Loan Analysis

Matt Rock

3/8/2020

Section 2: Introduction

In the aftermath of the 2008 recession, the question everyone asked the banking industry was “How did you miss this financial collapse? You had all the information, but ignored it.” The pressure to create more mortgage-backed tranches meant housing lenders were actually competing to lend to individuals with riskier credit, leading to a recession that still continues to have repercussions.

Now, with the spectre of financial ruin still lingering, banks need to protect themselves from bad risks of all kinds. The purpose of this project is to assist banks in being able to protect their investment portfolio with safe investments, while still creating a satisfying application experience for the loan officer and loan seeker..

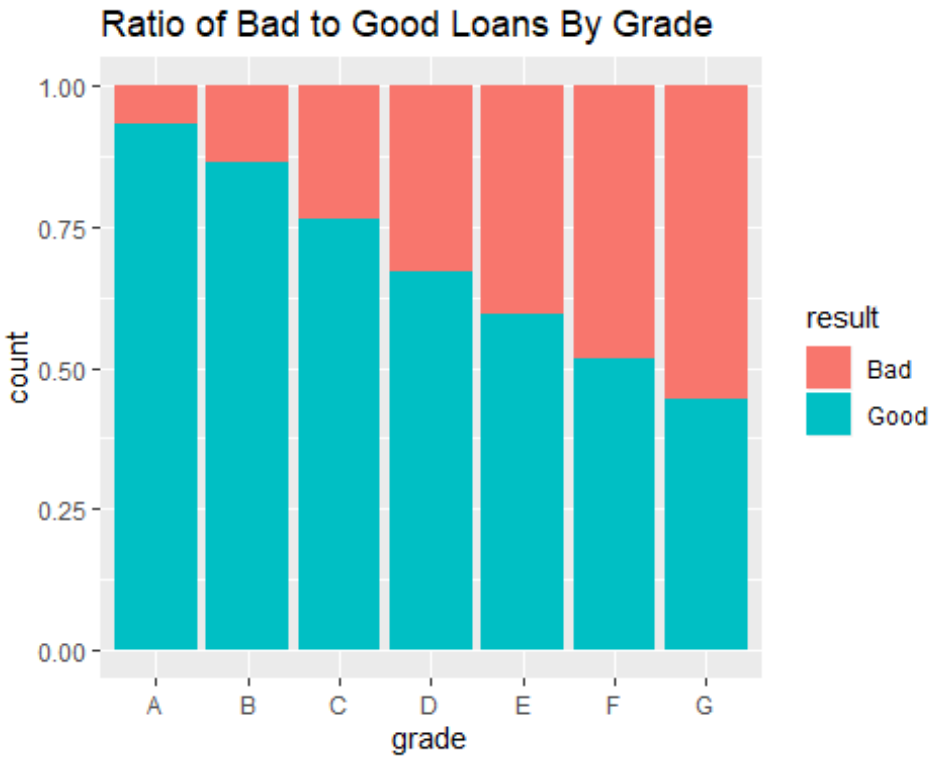
Section 3: Preparing and Cleaning the Data

The initial data set had 50,000 observations and 32 variables. Selecting only on the relevant loan status left 34,655 data points. The initial variable trimming removed loanID (randomized number), employment (covered by job length and salary), and state (with 50 states, random chance would lead to false positives). The interest rate of the loan, and therefore the loan payment, comes from the loan officer, so neither would be useful to help the bank project if a loan will fail.

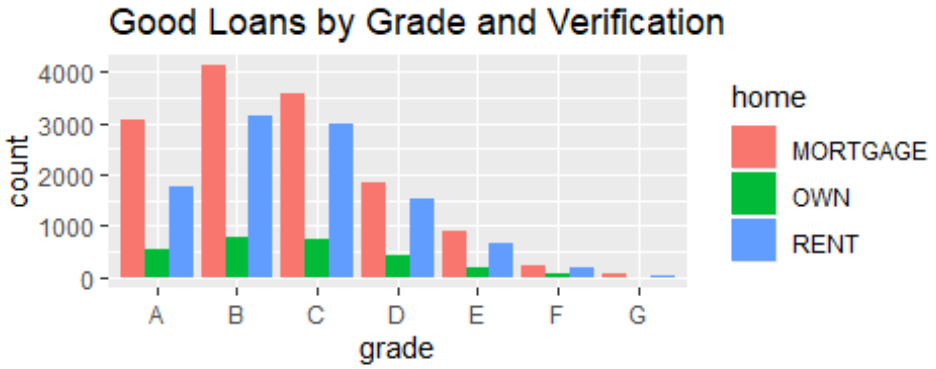
Finally, many of the variables were very similar. Total accounts is removed, as open accounts would work just as well. Likewise, credit card limits and installment limits fell under total limits. Total balance shows financial problems more than average balance. Finally, the ratio of used credit to total credit and total open credit are heavily correlated with the total credit used and total credit limits.

