Default Loan Prediction

Kenda Ransom

December 31, 2018

# Executive Summary

# Introduction

The task to be completed is to create a model that will accurately predict loans that will default, while minimizing the number of good loans that get predicted as bad. By predicting which loans will default and minimizing the misclassification of good loans, the bank will be able to optimize its profits.

In order to create this model, the bank provided a data set with 50,000 random samples. The data set has 32 potential predictor variables. The data is reviewed for missing data, anomalies, and other data cleansing issues. These issues will be dealt with appropriately and a logistics model will be built using various combinations of the predictor variables. After building the model, the model will be used to predict if loans will be good or bad. Finally, the model will be optimized to ensure the most profit for the bank, while still predicting bad loans.

After implementing the aforementioned techniques, the final model had a 73% overall accuracy rate, correctly predicted 49% of bad loans and 80% of good loans, and saw a realized profit of $1.9 million dollars.

# Preparing and Exploring the Data

## Data Exploration

The first step is to create a summary of the data. This will give an idea of the types of variables available, what, if any values, are missing, and give insight on how variables might be simplified, combined, or transformed.

summary(loans)

From the summary, there are 24 quantitative and 8 qualitative variables. The columns of **loanID** and and **employment** are removed due to a large number of levels that are not able to be combined into meaning data. For this analysis, the concern is only with loans identified as **Good** (Fully Paid Off) or **Bad** (Charged Off or Default).

A new, binary response variable, **resp**, will be created from the **status** column in the data set. After creating the **resp** column, the **status** column is removed. A new data frame is built, with the **loanID**, **employment**, and **status** variables removed.

loan\_status = c("Fully Paid","Charged Off","Default")  
kp.rows = which(loans$status %in% loan\_status)  
loans.df = loans[kp.rows, ]  
loans.df$resp = ifelse(loans.df$status == "Fully Paid",1,0)  
loans.df$resp = factor(loans.df$resp,labels = c("Bad","Good"))  
loans.df[,c("loanID", "employment", "status")] = list(NULL)  
loans.df = data.frame(loans.df)

## Data Quality

Now that the unnecessary predictor variables have been removed, the quality of the data can be checked. In reviewing the summary of the data, there are 19 variables that are missing data points.

From the summary of the data, the 3 variables with the highest count of NA’s are:

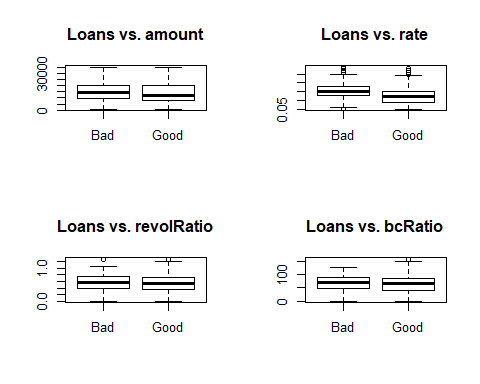
* **RevolRatio**
* **BcOpen**
* **BcRatio**

Using the *VIM* package in **R**, the 3 variables with the highest NA count makes up approximately 1% of the data. Because the data set is large, it is reasonable to remove the rows that have NA’s in the **RevolRatio**, **BcOpen**, and **BcRatio** columns.

## Data Distribution - Quantitative

This section evaluates the distribution properties of the quantitative predictor variables. The distribution of the quantitative variables is analyzed using box plots and histograms. The boxplots and histograms show the data to be from a non-normal distribution. Because the data set is large and the model being created is a logistics model, the data does not need to come from a normal distribution.

The examples below show the relationship between 4 quantitative variables and a good loans, and 4 quantitative variables and a bad loan.



The good loans have a lower median **amount** and a lower median **rate** value. There are some extreme outliers in the **rate** variable for good and bad loans. There are a few outliers in the **revolRatio** and **bcRatio** variables. The variance between these 4 variables and a good or bad loan appear to be about the same.

Because the data set is large and a logistics model is being created, the data set does not need to be normally distributed.

## Data Distribution - Qualitative

The qualitative variables will be analyzed. Looking at the summary of the qualitative variables, there are variables with factor levels that can be combined to help create a more robust data set.

