Predicting Loan Defaults with Logistic Regression

Julie Bazalewski

10/17/2020

1. Executive Summary

This report provides an analysis and examination of factors that can predict loan defaults with improved accuracy. The proposed model utilizes logistic regression to classify whether or not a loan will result in a default. All of the calculations and details related to developing this model can be found in the subsequent sections.

This model is predicted to increase profits by approximately 300% as opposed to not using such a model to reject high-risk loans. For the test calculations, a profit of $945,907 was predicted without using a model to reject loans compared to a total profit of $3,138,480 after employing the proposed model. The overall model accuracy is approximately 73%. “Good” loans are predicted with 81% accuracy and “bad” loans are predicted with 46% accuracy.

Limitations of this model result in some loans being rejected that would not result in a default and some loans issued that will default. However, the model takes into account the optimal balance of these errors to maximize profits. It is recommended that this model be employed in the loan issuing process in order to increase profits.

2. Introduction

Many variables factor into whether an applicant is likely to default on their loan. The loan outcome is binary, either repaid or defaulted. Therefore, using logistic regression, it may be possible to determine the outcome of a loan. Variables including loan term and rate, information relating to the individual’s credit usage, as well as income and employment history data among other factors will be considered. Statistical analysis will be performed to determine which of these variables is useful in determining the loan outcome.

3. Preparing and Cleaning the Data

The dataset, loans50k.csv, contains data about 50,000 loans with 30 variables. The variables range from categorical to numeric, both discrete and continuous. In order to simply the dataset, several variables were removed which are not considered to add much value. The “loanID” variable is only an identifier and provides no information regarding the loan. The “employment” variable, which contains various job titles and has no obvious use for the analysis, is also removed. Finally, the variable “state” is also removed, as this is not expected to impact the outcome of a loan in a significant way.

```
loans.clean <- loans
loans.clean$loanID <- NULL
loans.clean$employment <- NULL
loans.clean$state <- NULL
```

In addition, feature engineering was performed for the variables “reason” and “length” to reduce the number of levels in each. The variable “reason” was reduced to two levels, either “Credit Related” or “Other”. The reasons “credit_card” and “debt_consolidation” are considered to be related to credit, while all other reasons are considered to be for another reason (“other”).

```
loans.clean <- loans.clean %>% mutate(reason = case_when(
  (reason == "credit_card" | reason == "debt_consolidation") ~ "Credit
Related",
  TRUE ~ "Other"
))
```

In addition, the variable “length” was reduced to four levels of employment history: 1 year or less, 2 to 5 years, 6 to 9 years, and 10 or more years.

```
loans.clean <- loans.clean %>% mutate(length = case_when(
  (length == "< 1 year" | length == "1 year") ~ "1 year or less",
  (length == "2 years" | length == "3 years" | length == "4 years" | length
== "5 years") ~ "2 to 5 years",
  (length == "6 years" | length == "7 years" | length == "8 years" | length
== "9 years") ~ "6 to 9 years",
  (length == "10 years" | length == "10+ years") ~ "10 or more years"
))
```

Some of the loans have a value of “n/a” for length of employment. These were changed to the R missing value, NA.

```
loans.clean$length[loans.clean$length =='n/a'] <- NA
```

Next, the variable “status” was modified to a binary outcome of either “good” or “bad” loans. Good loans are all those that are fully paid. Bad loans are loans that have a status of charged off or default.

```
loans.clean <- loans.clean %>% mutate(status = case_when(
  (status == "Fully Paid") ~ "Good",
  (status == "Charged Off" | status == "Default") ~ "Bad",
  TRUE ~ "Unknown"
))
```

Loans that are late, current (being paid), or in grace period are removed from the data.

