Modeling and Predicting Loan Defaults

Kheng Horng Gan Edwin

March 22, 2018

**Introduction:** For this analysis, we used a dataset containing 50,000 loan recipient cases randomly sampled from around the United States. There are 30 variables in our dataset, however, some are irrelevant for identifying the likelihood of an applicant defaulting on their loan. We conducted a logistic regression modeling and analysis to create a model that predicts which applicants are likely to default on their loans. Banks should use this model as it can maximize their profit.

**Data Analysis:** For this analysis ‘Good’= fully paid loans and ‘Bad’= loans that are charged off or defaulted. Loans that are late, currently being paid, or in grace period have been removed from the dataset. We used logistic regression to create a training model for predicting loan status. A default threshold of 0.5 was used to classify the “Good” and “Bad” loans. Although the overall accuracy of the model was high, this was not an effective model to predict if a loan will be repaid because the training dataset was unbalanced.

We used the oversampling technique to ensure that there are an even number of “good” and “bad” loans in the dataset for this section. This fixed the issue that we had with the previous model and allowed for a more accurate one. Our second model was more effective because we increased the percentage of correctly predicted ‘bad’ loans as compared to the previous model. This means that the bank will be able to identify more cases where loans will be defaulted upon, saving their resources for clients who are able to pay off their loans.

Then, we found that a backward selection procedure was the best model because it has a low AIC and 65% accuracy. Next, we experimented with the threshold amount and found that when we increase the threshold, the number of incorrectly predicted loans that are actually good increases. By changing the classification threshold, the overall profit of the bank will fluctuate and the peak profit value for the bank will occur at 0.55 threshold, where the bank grants 65% of loans. To measure whether our model has high predictive ability, we completed a McFadden test and found that our model has some limitations, as the score from this test was low.

**Conclusion:** In summary, we recommend the model with backward selection procedure and a classification threshold of 0.55 as it will produce the highest profit for the bank at the highest level of overall accuracy. Although there are some limitations to the model’s predictive capability, we can predict loan default at 65% accuracy and assure higher profits for the bank at $3,559,022.00.

## Setup

## Load Packages

library(ggplot2)  
library(gridExtra)  
library(mice)   
library(VIM)  
library(leaps)  
library(pscl)  
library(ROSE)  
library(car)  
library(scales)  
library(ResourceSelection)  
library(ggcorrplot)

## Dataset

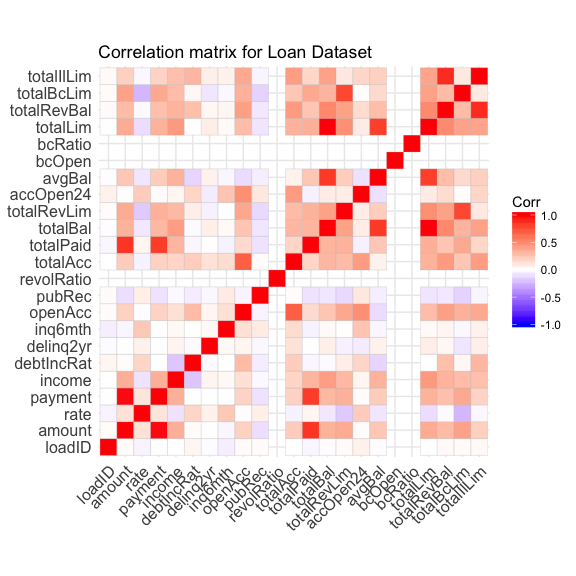
loandata <- read.csv('https://datascienceuwl.github.io/Project2018/loans50k.csv')