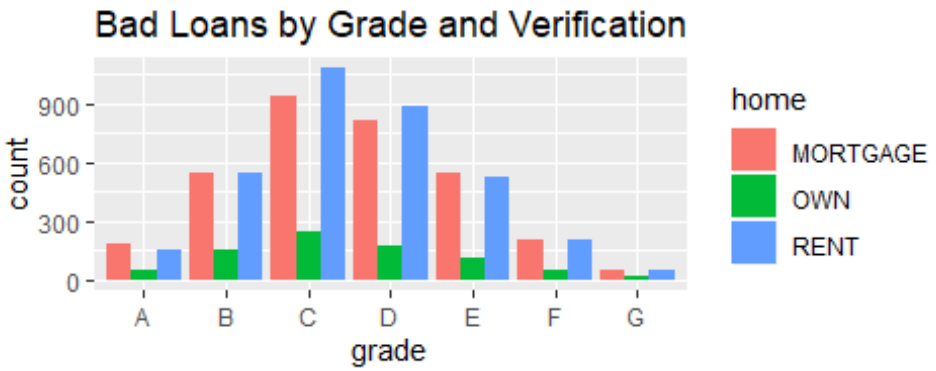
Section 4: Exploring the Data

The grade of the loan is the biggest indicator of its success. However, since the loan grade is the result of the loan process, rather than information around the application, it doesn’t make sense to use it as part of a model predicting if a loan is successful. Where the grade will be useful is benchmarking what variables indicate a good or bad loan.

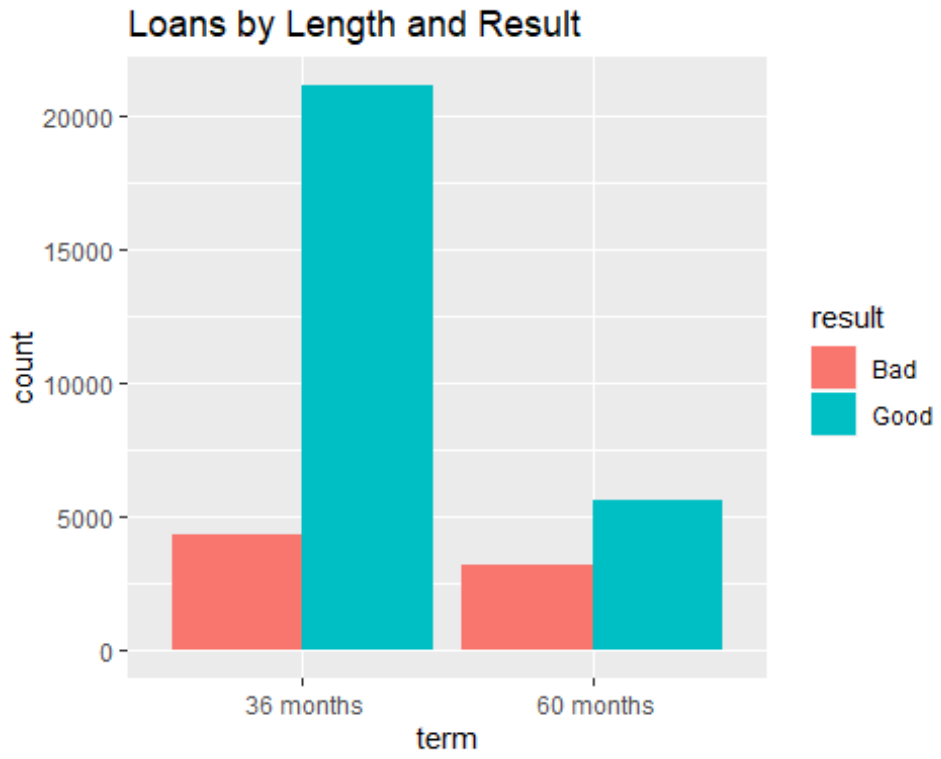


Loans to people with a mortgage were more successful than those to renters. About one out of every five loans given to someone with a mortgage failed, compared to renters failing with one of four loans. This was heavily influenced by the grade of the loan