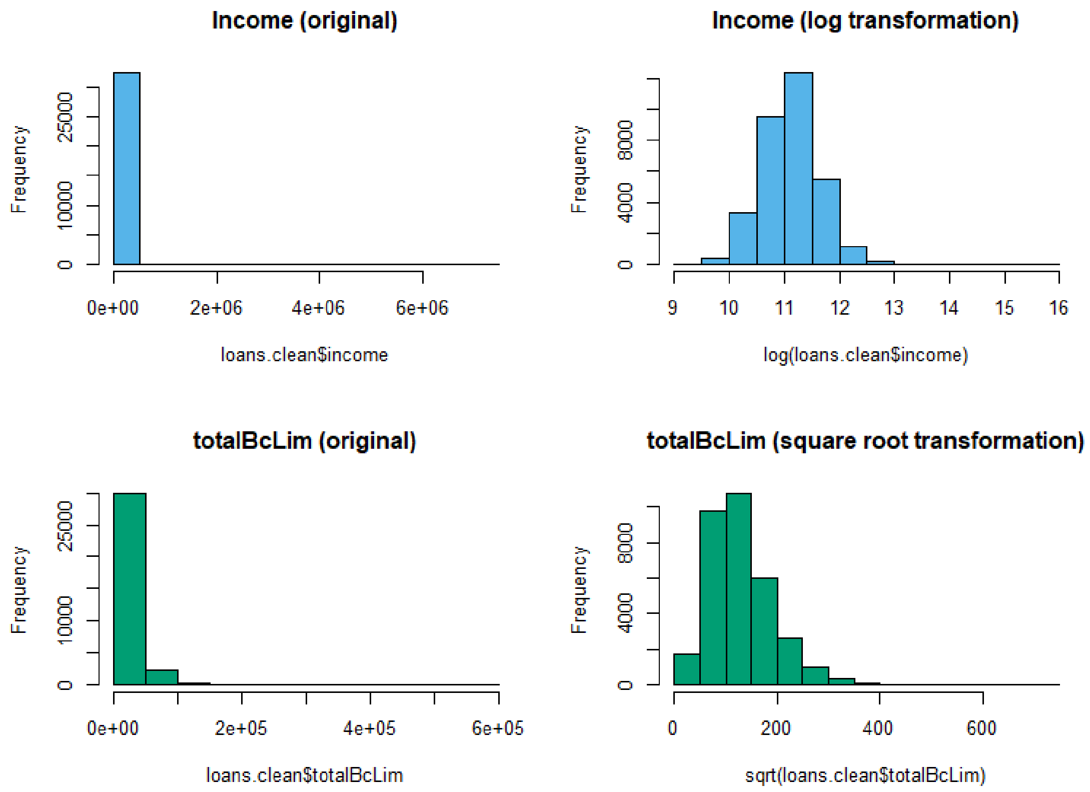
```
loans.clean <- loans.clean[!loans.clean$status == "Unknown", ]
loans.clean$status <- as.factor(loans.clean$status)
```

Several of the variables (length, bcRatio, and bcOpen) contain missing values. The “summary” function is used to examine the variables for which the missing values occur. Variables “revolRatio”, “bcRatio”, “bcOpen”, and “length” have missing values. The missing values are a small portion of the data (about 5%). Therefore, it is reasonable to remove these data points from the dataset. The prepared data set now includes 32,475 data points with 27 variables.

4. Exploring and Transforming the Data

Some of the variables in the loans dataset are strongly skewed. Histograms of each quantitative variable were examined to determine which have the most severe skew. Income and totalBcLim appear to be the most skewed continuous variables.