## Part 1: Introduction

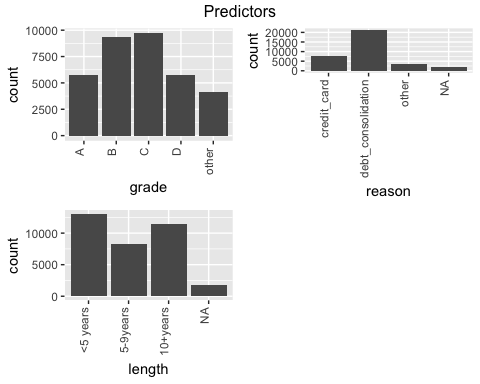
This dataset: loandata contains information about 50,000 loan recipients randomly sampled from around the United States. With this dataset, and for the purpose of this project, a logistic regression analysis will be conducted to attempt to predict which applicants are likely to default on their loans. There are 30 variables, however, for this analysis some are considered irrelevant for identifying the likelyhood of an applicant defaulting on their loan.

## Part 2: Preparing and Exploring the Data

We decided to add a new response variable, called NewStatus having two levels: Good (fully paid loans) and Bad (charged off and defaulted loans). Loans that are late, currently being paid, or in grace period have been removed from the dataset so that we are able to focus only on loans that have been paid in full or were defaulted.



We conducted a correlation test and some variables have been eliminated for this analysis (loadID, bcRatio, bcOpen, revolRatio, employment, state, status) because they do not directly impact the likelyhood that a loan will default. These variables describe the loan cases for filing purposes, but do not provide key insights about the applicant’s financial capacity.



As shown in the charts above, we have consolidated the variables grade and reason to include an ‘other’ cagegory for the seldom-occuring values. For length, we have grouped the data into three broader categories in addition to an ‘N/A’ cagegory for missing values.

There are multiple variables, which had missing values in this dataset, so we used imputation to supplement them through the mice package while minimizing bias in the analysis.

In the table below are the results we generated showing which variables have missing values. Length and reason have the highest amount of missing values in the dataset.

##   
## Missings in variables:  
## Variable Count  
## length 1823  
## reason 2021

Next, two datasets were created from the original one; dataset 1 “training” consists of 80% of the dataset while dataset 2 “test” contains 20%. These two datasets were chosen randomly to minimize bias. The variable ‘totalPaid’ will be removed from the train dataset as it is not a predictor of loan default.

## 80% of the sample size  
smp\_size <- floor(0.80 \* nrow(completedLoans))  
  
## set the seed to make your partition reproductible  
set.seed(123)  
train\_ind <- sample(seq\_len(nrow(completedLoans)), size = smp\_size)  
  
training <- completedLoans[train\_ind, ] #training dataset - 80%  
test <- completedLoans[-train\_ind, ] #test dataset - 20%

## Part 3: First Model and Diagnostics

We used logistic regession to create a training model, then ran the test dataset to predict the loan status for each case. A default threshold of 0.5 was used to classify the “Good” and “Bad” loans.

The training model is summarised below.

train.model <- glm(NewStatus~., data = train\_clean, family = binomial(link = 'logit'))  
summary(train.model)

