DS 705 Final Project

Jim Ryan

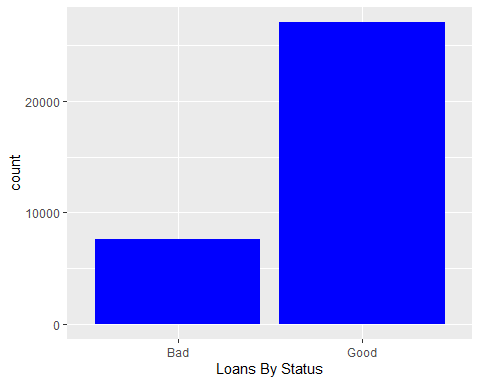
04/06/2020

Part 2: Introduction

For this project I will use logistic regression to predict which applicants are likely to default on their loans. I am going to use the loans50k data set that was given to us as part of this assignment. This data set contains 50,000 loans in various statuses and various amounts. There are 32 available variables in the data set. My response variable will be the status variable. I will start by preparing and cleaning the data.

Part 3: Preparing and Cleaning the Data

The data contains different variables collected as either part of the application process or information from when the loan was disbursed. I am trying to predict which variable or variables will allow me to predict whether a loan becomes default or not. I removed all loans that did not have a status of ‘Fully Paid’, ‘Default’ or ‘Charged Off’. Then I copied the response variable – status – into another field called status2. I then updated the status of loans that were Fully Paid to ‘Good’ and the ones that were ‘Default’ or ‘Charged Off’ to ‘Bad’.

There are 27,074 loans or 78% with a status of ‘Good’ and 7,581 loans or 22 percent with a status of ‘Bad’. Here is a Histogram of that shows the number of good and bad loans after the status has been updated.  


I used sapply with a sum on the NA in the data set to determine which variable had NA in them and how many there were. I found that the only 3 variables that had NA values in them were revolRatio (15 NA), bcOpen (360 NA) and bcRatio (384 NA).Since the missing values were only around 1 percent of the total number for these variables, I imputed them using the mean of all the values for that specific variable.

## loanID amount term rate payment grade   
## 0 0 0 0 0 0   
## employment length home income verified status   
## 0 0 0 0 0 0   
## reason state debtIncRat delinq2yr inq6mth openAcc   
## 0 0 0 0 0 0   
## pubRec revolRatio totalAcc totalPaid totalBal totalRevLim   
## 0 15 0 0 0 0   
## accOpen24 avgBal bcOpen bcRatio totalLim totalRevBal   
## 0 0 360 384 0 0   
## totalBcLim totalIlLim status2 amount2   
## 0 0 0 0

Then I focused my attention on the employment variable. There were 15,268 different values for employment including 1918 where the value is blank. I selected all rows where the value had a count greater than 100. I updated 1789 values of the employment variable that had the word manager in the to be just ‘Manager’. There were 934 different variations of the word ‘Teacher’, so I updated them all to be teacher.

The ‘Grade variable was updated from ‘A’ thru ‘G’ to 0 thru 6 and made into a factor. I update the ‘verified’ variable by updating those with a value of ‘Source Verified’ to ‘Verified’ and made that variable a factor. I did something similar with the home variable. I set the variable to OWN when the value was mortgage so then there were just 2 categorical variables - ‘own’ and ‘rent’ made the variable a factor. The variable term only had 2 values (36 months and 60 months) so the variable term2 was updated to a factor.

sqldf("Select count(\*) from LoansDT where employment like '%manager%' ")

## count(\*)  
## 1 5228

# update all employment rows with word manager in the to Manager  
LoansDT <- sqldf(c("update LoansDT set employment = 'manager' where employment like '%manager%'","Select \* from LoansDT"))  
  
# update all employment rows with word manager in the to manager  
LoansDT <- sqldf(c("update LoansDT set employment = 'Teacher' where employment like '%teach%'","Select \* from LoansDT"))  
  
  
#Update grade to numeric (0 thru 6)  
 LoansDT <- sqldf(c("UPDATE LoansDT SET grade = CASE grade WHEN 'A' THEN 0 WHEN 'B' THEN 1 WHEN 'C' THEN 2 WHEN 'D' THEN 3 WHEN 'E' THEN 4 WHEN 'F' THEN 5 WHEN 'G' THEN 6 END","Select \* from LoansDT"))  
# set grade as a factor  
 LoansDT$grade <- as.factor(LoansDT$grade)  
  