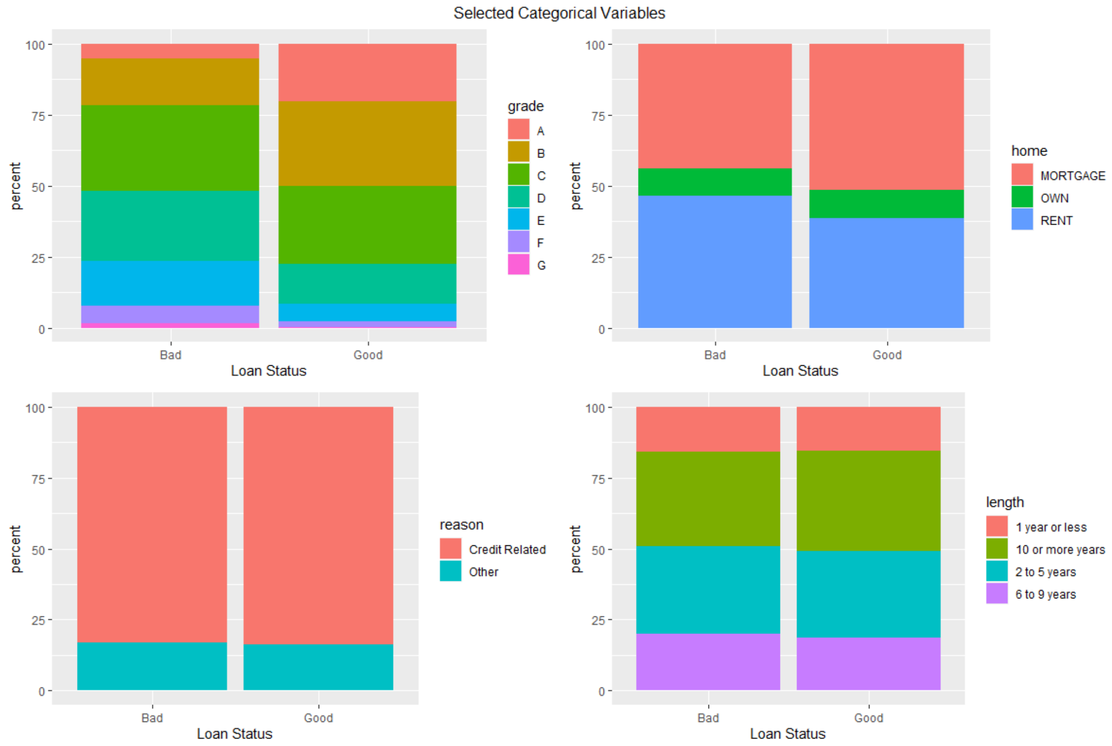
A log transformation was applied to "income" to reduce the number of extreme values and to make it more normally distributed. The variable "totalBCLim" cannot have a log transformation applied because there are zero values in the data. Therefore, a square root transformation, while less strong of a transformation, is applied to make the data more normal. The distribution of variables "income" and "totalBcLim" before and after transformation is shown below.



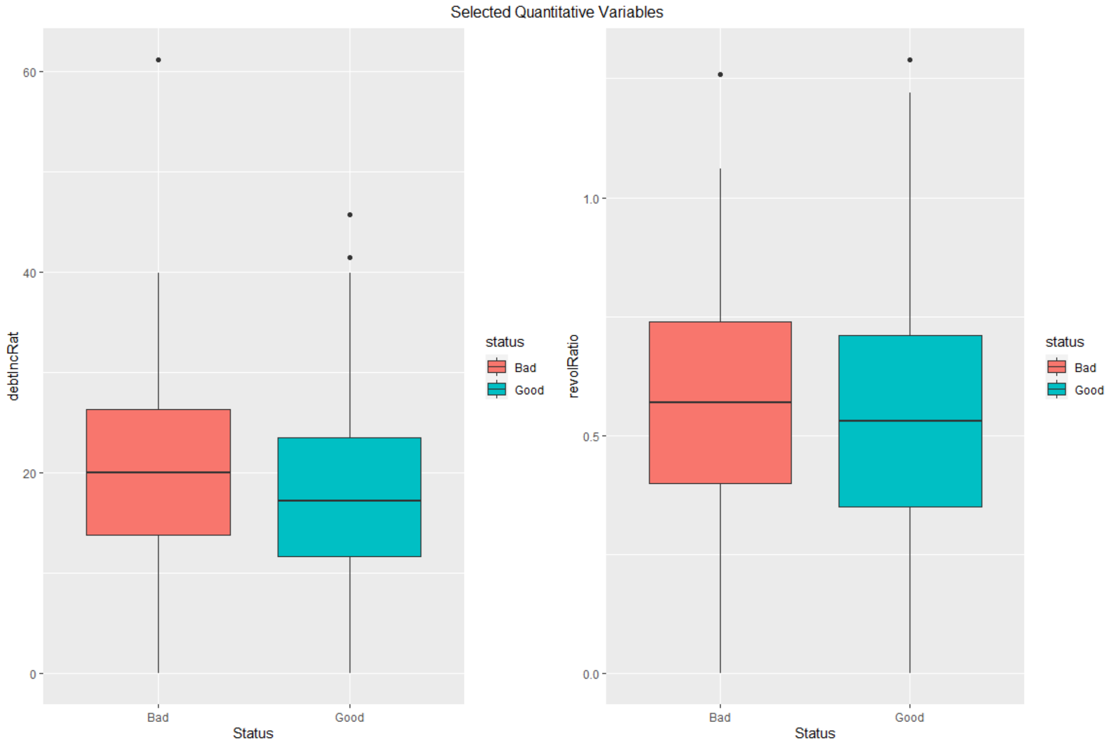
The transformations are applied as follows:

```
loans.clean$income <- log(loans.clean$income)
loans.clean$totalBcLim <- sqrt(loans.clean$totalBcLim)
```