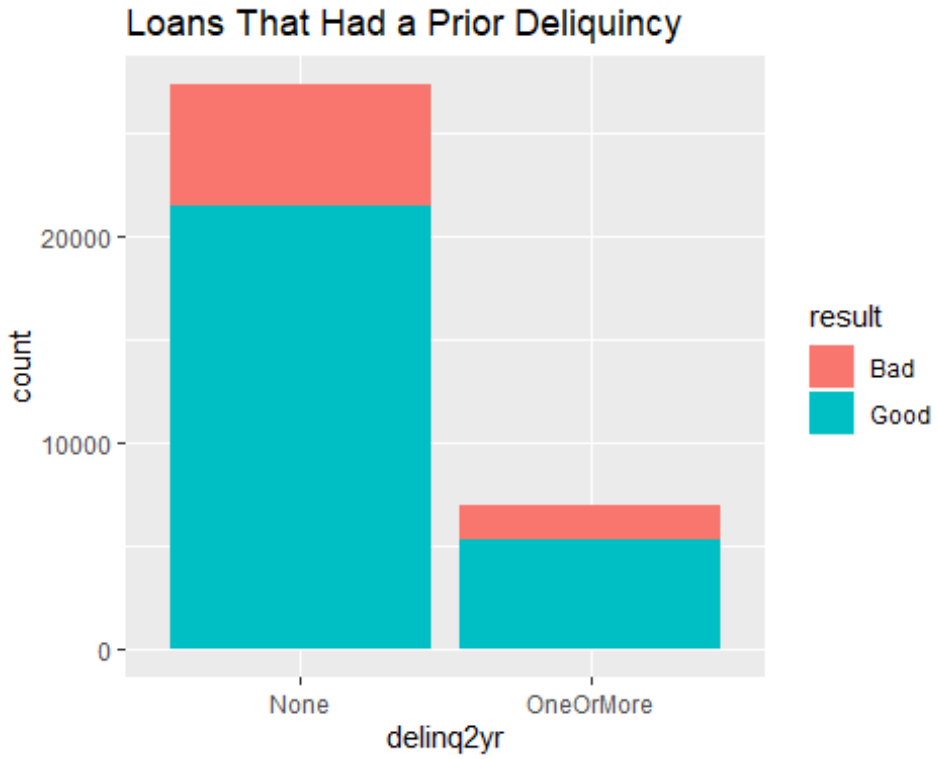




Similarly, the length of the loan was a very strong indicator of success or failure. Around one of six loans with a 3 year term failed, but over a third of 5 year loans failed.

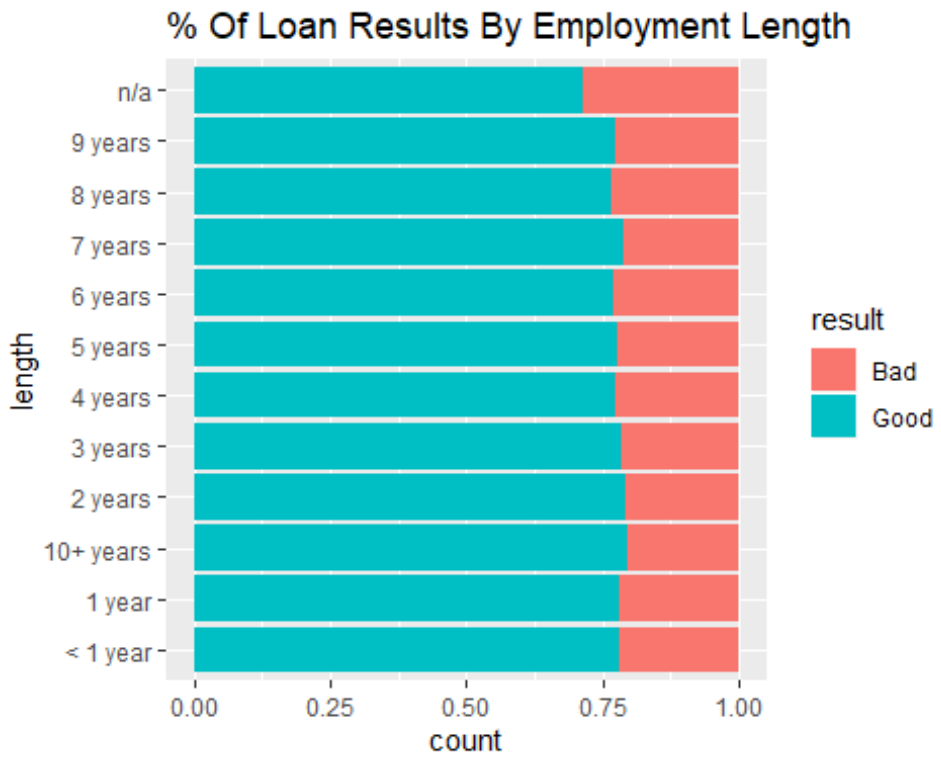


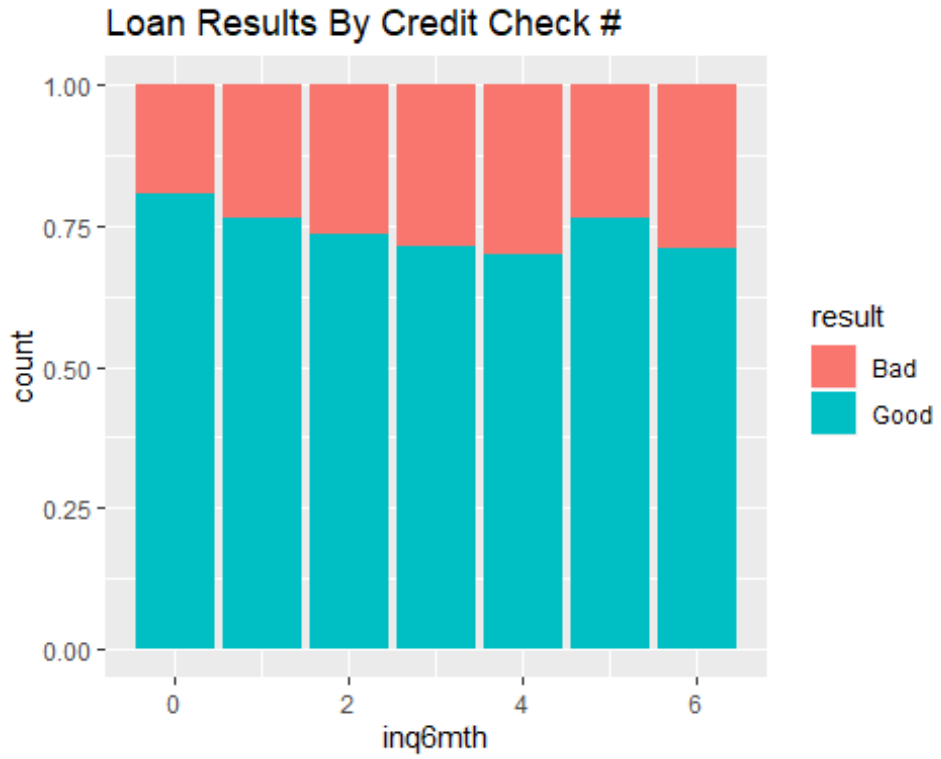
The majority of loans were given to people with no late payments in the last 2 years, but a loan was more likely to default if it was issued to someone without a late payment than

