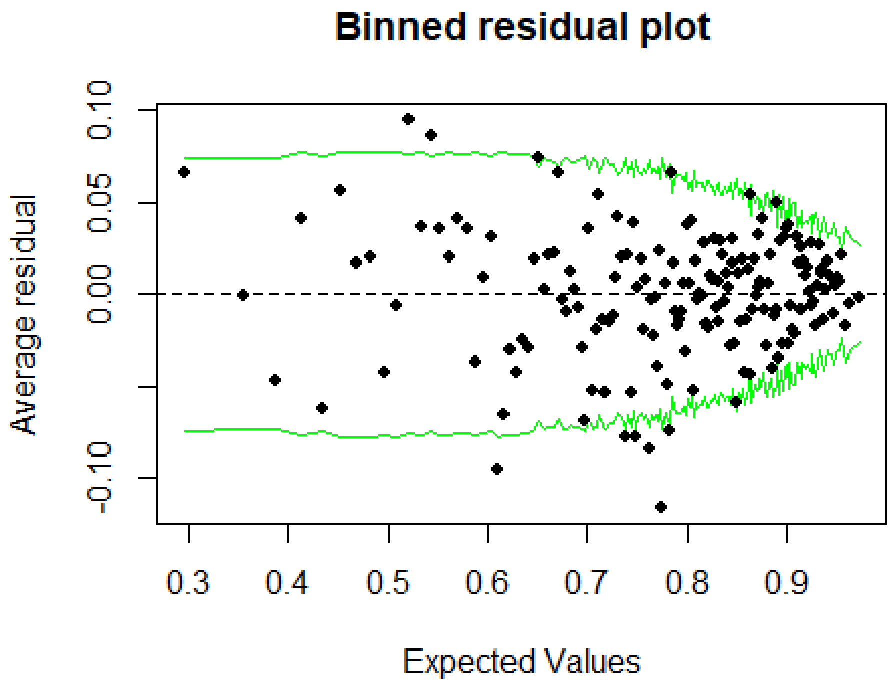
Even though sixteen variables were deemed to be significant, I decided to only keep the ten best, plus interaction terms. This is because I didn't want to overcomplicate the model, especially since I'd be presenting it to management. Ten felt reasonable. Below is the regsubests graph with the 10 predictors it chose, with the ten best variables being amount, term, payment, grade, debtIncRat, delinq2yr, inq6mth, totalAcc, accOpen24, and totalLim.



Next I used forward selection with the model variables to see if any interaction terms made it better. After testing the accuracy of different models with different interaction terms added and removed, I found the most accurate model was when only the term:totalLim interaction term was added. This model ended up having the predictors amount, term, payment, grade, debtIncRat, delinq2yr, inq6mth, totalAcc, accOpen24, totalLim, and term:totalLim. The AIC for this model was 26191.11. It was not the lowest AIC, but again, it did correlate with the highest accuracy of predicted “Good”/“Bad” loans which is why I chose it. Due to the length of the output, I have decided to hide it and just include the code block.

Below is the final model, and the diagnostic plots for that model. The typical residual vs. fitted values plot and QQ plots isn’t useful for binomial logistic regression models (which ours is), so instead I’ll be using a binned residuals plot. Binned residual plots are achieved by “dividing the data into categories (bins) based on their fitted values, and then plotting the average residual versus the average fitted value for each bin.” (Gelman, Hill 2007: 97). If the model were true, one would expect about 95% of the residuals to fall inside the error bounds (here, the green lines). As we can see more than 95% of the residuals fall inside the error bounds, so our model is valid.