Both the categorical and quantitave variables were explored to examine the relationship between the predictors and the loan outcome. Bar charts using percentages instead of counts are utilized to explore some of the categorical variables. Percentages are easier to compare because the “good” and “bad” loan sample sizes are different. Variables “grade”, “home”, “reason”, and “length were chosen as likely candidates to have an effect on loan outcome. As expected, the grade of the risk of the loan differs between”good" and “bad” loans. Also, the home ownership variable varies slightly for loan type. About 51% of “good” loans are from those with a mortgage, compared to about 44% of “bad” loans. The loan reason and the length of employment do not vary much based on the loan type.



Quantitative variables “payment”, “income”, “debtIncRat”, “totalBcLim”, “revolRatio”, and “amount” were chosen as likely candidates to have an effect on loan outcome. Boxplots were created for each of these variables. The applicant’s income, payment, and loan amount did not vary much with loan outcome type. Plots for ratio monthly non-mortgage debt payment to monthly income (debtIncRat) and the proportion of revoling credit in use (revolRatio) are included below. These two variables exhibited the most significant difference between “good” and “bad” loans, where a higher value for both variables corresponds to more “bad” loans.



5. The Logistic Model

Next, the logistic regression model is created. The first step is to split the data into a training set that contains 80% of the data, and a test or "validation" set that contains the other 20% of the data. Additionally, the "totalPaid" variable is removed from the training set because it cannot be used in creating the regression model because the data is not known at the time of loan issuance. The training data has 25980 observations and the test data has 6495.

```
train.index <- sample(seq_len(nrow(loans.clean)), size = floor(0.8 *
nrow(loans.clean)))
train.loans <- loans.clean[train.index, ]
test.loans <- loans.clean[-train.index, ]
```

The first attempt at a model uses all of the predictor variables. The AIC is 24085. Forward and Backward stepwise regression models using the "step" function are also evaluated. The foward stepwise model results in an AIC of 24070, with 19 predictor variables. The backward model has an AIC of 24068 with 18 variables, making it slightly better in terms of AIC and model simplicity.

```
fit.full <- glm(status~., data=train.loans, family="binomial")
```

The "predict" function is used to predict the loan status for loans in the test data set with a threshold of 0.5 for both the full model "preds" and the backwards stepwise model "preds.backward". The binary variable "status" is assigned 0 or 1 for "bad" and "good" loans respectively. R assigns this alphabetically. This is also confirmed with the output because there are more "good" loans in the dataset.

Both models predict correctly 78.5% of the time. Therefore, the backwards stepwise regression model will be used because it is a simpler model with the same level of accuracy.