##   
## Call:  
## glm(formula = NewStatus ~ ., family = binomial(link = "logit"),   
## data = train\_clean)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4622 0.3024 0.5122 0.7291 1.8587   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.127e+00 1.291e-01 24.216 < 2e-16 \*\*\*  
## amount 4.261e-05 1.370e-05 3.110 0.00187 \*\*   
## term2 -9.143e-01 8.853e-02 -10.328 < 2e-16 \*\*\*  
## rate -1.745e+00 1.130e+00 -1.544 0.12251   
## payment -1.729e-03 4.290e-04 -4.032 5.54e-05 \*\*\*  
## grade2 -3.546e-01 7.841e-02 -4.522 6.12e-06 \*\*\*  
## grade3 -7.697e-01 9.554e-02 -8.057 7.83e-16 \*\*\*  
## grade4 -9.148e-01 1.257e-01 -7.277 3.40e-13 \*\*\*  
## grade5 -1.008e+00 1.633e-01 -6.172 6.74e-10 \*\*\*  
## length2 -3.567e-02 3.913e-02 -0.912 0.36196   
## length3 1.954e-02 3.752e-02 0.521 0.60241   
## home2 -1.297e-01 5.524e-02 -2.347 0.01890 \*   
## home3 -2.297e-01 4.074e-02 -5.638 1.72e-08 \*\*\*  
## income 5.136e-07 5.238e-07 0.980 0.32691   
## verified2 -9.928e-02 4.085e-02 -2.430 0.01509 \*   
## verified3 -1.227e-01 4.443e-02 -2.761 0.00577 \*\*   
## reason2 6.390e-02 3.959e-02 1.614 0.10650   
## reason3 1.868e-02 5.922e-02 0.315 0.75243   
## debtIncRat -2.668e-02 2.265e-03 -11.781 < 2e-16 \*\*\*  
## delinq2yr -8.803e-02 1.683e-02 -5.232 1.68e-07 \*\*\*  
## inq6mth -5.389e-02 1.597e-02 -3.374 0.00074 \*\*\*  
## openAcc -1.092e-02 4.866e-03 -2.245 0.02476 \*   
## pubRec -1.050e-02 2.520e-02 -0.417 0.67680   
## totalAcc 1.302e-02 1.915e-03 6.796 1.08e-11 \*\*\*  
## totalBal 8.208e-07 1.135e-06 0.723 0.46948   
## totalRevLim 4.594e-06 1.600e-06 2.871 0.00409 \*\*   
## accOpen24 -6.632e-02 5.986e-03 -11.079 < 2e-16 \*\*\*  
## avgBal 5.404e-06 2.919e-06 1.851 0.06413 .   
## totalLim -4.300e-07 1.035e-06 -0.415 0.67786   
## totalRevBal -6.454e-06 1.489e-06 -4.334 1.47e-05 \*\*\*  
## totalBcLim 3.483e-06 1.521e-06 2.291 0.02198 \*   
## totalIlLim 7.147e-06 1.484e-06 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: 29101 on 27723 degrees of freedom  
## Residual deviance: 26029 on 27692 degrees of freedom  
## AIC: 26093  
##   
## Number of Fisher Scoring iterations: 5

## predGoodBad  
## Bad Loans Good Loans Sum  
## Bad 185 1342 1527  
## Good 145 5259 5404  
## Sum 330 6601 6931

## [1] "Proportion correctly predicted = 0.79"

## [1] "Proportion of correctly predicted Bad loans = 0.12"

## [1] "Proportion of correctly predicted Good loans = 0.97"

Based on the results using a threshold of 0.5, the overall accuracy of the training model is 79%. However, the percentage of accuracy for bad loans being correctly predicted is 12%. Although the overall accuracy of the model is high, this is not an effective model to predict if a loan will be repaid because the training dataset is unbalanced.

## Part 4: Improved Model and Diagnostics

We used the oversampling technique to ensure that there are an even number of “good” and “bad” loans in the dataset. This should fix the issue that we had in the previous section and allow for a more accurate model.

## predGoodBad.oversample  
## Good Loans Bad Loans Sum  
## Bad 512 1015 1527  
## Good 3513 1891 5404  
## Sum 4025 2906 6931

## [1] "Proportion correctly predicted = 0.65"

## [1] "Proportion of correctly predicted Bad Loans = 0.66"

## [1] "Proportion of correctly predicted Good Loans = 0.65"

As shown above, we applied the logistic regression model to our balanced training dataset to form our 1st model and found that the overall accuracy of the model was 65%. The percentage of “bad” loans correctly identified is 66% and the percentage of “good” loans correctly identified is 65%. From these results, we can conclude that the model is more effective because we increased the percentage of correctly predicted ‘bad’ loans as compared to the previous model. This means that the bank will be able to identify more cases where loans will be defaulted upon, saving their resources for clientes who are able to pay off their loans. The AIC for this model is 53201.06.