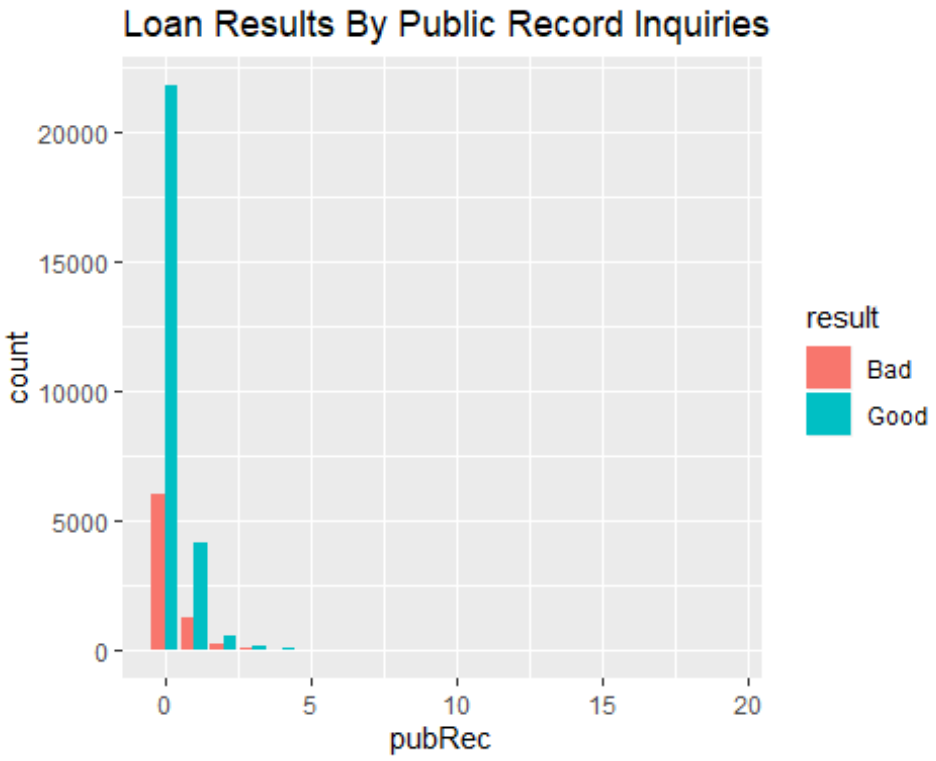


with.

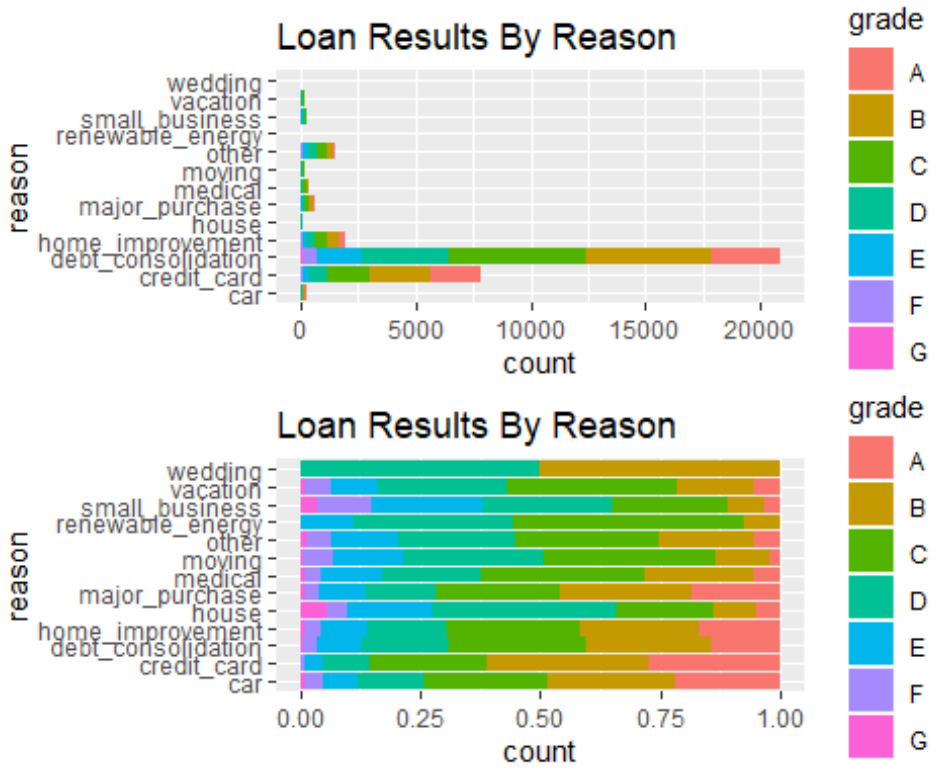
Some variables involved loan approvals more than loan results, and were removed. Length of employment, public record hits, and 6 month inquiries had no real clear bearing on loan quality.