There is room for improving the quality of the qualitative variables. There are 13 levels in the **length** variable, which describes how long a loan applicant has been employed. This can be consolidated down to 4 levels:

* 0 to 5 - level 1
* 6 to 9 - level 2
* 10+ - level 3
* n/a - level 4

The n/a level us kept in the dataset. The assumption is that employment length of n/a responds to unemeployement, which may be important in determining if a loan will be repaid or not.

The **verified** variable has 3 levels which can be condensed down to 2 levels:

* verified
* not verified

The **reason** variable has 14 levels. These can be condensed down to 3 levels:

* debt
* purchases
* other

levels(qual$reason) = c(levels(qual$reason),"purchases")  
  
qual$reason[qual$reason == 'car'] = 'purchases'  
qual$reason[qual$reason == 'credit\_card'] = 'purchases'  
qual$reason[qual$reason == 'home\_improvement'] = 'purchases'  
qual$reason[qual$reason == 'house'] = 'purchases'  
qual$reason[qual$reason == 'major\_purchase'] = 'purchases'  
qual$reason[qual$reason == 'medical'] = 'purchases'  
qual$reason[qual$reason == 'moving'] = 'purchases'  
qual$reason[qual$reason == 'renewable\_energy'] = 'purchases'  
qual$reason[qual$reason == 'small\_business'] = 'purchases'  
qual$reason[qual$reason == 'vacation'] = 'purchases'  
qual$reason[qual$reason == 'wedding'] = 'purchases'

The last qualitative variable to be address is the **state** variable. This variable has 51 levels, 50 states plus DC. The states will be condensed into 5 regions:

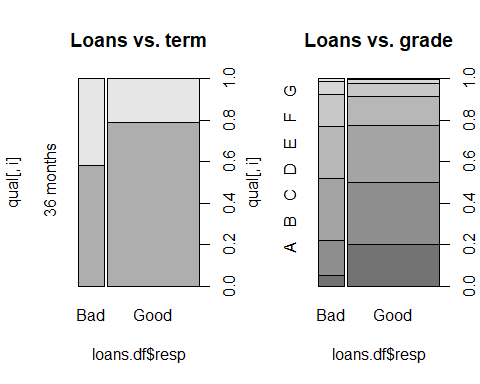
* NorthEast
* SouthEast
* Midwest
* SouthWest
* West

The NorthEast region will include DC.

levels(qual$state) = c(levels(qual$state),"N", "SE", "MW", "SW", "W")  
  
for(i in c('ME','NH','VT','NY','PA','MA','RI','CT','NJ','DE','MD','DC')){  
 qual$state[qual$state == i] = 'N'  
}  
  
for(i in c('WV','VA','KY','TN','NC','SC','AR','LA','MS','AL','GA','FL')){  
 qual$state[qual$state == i] = 'SE'  
}  
  
for(i in c('ND','SD','NE','KS','MN','IA','MO','WI','IL','IN','MI','OH')){  
 qual$state[qual$state == i] = 'MW'  
}  
  
for(i in c('AZ','NM','TX','OK')){  
 qual$state[qual$state == i] = 'SW'  
}  
  
for(i in c('AK','HI','WA','MT','OR','ID','WY','CA','NV','UT','CO')){  
 qual$state[qual$state == i] = 'W'  
}

The examples below show the relationship between 4 qualitative variables and a good loans, and 4 qualitative variables and a bad loan after reducing the number of factors in the previously mentioned variables.

par(mfrow = c(1,2))  
for(i in c(1,2)){  
 plot(qual[,i]~loans.df$resp, main = paste("Loans vs.", colnames(qual)[i]))  
}



par(mfrow = c(1,1))

Based on these plots, bad loans have a longer terms. Based on the definition of a risky loan, the higher the alphabeta the larger the risk, the majority of bad loans are grade C or higher.

The last data check for the qualitative variables is the Chi-Square test for independence, which will test if the response and qualitative variables are associated or not.