Prediction results for full model:

```
##       preds
##         Bad Good  Sum
##   Bad   156 1275 1431
##   Good  123 4941 5064
##   Sum   279 6216 6495

## [1] "Proportion correctly predicted =  0.784757505773672"
```

Prediction results for backward stepwise model:

```
##       preds.backward
##         Bad Good  Sum
##   Bad   159 1272 1431
##   Good  127 4937 5064
##   Sum   286 6209 6495

## [1] "Proportion correctly predicted =  0.784603541185527"
```

A collinearity check is then performed on the model. The largest VIF is for the “amount” variable, with a value of 65.9. This is significantly larger than the desired limit of 10. Therefore, this variable is removed from the model. In addition, “amount” is not statistically significant at a level of $\alpha = 0.01$ like the other remaining predictor variables. After “amount” is removed, the largest VIF is totalRevBal with a value of 12.5. While this is still greater than 10, collinearity does not actually affect model predictions and the remaining variables will be retained.

The remaining variables and corresponding VIF values are included below:

```
fit.backward2 <- glm(status ~ term + payment + grade + home + debtIncRat +
    delinq2yr + inq6mth + openAcc + revolRatio + totalAcc + totalRevLim +
    accOpen24 + avgBal + bcOpen + totalRevBal + totalBcLim +
    totalIlLim, data=train.loans, family="binomial")
vifs <- vif(fit.backward2)
vifs

## term60 months       payment        gradeB        gradeC        gradeD
##      1.424713      1.431132      2.129479      2.551916      2.408034
##        gradeE        gradeF        gradeG       homeOWN      homeRENT
##      2.054070      1.455145      1.130916      1.144717      1.505557
##    debtIncRat     delinq2yr       inq6mth       openAcc    revolRatio
##      1.384627      1.075812      1.143599      2.676252      1.754332
##      totalAcc   totalRevLim     accOpen24        avgBal        bcOpen
##      2.283153      4.915853      1.636995      1.677950      4.435903
##   totalRevBal    totalBcLim    totalIlLim
##     12.458835      4.641565     10.830490
```

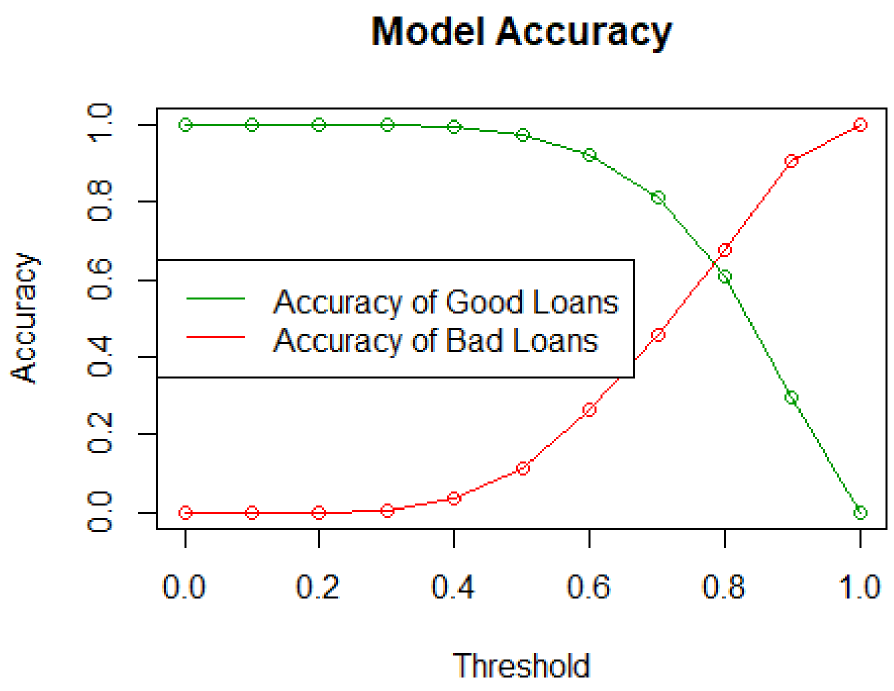
With “amount” removed, the model accuracy is now 78.4%, which is not significantly different from the original model. While this initially appears to be a high level of accuracy, many of the bad loans were predicted as “good”. While 97% of the “good” loans were classified correctly, only 160 of the 1431 “bad” loans (11%) were identified. This is not an effective model for predicting if a loan will be repaid because it poorly identifies the “bad” loans at a threshold of 0.5.