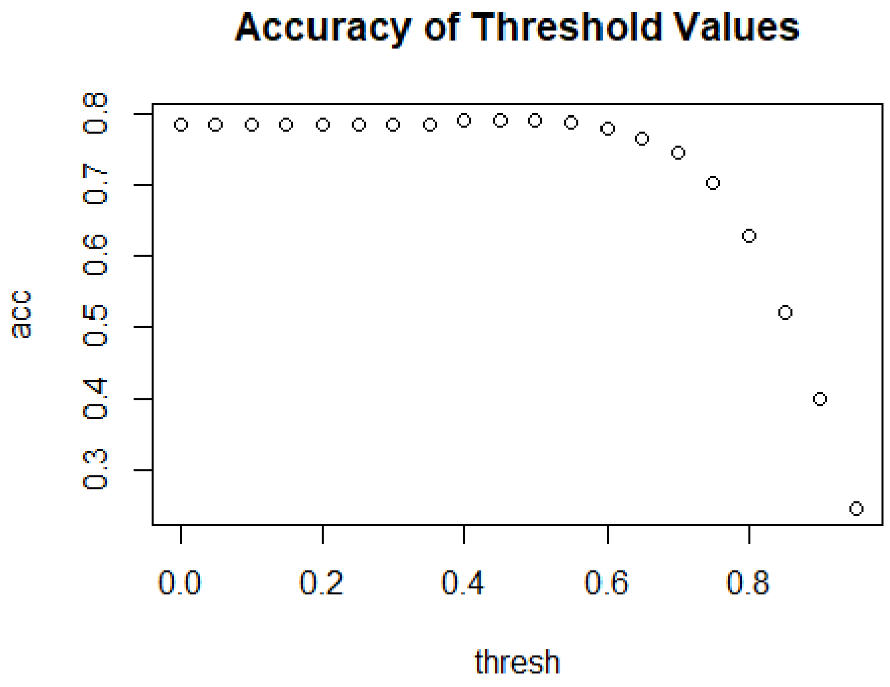
```
## 
## Call:
## glm(formula = status ~ amount + term + payment + grade + debtIncRat + 
##     delinq2yr + inq6mth + totalAcc + accOpen24 + totalLim + term:totalLim, 
##     family = "binomial", data = train_trans_loans_df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.7674   0.3198   0.5267   0.7166   1.7679  
## 
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)    
## (Intercept)             4.784e+00  2.565e-01  18.653  < 2e-16 ***
## amount                  1.803e-05  7.972e-06   2.262  0.02371 *  
## term 60 months         -1.957e+00  2.176e-01  -8.990  < 2e-16 ***
## payment                -8.193e-03  2.910e-03  -2.816  0.00486 ** 
## grade                  -2.548e+00  1.840e-01 -13.851  < 2e-16 ***
## debtIncRat             -2.983e-02  1.931e-03 -15.452  < 2e-16 ***
## delinq2yr              -1.688e-01  3.349e-02  -5.040 4.66e-07 ***
## inq6mth                -1.145e-01  2.913e-02  -3.932 8.44e-05 ***
## totalAcc                2.557e-01  3.909e-02   6.541 6.11e-11 ***
## accOpen24              -5.423e-01  4.291e-02 -12.638  < 2e-16 ***
## totalLim                8.866e-03  9.533e-04   9.301  < 2e-16 ***
## term 60 months:totalLim  7.406e-03  1.537e-03   4.818 1.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 29176  on 27711  degrees of freedom
## Residual deviance: 26167  on 27700  degrees of freedom
## AIC: 26191
## 
## Number of Fisher Scoring iterations: 4
```



Section 6: Oprimizing the Threshold for Accuracy

Below are the prediction accuracies for the different threshold values, starting at 0.05 and incrementing by 0.05 up to 0.95:

```
## [1] "Proportion correctly predicted at 0 threshold:  0.784642032332563"
## [1] "Proportion correctly predicted at 0.05 threshold:  0.784642032332563"
## [1] "Proportion correctly predicted at 0.1 threshold:  0.784642032332563"
## [1] "Proportion correctly predicted at 0.15 threshold:  0.784642032332563"
## [1] "Proportion correctly predicted at 0.2 threshold:  0.784642032332563"
## [1] "Proportion correctly predicted at 0.25 threshold:  0.784786374133949"
## [1] "Proportion correctly predicted at 0.3 threshold:  0.784930715935335"
## [1] "Proportion correctly predicted at 0.35 threshold:  0.786662817551963"
## [1] "Proportion correctly predicted at 0.4 threshold:  0.790271362586605"
## [1] "Proportion correctly predicted at 0.45 threshold:  0.790271362586605"
## [1] "Proportion correctly predicted at 0.5 threshold:  0.791714780600462"
## [1] "Proportion correctly predicted at 0.55 threshold:  0.787673210161663"
## [1] "Proportion correctly predicted at 0.6 threshold:  0.780311778290993"
## [1] "Proportion correctly predicted at 0.65 threshold:  0.766021939953811"
## [1] "Proportion correctly predicted at 0.7 threshold:  0.746102771362587"
## [1] "Proportion correctly predicted at 0.75 threshold:  0.703088914549654"
## [1] "Proportion correctly predicted at 0.8 threshold:  0.630196304849885"
## [1] "Proportion correctly predicted at 0.85 threshold:  0.521939953810624"
## [1] "Proportion correctly predicted at 0.9 threshold:  0.39939376443418"
## [1] "Proportion correctly predicted at 0.95 threshold:  0.245814087759815"
```



The model accuracy seems to be roughly equal until a threshold value of 0.6, after which it noticeably drops off. The most accurate threshold value is 0.5, with a 79.17% accuracy rate. This is about 0.71% better than if every loan was treated as "Good." The contingency table for the 0.5 threshold value is shown below.

```
##      predLoan_best
##        Bad Good  Sum
##   0    220 1272 1492
##   1    171 5265 5436
##   Sum  391 6537 6928

## [1] "Proportion correctly predicted =  0.791714780600462"
```