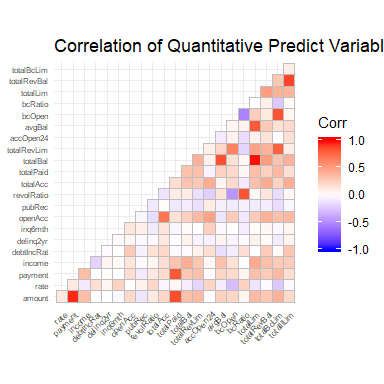
The results of the Chi-Squared test for independence at the 5% significance level indicate that there is statistical evidence to claim that the **resp** variable and each of the categorical variables are associated.

The final data set will have 22 quantitative variables and 7 qualitative variables, not including the response variable. This data set is then split into 80% training data and 20% testing data. The **totalPaid** column will be removed from the training data set, as it can not be used as a predictor in the model, and the **resp** variable will be added to the training data set.

# First Model and Diagnostics

## Correlation and Interaction

The quantitative variables are examined for correlation and higher order interactions with the response variable. A correlation graph below will show which variables are highly correlated.



Based on the correlation matrix plot, the highest correlation values are between **amount** and **payment**, and **totalBal** and **totalLim**.The findCorrelation() function is run with a 0.9 cutoff to determine which correlation variables should be removed from the data set.

findCorrelation(corrdata, cutoff=0.9, names = TRUE)

## [1] "totalLim" "amount"

The variables of **amount** and **totalLim** will be removed from the training data set when the model is created.

## Model and Model Diagnostics

The data set created is now used to create a first order logistic model. The output of the logistics model is used to predict if loans will be classified as good or bad. Below is the full model using all of the predictor variables. The model metrics of AIC, psuedo r^2, and Hosmer-Lemeshow Goodness-of-Fit are reported

The first order logistics model is as follows:

model1 = glm(resp ~ payment + rate + income + debtIncRat + delinq2yr + inq6mth + openAcc + pubRec + revolRatio + totalAcc + totalBal + totalRevLim + accOpen24 + avgBal + bcOpen + bcRatio + totalRevBal + totalBcLim + totalIlLim + term + grade + length + home + verified + reason + state, data = train\_data, family = "binomial")

The full model has an AIC value of 25900, a psuedo r^2 of 0.11, and a Goodness-of-Fit p-value of <2.2e-16. The full model is used with the predict() function to classify if loans will be good or bad. The classification of good or bad is based on a 0.5 threshold.

|  |  |  |
| --- | --- | --- |
|  | Actual Bad | Actual Good |
| Predicted Bad | 179 | 117 |
| Predicted Good | 1288 | 5271 |

The full model overall accuracy is 80%. This first order model classifies 12% of bad loans correctly, and classifies 98% of good loans correctly. Since the purpose of the model is to predict if someone will default on their loan, this initial model does not meet the stated purpose. This model is not effective at predicting loans which will go into default.

# Improved Model and Diagnostics

The previous model was tested on an a data set that did not have an equal distribution of good and bad loans. Since the objective of this model is to attempt to accurately predict all the bad loans correctly, the data needs to be equally distributed between good and bad loans.

The loans classified as bad in the original training data set are re-sampled with replacement to create a new data set that has an equal number of good and bad loans and includes the original bad loans.

# create addition bad loan data  
bad\_rows = which(train\_data$resp == "Bad")  
bad\_loans = train\_data[bad\_rows,]  
ss = length(which(train\_data$resp == 'Good'))-length(which(train\_data$resp == 'Bad'))  
  
set.seed(123)  
  
ind = sample(nrow(bad\_loans),ss, replace = TRUE)  
  
bl\_new = bad\_loans[ind,]  
  
new\_train\_data = rbind(train\_data,bl\_new)  
  
new\_train\_data = droplevels(new\_train\_data)  
summary(new\_train\_data$resp)

## Bad Good   
## 21379 21379

A new logistic regression model is built with the new training data set, which has the sample predictor variables as the first order model.

new\_model1 = glm(resp~.,data=new\_train\_data, family = "binomial")

The test data set is used with the newly created logistic model to create a new contingency table of good and bad loans with a 0.5 threshold.

|  |  |  |
| --- | --- | --- |
|  | Actual Bad | Actual Good |
| Predicted Bad | 1006 | 1938 |
| Predicted Good | 461 | 3450 |

The overall accuracy of the new logistic model using the new training data set decreased to 65% from 80%. The ability of the new model to accurately predict bad loans increased from 12% to 69%. The ability of the new model to accurately predict good loans decreased from 98% to 64%. Based on the purpose of the model, the new model with the new training data set is more effective than the original model due to the increase in prediction of bad loans. This model can be improved upon using automatic model selection processes.