% Of Loan Results By Employment Length
n/a
9 years
8 years
7 years
6 years
5 years
4 years
3 years
2 years
10+ years
1 year
< 1 year
length
0.00
0.25
0.50
0.75
1.00
count
result
Bad
Good


Loan Results By Credit Check #
1.00
0.75
0.50
0.25
0.00
count
0
2
4
6
inq6mth
result
Bad
Good

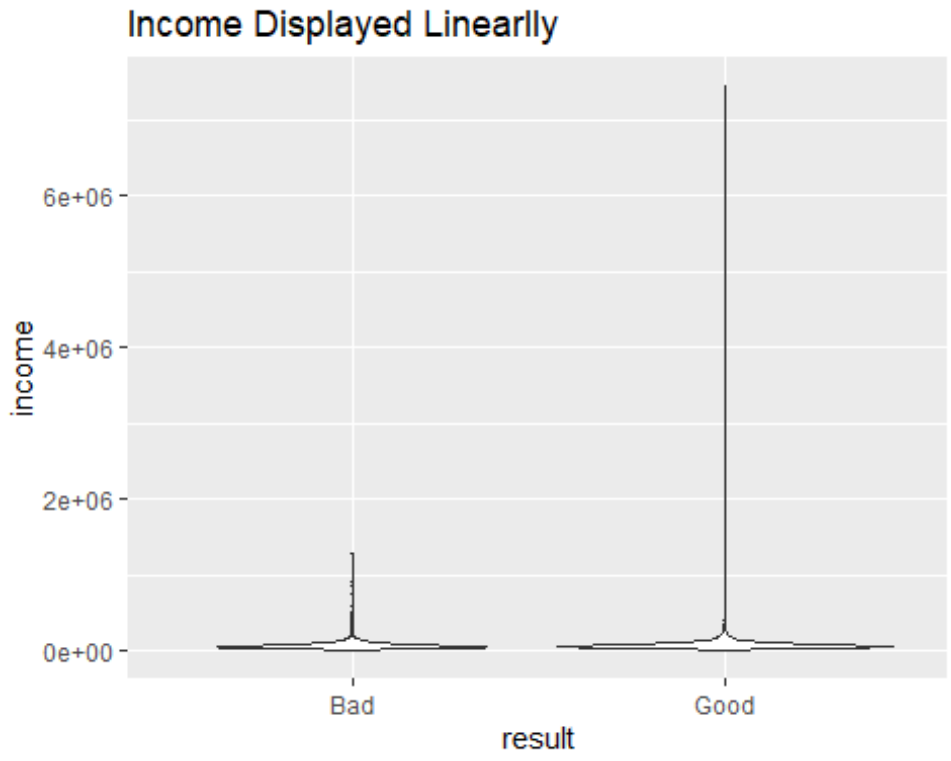


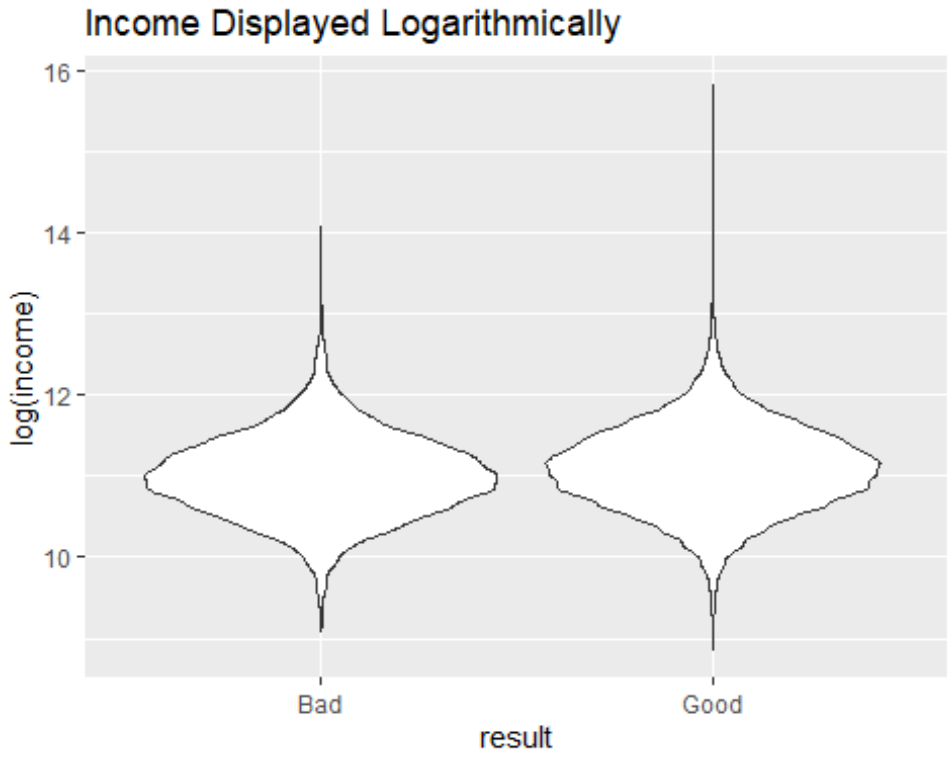
Looking at the reasons for loans, credit card not only had enough data points to be useful, it also was strongly associated with higher grades. I turned the reason into a credit card indicator variable.