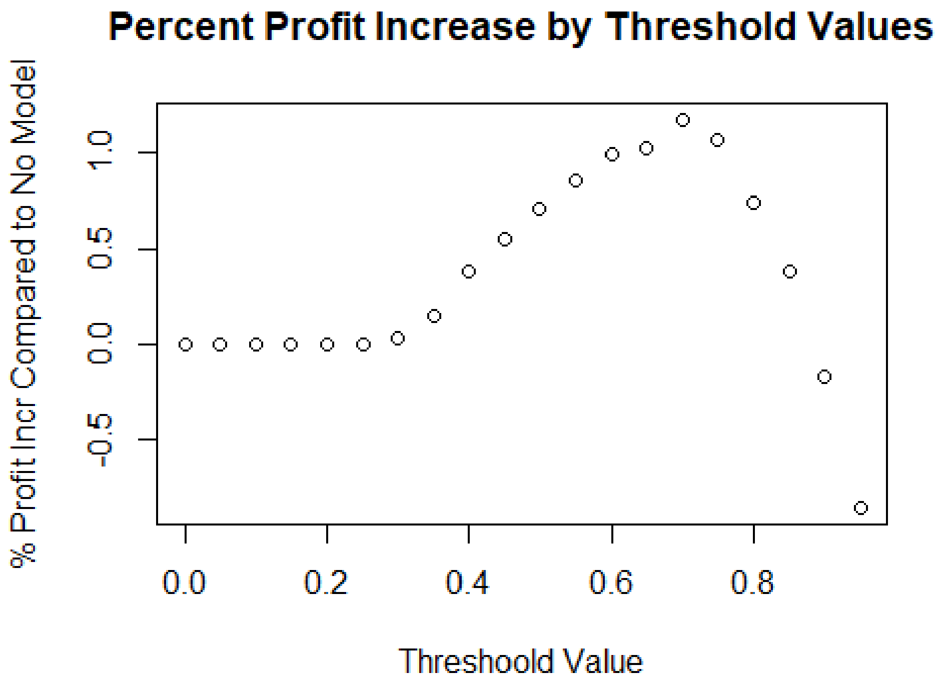
Section 7: Optimizing the Threshold for Profit

Next was determining the profit of the test dataframe when the model wasn't used; i.e. every loan was treated as "Good":

```
## [1] "Profit without using the model: $1728486.91"
```

Finally, I calculated the profit when the model was used for different thresholds. With the model, ideally loans would only be granted if they were accurately predicted to be "Good." Here is a graph of the results:

```
## [1] "Profit using the model at 0 threshold: $1728486.91"
## [1] "Profit using the model at 0.05 threshold: $1728486.91"
## [1] "Profit using the model at 0.1 threshold: $1728486.91"
## [1] "Profit using the model at 0.15 threshold: $1728486.91"
## [1] "Profit using the model at 0.2 threshold: $1728486.91"
## [1] "Profit using the model at 0.25 threshold: $1732587.3"
## [1] "Profit using the model at 0.3 threshold: $1776358.55"
## [1] "Profit using the model at 0.35 threshold: $1987949.31"
## [1] "Profit using the model at 0.4 threshold: $2381009.4"
## [1] "Profit using the model at 0.45 threshold: $2668597.63"
## [1] "Profit using the model at 0.5 threshold: $2952928.09"
## [1] "Profit using the model at 0.55 threshold: $3196382.76"
## [1] "Profit using the model at 0.6 threshold: $3446686.38"
## [1] "Profit using the model at 0.65 threshold: $3497896.62"
## [1] "Profit using the model at 0.7 threshold: $3753841.74"
## [1] "Profit using the model at 0.75 threshold: $3575791.03"
## [1] "Profit using the model at 0.8 threshold: $2999602.06"
## [1] "Profit using the model at 0.85 threshold: $2380621.51"
## [1] "Profit using the model at 0.9 threshold: $1440221.71"
## [1] "Profit using the model at 0.95 threshold: $249016.01"

## [1] "The maximum profit of $3753841.74 occurs at a threshold value of 0.7"
```



Section 8: Results Summary

As stated, my model was limited to ten variables plus interaction terms, the most accurate being one interaction term. The reasons for this were stated in Section 5. A number of variables were removed from consideration for a number of reasons, which I also stated in previous sections 3, 4, and 5. The chosen variables ended up being amount, term, payment, grade, debtIncRat, delinq2yr, inq6mth, totalAcc, accOpen24, and totalLim, with the interaction variable being term:totalLim, as this model gave the greatest accuracy despite not having the lowest AIC value.

Interestingly, the greatest profit came when the threshold was at 0.7, with a profit of $3,753,841.74. This is a 117.18% increase from the profit found when the model was not used. However, the most accurate model was when the threshold was 0.5, with an accuracy of 79.17% . It is also worth noting that when the threshold was 0.5 the profit was $2,952,928.09, a 70.83% increase. Also, when the threshold was 0.7, the accuracy was 74.61%. Although management would mostly likely want to maximize profit, the option for them to go with accuracy is there.

If I were to redo this, I wouldn't limit my model to just 10 variables; I'd use all that were deemed significant. Additionally, I'd build the model to maximize profit instead of accuracy. Currently, the profit is maximized based on threshold value and not included variables in the model.