## Automatic Model Selection

A more accurate model can be produced through the use of the automatic model selection functions in **R**. Three different models will be compared to each other to aid in selecting the best model:

* Full Model (new\_model1, previously calculated)
* Step Model (new\_model2)
* Regsubset Model (new\_model3)

The three models are compared using ANOVA, AIC, pseudo r^2, and Hosmer-Lemeshow Goodness-of-Fit. Based on these results, a “best” model will be selected.

The second model uses step function with the default forward and backward direction.

# Step function   
step\_model = step(glm(resp~., data = new\_train\_data, family = "binomial"))

The step function with the default direction generated the following model:

new\_model2 = glm(resp ~ rate + payment + debtIncRat + delinq2yr + inq6mth + openAcc +   
 pubRec + revolRatio + totalAcc + totalBal + totalRevLim +   
 accOpen24 + bcRatio + totalRevBal + totalBcLim + totalIlLim +   
 term + grade + length + home + verified + reason + state, data = new\_train\_data, family = "binomial")

This model has 23 predictor variables, which is 3 less than the full model. The variables of **income**, **bcOpen**, and **avgBal** were removed from the step model.

The third model uses the automatic model selection regsubset function with the maximum subsets equal to the number of predictor variables generated from the step function. The best model from the regsubset function is selected based on the adjusted r^2 value.

# Regsubsets function  
rgs\_model = regsubsets(resp~.,nvmax=23,data=new\_train\_data)

The best regsubset model with the highest adjusted r^2 (adjr^2 = 0.151) is:

new\_model3 = glm(resp ~ payment + grade + debtIncRat + term + totalBal + accOpen24 + length + totalAcc + revolRatio + home + state + delinq2yr + rate + totalIlLim + totalRevBal + totalRevLim , data = new\_train\_data, family = "binomial")

This model has 16 predictor variables, which is 11 less than the full model. The variables of **income**, **bcOpen**, **avgBal**, **inq6mth**, **openAcc**, **pubRec**, **bcRatio**, **totalBcLim**, **verified**, **reason** were removed from the regsubset model.

The comparison of the three models using pseudo r^2, Hosmer-Lemeshow Goodness-of-Fit, and AIC is listed below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | AIC | Pseudo r^2 | GoF |
| full model | 52337 | 0.12 | 0 |
| step model | 52331 | 0.12 | 0 |
| regsubset model | 52374 | 0.12 | 0 |

The pseudo r^2 value for each of the three models is the same at 0.12. There is a difference in the AIC values of each model. The full model had an AIC of 52337, while the step and regsubset functions have AIC’s of 52331 and 52374 respectively. The null hypothesis of the Hosmer-Lemeshow Goodness-of-Fit test is rejected for all three models. None of these models are a good fit for the data.

The ANOVA results are different when comparing the step model and regsubset model to the full model.

# Compare Regsubset Model to Full Model  
anova(new\_model3, new\_model1, test = "Chi")  
# Compare Step Model to Full Model  
anova(new\_model2, new\_model1, test = "Chi")

The p-value reported from the comparison of the full model and the regsubset model is very small (p=1.129e-08). This low p-value means the null hypothesis should be rejected. There is sufficient evidence to claim these models are different. This eliminates the use of regsubset model since the outputs of this model is statistically different from the full model.

The p-value reported from the comparison of the full model and the step model is very large (p=.9533). This high p-value means the null hypothesis should not be rejected. There is not sufficient evidence to claim these models are different.

Based on the elimination of the regsubset model and the rejection of the null hypothesis for the step model, the step model is selected as the best model since it has less predictor variables than the full model, but is not statistically different than the full model.