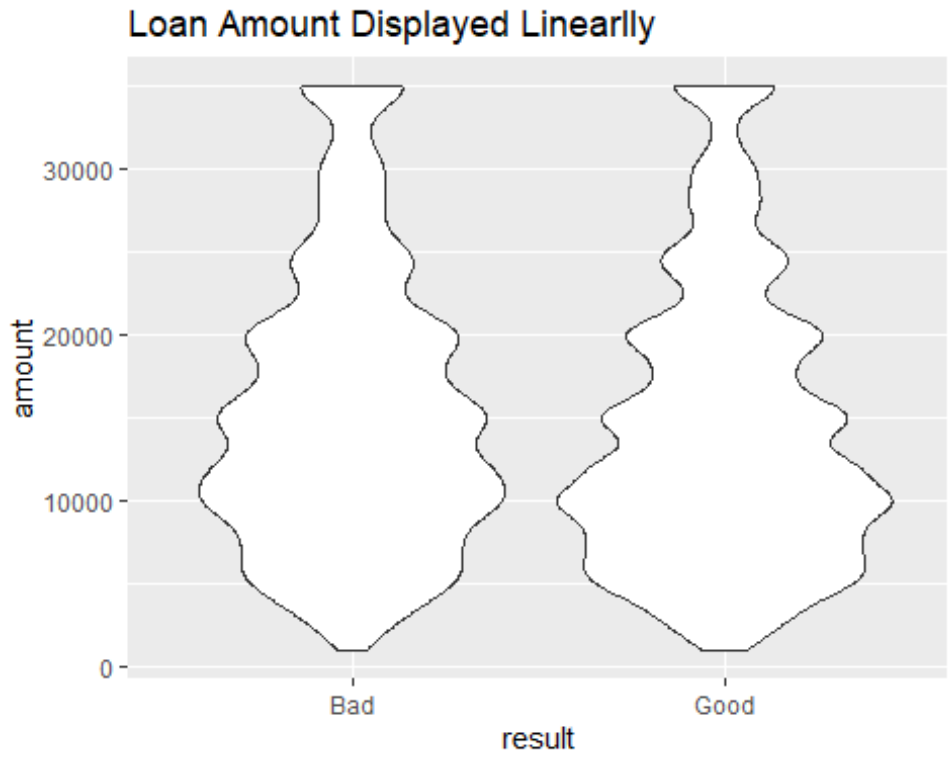


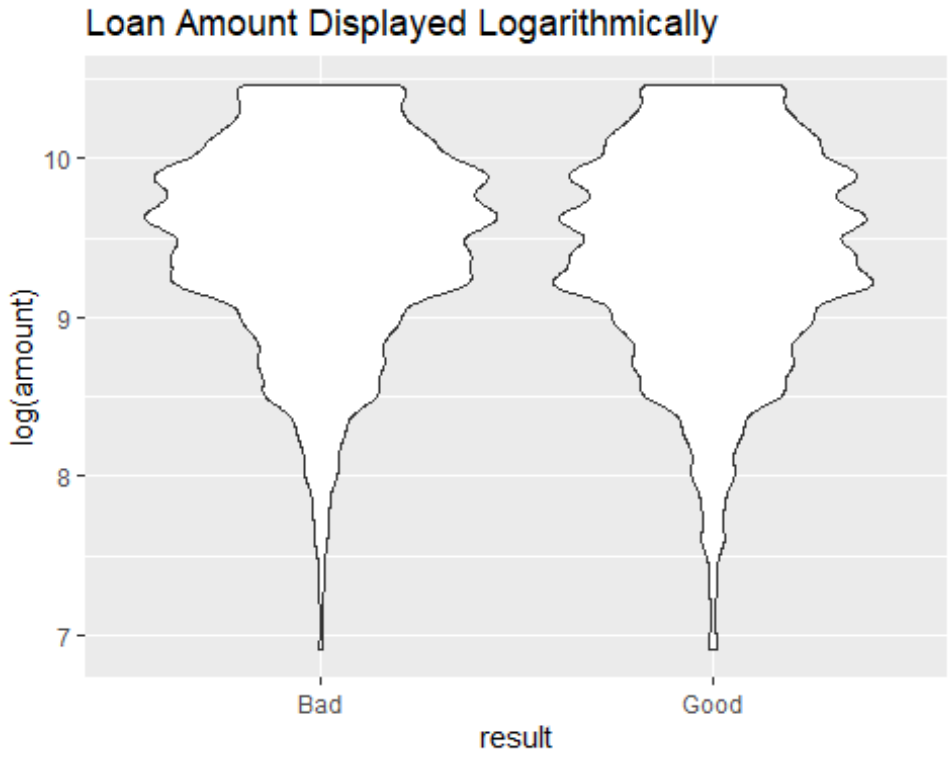
The variables that involved total income, debt or credit levels were heavily skewed. Logarithmically transforming those variables helped reduce skewness. There were still a few outliers, but the transformation helped normalize the data significantly.