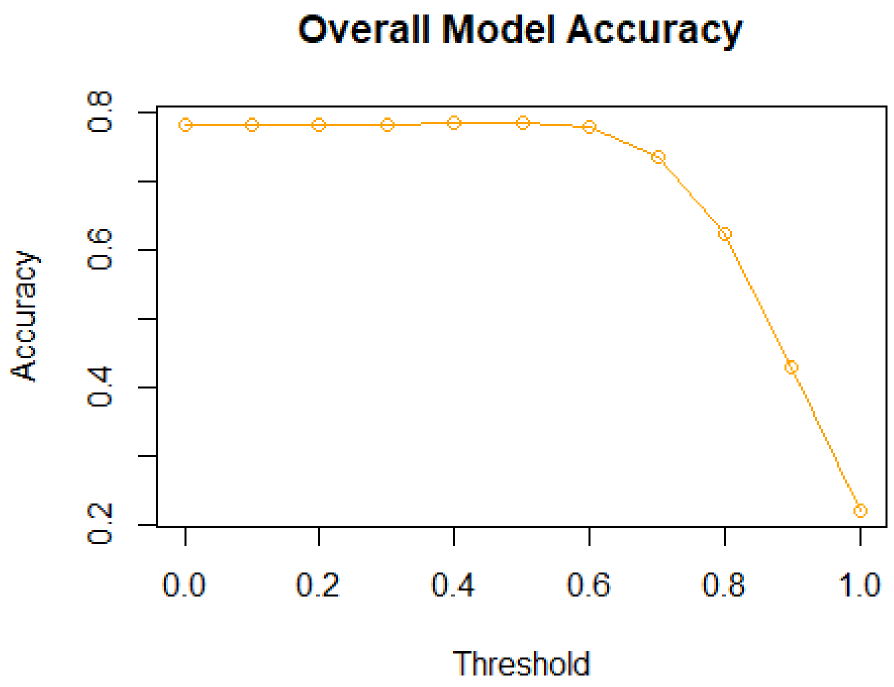
```
##       preds.backward
##         Bad Good  Sum
##   Bad   160 1271 1431
##   Good  131 4933 5064
##   Sum   291 6204 6495

## [1] "Proportion correctly predicted =  0.784141647421093"
```

6. Optimizing the Threshold for Accuracy

In order to improve the model so that loans that will likely not be repaid can be identified, the threshold must be modified to allow for better detection of “bad” loans. In order to identify more “bad” loans, the accuracy of the prediction of “good” loans will be decreased. Practially, this means more loans that may be “good” would not be issued, but it will also allow the bank to identify the “bad” loans with more accuracy.

Threshold values between 0 and 1 at 0.1 increments were plotted against “good”, “bad”, and overall model accuracy by looping over each threshold value and calculating the accuracies.