Even though the models are not a good fit for the data, that does not necessarily mean the models will be bad at maximizing the profits for the bank. Even though the models have little predictive power and low practicality, that is a reflection of the data used to build each model. Better models can be built with a better dataset. The practicality of the best model can be optimized by tuning the threshold which defines a good loan and bad loan. This would make the model deverived off of this data set more valuable to the bank.

## Model Tuning

A deeper examination of the best model is warranted to see if it is possible to increase the Hosmer-Lemeshow Goodness-of-Fit test results. Collinearity is checked for using the vif() function and the best model.

require(car)  
vif(new\_model2)

From the vif() function, the **rate** and **totalRevlBal** variables show collinearity. Some of the factors within the **grade** variable are collinear as well. Because all of the factors within the **grade** variable are not collinear, this variables will not be removed. The **rate** and **totalRevlBal** variables will be removed one at a time in an effort to continue to simplify the model. The resulting models are compared to the step model using Hosmer-Lemeshow Goodness-of-Fit, AIC, and ANOVA.

The p-value reported from the comparison of the model with **rate** and the model without **rate** is very small (p=0.0002327). This low p-value means the null hypothesis should be rejected. There is sufficient evidence to claim these models are different. The Hosmer-Lemeshow results did not change between the models. The AIC of the model with **rate** removed (AIC=52342) is higher than the AIC of the model with **rate** included (AIC=52331).

Removing the **rate** variables made the model worse. The **rate** variable will be added back into the model since collinear variables do not affect the output of a logistic model. Next the **totalRevBal** variable will be removed from the model.

The p-value reported from the comparison of the model with **totalRevBal** and the model without **totalRevBal** is again very small (p=1.286e-11). This low p-value means the null hypothesis should be rejected. There is sufficient evidence to claim these models are different. The Hosmer-Lemeshow results did not change between the models. The AIC of the model with **totalRevBal** removed (AIC=52375) is higher than the AIC of the model with **totalRevBal** included (AIC=52331).

Removing the **totalRevBal** variables made the model worse. The **totalRevBal** variable will be added back into the model.

## Best Model Classification Results

With the threshold at 0.5, the metrics for the full and step models are below:

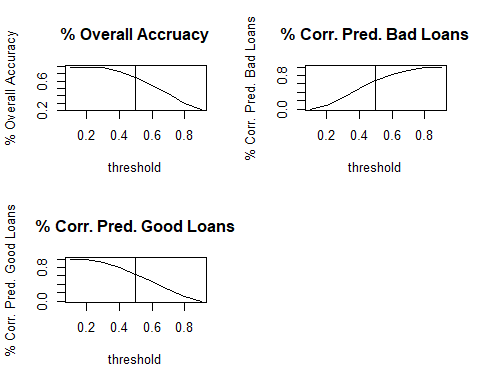
|  |  |  |  |
| --- | --- | --- | --- |
|  | % Overall Acc. | % Correctly Predicted Bad Loans | % Correctly Predicted Good Loans |
| full model | 0.65 | 0.69 | 0.64 |
| step model | 0.65 | 0.69 | 0.64 |

The metrics of the step model are the same as the full model. The step model will be used as the best model for tuning the loan predictions.

# Tuning the Predictions and Profit Analysis

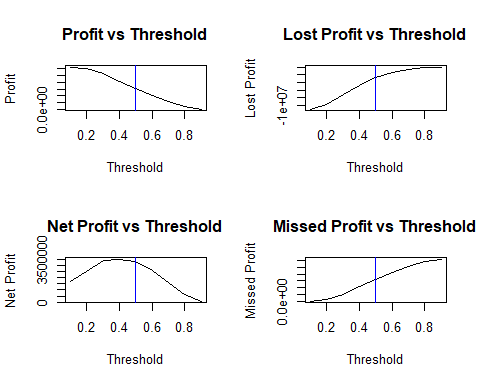
The bank will increase its profits by decreasing the amount of bad loans it approves. The default prediction model needs to optimize the bank’s profit by reducing the amount of bad loans approved, while minimizing the amount of good loans that are not approved. A profit per loan value can be calculated using the **totalPaid** - **amount**. The profit per loan can be viewed as the actual profit of good loans or the lost profit of bad loans. By tuning the threshold for loan classification, the change in profit and lost profit can be analyzed. Also the change in number of loans approved can be analyzed.