Loan amounts were not altered, as they did not span nearly the same range as the transformed variables.


Income Displayed Linearlly
6e+06
4e+06
2e+06
0e+00
income
Bad
Good
result


Income Displayed Logarithmically
16
14
12
10
log(income)
Bad
Good
result


Loan Amount Displayed Linearlly
amount
30000
20000
10000
0
Bad
Good
result


Loan Amount Displayed Logarithmically
log(amount)
10
9
8
7
Bad
Good
result

Section 5: Building The Model

First, I split the data into two sets. 80% of the loans were used to build the models, and the grade variable was dropped from them. The remaining 20% would be held to verify the model's accuracy.

The initial pass of using only first-order variables to build the model looks like the current selection was very good. Open accounts and total revenue balance were not statistically significant at the .05 level, but ended up being significant for loan profitability.

```
##
## Call:
## glm(formula = result ~ amount + term + income + verified + debtIncRat +
##     delinq2yr + revolRatio + totalBal + totalRevLim + totalRevBal +
##     accOpen24 + totalLim + creditcard + openAcc, family = "binomial",
##     data = loans.modelbuilding)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6760   0.3539   0.5451   0.7182   2.1568
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -5.393e+00  4.602e-01 -11.718  < 2e-16 ***
## amount                 -1.956e-05  2.617e-06  -7.471 7.97e-14 ***
## term 60 months         -9.557e-01  3.633e-02 -26.306  < 2e-16 ***
## income                  2.534e-01  5.094e-02   4.974 6.57e-07 ***
## verifiedSource Verified -1.213e-01  4.034e-02  -3.008  0.00263 **
## verifiedVerified        -2.285e-01  4.352e-02  -5.251 1.52e-07 ***
## debtIncRat              -2.865e-02  2.615e-03 -10.955  < 2e-16 ***
## delinq2yrOneOrMore      -2.007e-01  3.881e-02  -5.171 2.32e-07 ***
## revolRatio              -5.756e-01  8.063e-02  -7.139 9.41e-13 ***
## totalBal                -1.596e-01  6.686e-02  -2.387  0.01697 *
## totalRevLim              2.079e-01  2.829e-02   7.348 2.01e-13 ***
## totalRevBal              5.748e-02  3.184e-02   1.805  0.07108 .
## accOpen24               -9.998e-02  5.726e-03 -17.460  < 2e-16 ***
```

```
## totalLim                4.451e-01  8.017e-02   5.553 2.82e-08 ***
## creditcard               1.409e-01  3.848e-02   3.662  0.00025 ***
## openAcc                 -5.910e-03  3.860e-03  -1.531  0.12571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28819  on 27415  degrees of freedom
## Residual deviance: 26352  on 27400  degrees of freedom
## AIC: 26384
##
## Number of Fisher Scoring iterations: 4

## [1] "First-order model"

##                   trueResponses
## estimatedResponses FALSE  TRUE
##              FALSE   570   415
##              TRUE   5433 20998

## [1] 0.7866939
```

Now to use the testing dataset to see how we did.

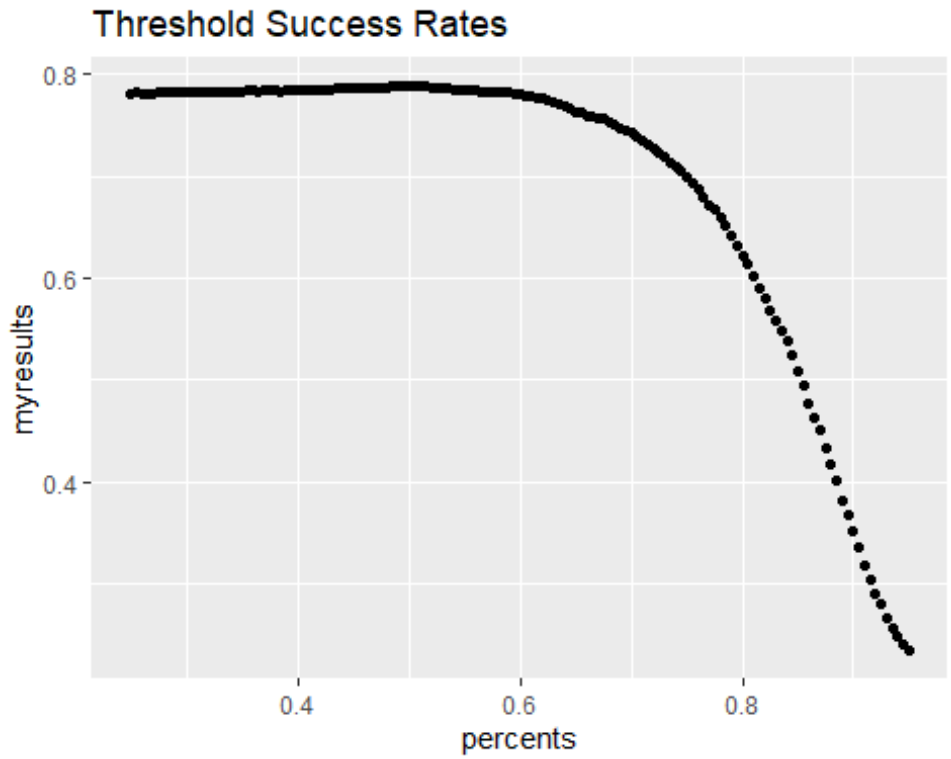
```
##                   trueResponses
## estimatedResponses FALSE TRUE
##              FALSE   159  112
##              TRUE   1342 5241

## [1] 0.7878611
```

We have a success rate in this data set very close to the training set. Let's take it one step further. How can we see if our roughly 78.8% success rate at identifying if a loan will succeed or not is actually worthwhile? Compare the model to the bank's success rate of 78.1%, we're slightly ahead of them. Let's see if we can improve.