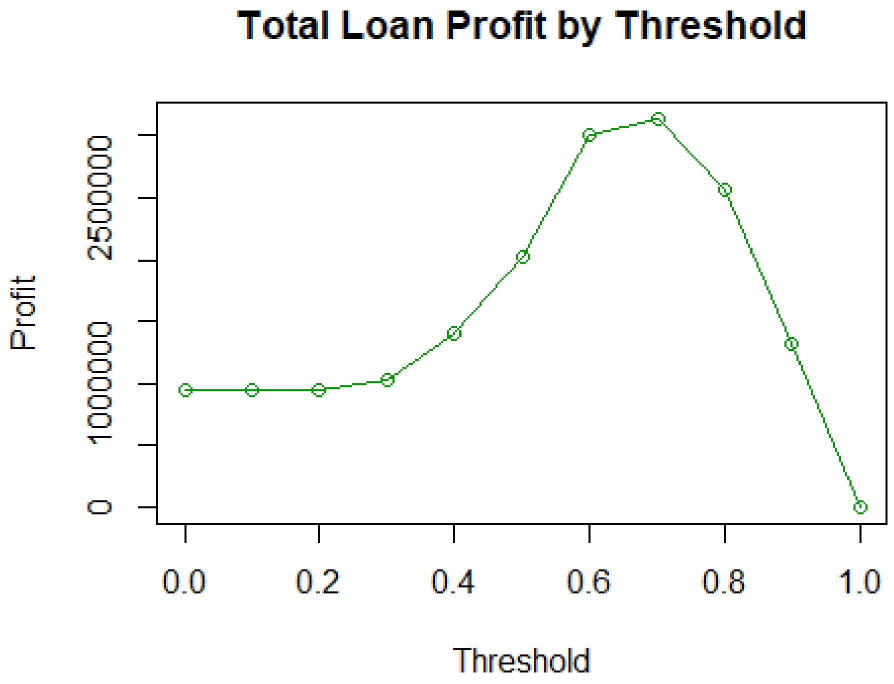




The overall greatest model accuracy (78.4%) is achieved with a threshold of 0.5. However, the accuracy is near 78% for all thresholds between 0 and 0.6. At a threshold of 0.6, significantly more "bad" loans can be identified compared to a threshold of 0.5. The "bad" loan detection increases from 11% to 26% with an increase in threshold of 0.5 to 0.6. The crossover point for “good” and “bad” loan accuracy is at a threshold of 0.8. This may be a reasonable threshold to use if identifying potentially bad loans is more imporant than incorrectly classifying good loans as “bad”. The overall model accuracy at a threshold of 0.8 is about 68%.

7. Optimizing the Threshold for Profit

As discussed in the previous section, there is a tradeoff between correctly predicting “good” and “bad” loans. From the bank’s perspective, achieving maximum profit is the most important model feature. Profit from “good” loans is determined by subtracting the variable “amount” from “totalPaid”. This value is added to the test data set as the variable “profit”. For each threshold value from 0 to 1, the total profit from “good” loans in the test set is calculated.



The maximum profit occurs at a threshold of 0.7 with a total profit of $3,138,480 from the test set. This is significantly higher than the profit if no “bad” loans were rejected by not using the model. Without the model, the profit is $945,907, only about 30% of the profit with the model. If all “bad” loans were rejected, however, the profit would be $0 because in order to detect all of the “bad” loans, all of the “good” loans are also rejected, resulting in no profit.

At a threshold of 0.7 (the point of maximum profit), the overall model accuracy is about 73%. This is slightly below the maximum overall accuracy of 78% at a threshold of 0.4. For this threshold, “good” loans are predicted with 81% accuracy and “bad” loans at 46%. Profits are maximized when there is a reasonable balance between detecting good and bad loans. A threshold of 0.7 achieves this balance between accepting enough “bad” loans as “good” to prevent rejecting profitable “good” loans. Overall model accuracy does not necessarily correspond to maximized profit.

8. Results Summary

The final logistic regression model to predict the binary outcome for “good” and “bad” loans includes first order predictor variables: term, payment, grade, home, debtIncRat, delinq2yr, inq6mth, openAcc, revolRatio, totalAcc, totalRevLim, accOpen24, avgBal, bcOpen, totalRevBal, totalBcLim, and totalIlLim. Each of the variables is significant at an α = 0.05 significance level. At a threshold of 0.7, the model has an overall accuracy of 73%. “Good” loans are predicted with 81% accuracy and “bad” loans are predicted with 46% accuracy. 654 of the 1431 “bad” loans in the test set are correctly identified by the model, while 4109 of 5064 of the “good” loans are identified. This model is predicted to increase profits by approximately 300% compared to not using a model to reject “bad” loans.