The prediction threshold is modified from 0.1 to 0.9. The prediction threshold is the tuning parameter for increasing profitability within the bank.



From a prediction threshold of 0.5, the overall model accuracy increases as the prediction threshold decreases. As the prediction threshold decreases from 0.5, the percentage of correctly classified bad loans decreases while the percentage of correctly classified good loans increases. With an increasing prediction threshold, the overall model accuracy decreases. The percentage of accurately classified bad loans increase and the percentage of accurately classified good loans decrease with a increasing predition threshold.

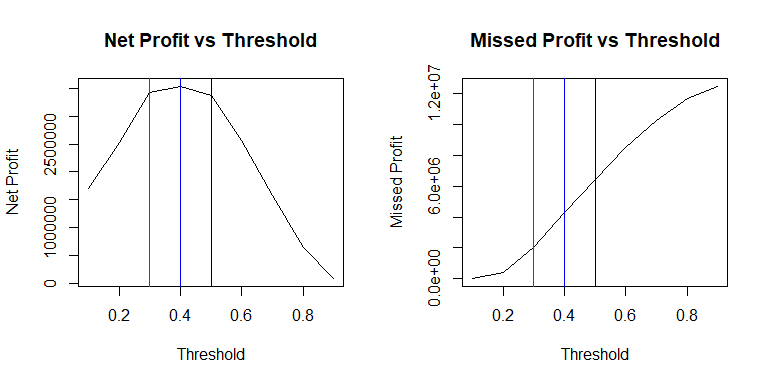
From a loan approval perspective, it is easier to understand the affect of the model statistics in terms of actual profit and profit loss. Profit is defined as actual good loans that are classified as good. Lost Profit is defined as actual bad loans that are classified as good. Missed Profit is defined as actual good loans that are classified as bad.



When the prediction threshold decreases from 0.5, the profit and lost profit of the bank both increases. The lost profit is considered an increase as the negative value gets larger. This is due to the increase in bad loans being approved. The overall net profit, profit minus the absolute value of lost profit, of the bank is small and not optimized.

When the prediction threshold increases from 0.5, profit and lost profit of the bank both decrease. This is due to the increase in good loans not being approved. This aligns with the low actual profit that is seen with an increasing prediction threshold. Again, the overall net profit of the bank is small and not optimized. This is especially evident as the missed profits of the bank get larger as the prediction threshold increases.

Ideally, the bank wants to approve more loans to raise profits, while minimizing the number of bad loans costs the bank. The prediction model should be optimized to meet this criteria.



A threshold of 0.4 has the largest net profit of million dollars, with a missed profit of million dollars. At the 0.3 threshold, the net profit is million dollars, but the missed profit is only million dollars. The difference in the actual profit and missed profit between the 0.3 and 0.4 threshold is approximately million dollars.

# Results Summary

|  |  |  |
| --- | --- | --- |
|  | Actual Bad | Actual Good |
| Predicted Bad | 713 | 1102 |
| Predicted Good | 754 | 4286 |

|  |  |  |
| --- | --- | --- |
| Overall Acc. | Sensitivity | Specificity |
| 0.73 | 0.49 | 0.8 |

Based on the above profit analysis, the optimized threshold for this model is 0.4. At this threshold, the model has an overall accuracy of 73%. This model will correctly predict 49% of actual bad loans accurately and 80% of actual good loans correctly. This model will create a type I error 15% of the time by classifying 1102 good loans as bad. The model will create a type II error 61% of the time by classifying 754 bad loans as good. Even though the type II error rate is high, consideration must be given to the fact that the original loan data set had an unbalanced distribution of loans. The cost associated with the type II error is minimized as compared to the actual profit and the low type I error rate.

This model should generate the bank an additional profit of million dollars, while approving a total of 5388 loans, 4286 of which will be good and 1102 of which will be bad.