Section 6: Optimizing the Threshold for Accuracy

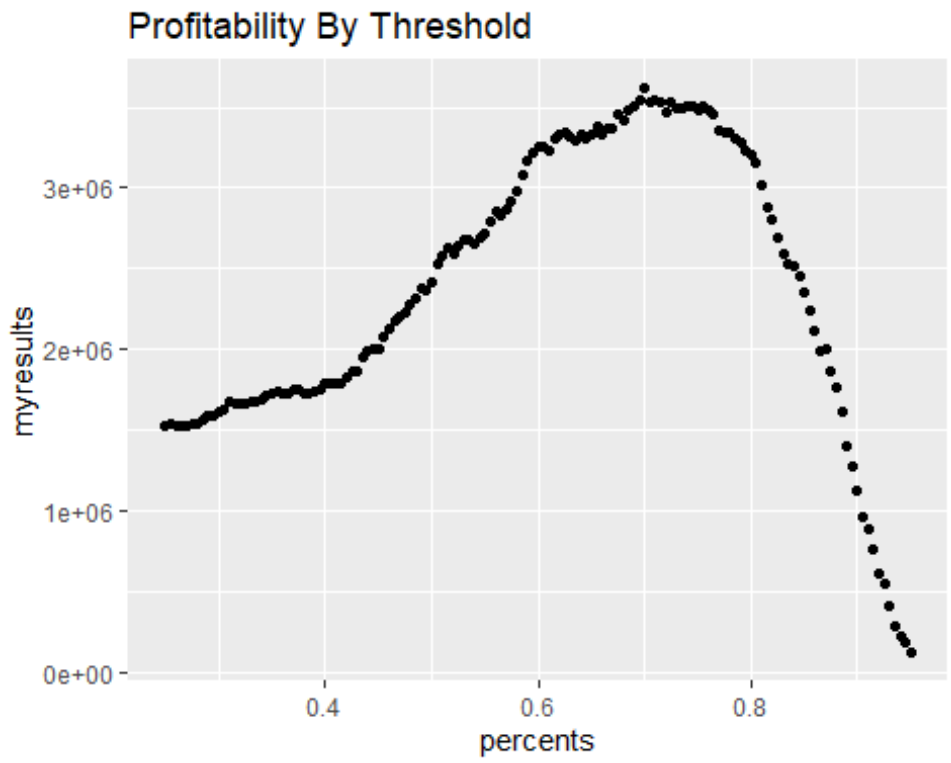
What's the optimal threshold? Let's try other thresholds to see if we can get more accurate results.



The maximum accuracy of 78.9% results from using a threshold of 0.515. There wasn't much difference in prediction quality until around the .60 threshold, when it began to decline slowly, then rapidly. While "being right" is a great reason to build a model, "making money" is another one. How does changing the threshold alter how much money the bank makes?

Section 7: Optimizing the Threshold for Profit

I'll use the same general idea in looking at accuracy, but instead of strictly on pass/fail, I'll include how profitable the loan ends up being. We know that the bank profit of loans that are either fully paid or written off is $1,474,366. If our model was perfect, the maximum possible profit in our sample data is $12,523,305. What's the best the model can do?



Here, the graph peak is at the threshold of 0.7, with a profit of $3,619,615. Interestingly, peak profitability comes with the model's threshold set at a point where it loses accuracy. We can see why if we break the loans into chunks based on the value the model assigned to each one.

```
##   from    to loancount    profits
## 1 0.000 0.515       332 -1154450.2
## 2 0.515 0.700      1212  -990799.7
## 3 0.700 0.800      1575   419076.4
## 4 0.800 1.000      3735  3200539.1
```

The poorly ranked loans didn't just fail a little, they'd become a significant hit to the bank's finances. The loans rated between .515, our most accurate setting, and .70 included 1212 loans that cost the bank $-990,800. While there must be some successful loans the model is rejecting, by sticking to the highest-ranked loans we can significantly increase profitability.

Section 8: Results Summary

I had three goals for creating a method to improve the performance of personal loans: speed, accuracy, and profitability. How did each turn out?

Speed

The amount of information needed to complete a loan application shrunk dramatically. Gone are questions about occupation, or public record queries. Loan officers can do more and better work, and loan seekers feel less stress around a stressful situation.

Accuracy

While it's possible to use the model to predict loan success at a higher rate than a bank, it's not prudent. The threshold of 0.7 that I'm proposing with my model has a 74.3% accuracy. It falls under the bank's success rate of 78.1%, but with good reason.

Profit

That reason is being accurate costs $-990,800. For any two fully paid loans that returns a slight profit, those proceeds are eaten up by one loan that's a drain on the financial books.