Next, we used regsubsets to generate the 2nd model for predicting the response variable ‘Newstatus’, with up to 8 the predictor variables in the balanced training dataset. Below is the 2nd model with 8 predictors, with an AIC of 53601.15. None of the VIFS exceed 10, so there is no serious problem with collinearity among these predictors.

From regsubsets (2nd model): Step: AIC=53601.15

NewStatus ~ term + rate + grade + debtIncRat + totalAcc + accOpen24 + totalLim

Then, we used the step function with backward and forward selection to find the 3rd and 4th models for predicting the response variable “Newstatus”. The AIC for the model from the backward selection procedure is 53212.12, which is smaller than the AIC from the forward selection procedure and the regsubsets model. So the 3rd model with backward selection procedure is the best model.

The following predictors are important and were applied in our automatic models selection:

From backward (3rd model): Step: AIC=53212.12

NewStatus ~ amount + term + rate + payment + grade + length + home + income + verified + reason + debtIncRat + delinq2yr + inq6mth + openAcc + pubRec + totalAcc + totalBal + totalRevLim + accOpen24 + avgBal + totalRevBal + totalBcLim + totalIlLim

From forward (4th model): Step: AIC=53214.11

NewStatus ~ grade + debtIncRat + term + totalLim + accOpen24 + totalAcc + delinq2yr + home + verified + rate + payment + amount + totalBal + openAcc + totalIlLim + inq6mth + totalRevBal + totalRevLim + reason + length + pubRec + totalBcLim + income + avgBal

## pred.auto.loan  
## Good Loans Bad Loans Sum  
## Bad 512 1015 1527  
## Good 3513 1891 5404  
## Sum 4025 2906 6931

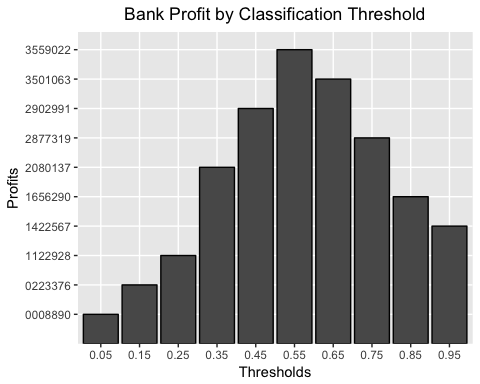
## [1] "Proportion correctly predicted = 0.65"

## [1] "Proportion of correctly predicted Bad loans = 0.66"

## [1] "Proportion of correctly predicted Good loans = 0.65"

Now that we have found our best model, we tested it out with a 0.5 classification threshiold. After running the test, the results above show it is just as effective as the first model using the balanced training dataset with 65% accuracy. The percentage of “bad” loans correctly identified is 66% and the percentage of “good” loans correctly identified is 65%. None of the VIFs exceed 10, so there is no serious problem with collinearity among these predictors

To measure whether our model has high predictive ability, we completed a McFadden test. We found that the McFadden score for our best model is 0.1153736, which means that it has relatively low explanatory power. With this evidence we can condlude that our model has some limitations when it comes to it’s predictive capabilities.



## Part 5: Tuning the Predictions and Profit Analysis

We experimented with the threshold amount and found that when we increase the threshold, the number of incorrectly predicted loans that are actually good increases. By changing the classification threshold, the overall profit of the bank will fluctuate. As we can see this in the graph above, from a threshold amount of 0-0.55 the bank’s overall profit will increase and from values 0.55-1, the bank’s profits decrease. This means that the peak profit value for the bank will occur at 0.55 threshold.

## Part 6: Results Summary

In summary, we recommend the model with backward selection procedure and a classification threshold of 0.55 as it will produce the highest profit for the bank at the highest level of overall accuracy. Although there are some limitations to the model’s predictive capability, we can predict loan default at 65% accuracy and assure higher profits for the bank at $3,559,022.00.