#Update home to OWN when value is mortgage, set own = 1 and rent = 0 and maake it a factor   
 LoansDT <- sqldf(c("Update LoansDT set home = 'OWN' where home = 'MORTGAGE'","Select \* from LoansDT"))  
 LoansDT$home <- as.factor(LoansDT$home)  
  
# Update Source Verified amd Verified to 1 Not verified as 0 make it a factor  
 LoansDT <- sqldf(c("UPDATE LoansDT SET verified = 'Verified' where verified = 'Source Verified'","Select \* from LoansDT"))   
  
 LoansDT$verified <- as.factor(LoansDT$verified)  
   
# make term column a factor  
LoansDT$term <- as.factor(LoansDT$term)

I removed the following variables because more than 25 percent of them were 0: pubRec, delinq2yr, and inq6mth. I also dropped the original status variable.

# remove original status and original term columns  
LoansDT$status <- NULL  
sqldf ("select count (\*) from LoansDT where pubRec = 0 ")

## count (\*)  
## 1 28076

sqldf ("select count (\*) from LoansDT where delinq2yr = 0 ")

## count (\*)  
## 1 27600

sqldf ("select count (\*) from LoansDT where inq6mth = 0 ")

## count (\*)  
## 1 19237

sqldf ("select count(distinct(employment)) from LoansDT")

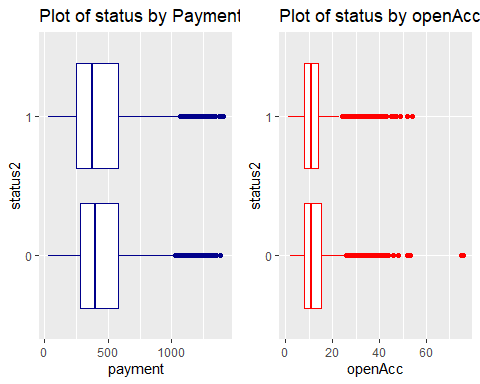
## count(distinct(employment))  
## 1 13368

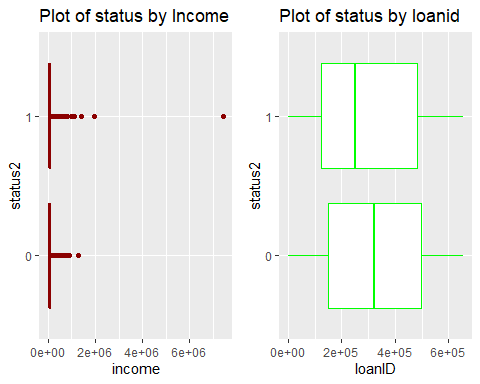
# following columns are removed as over 25% of values are blank or 0  
LoansDT$pubRec <- NULL   
LoansDT$delinq2yr <- NULL   
LoansDT$inq6mth <- NULL

Part 4:Exploring and Transforming the Data

Now I delve into some data exploration and transformation. First I do some exploration on the data set as a whole. I create box plots of payment and status to see if there is an observable relationship between payment and status. There is no observable relationship between payment and status for the entire dataset. Then I did the same for openAcc income and loanID The only one that showed an obvious relationship was totalBal. The ‘LoanID’ variable looks constant no matter what the status. I assume this is just a loan Identifier and will drop it. The other 3 variables I will keep for for part 5.

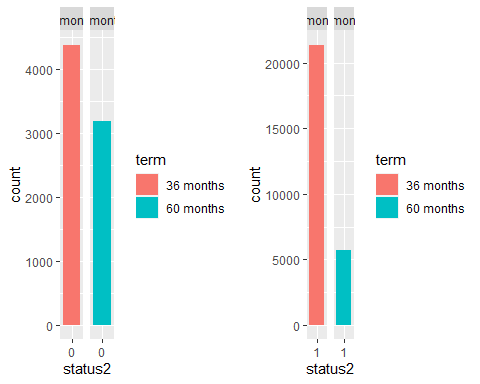
bp1 <- ggplot(aes(x=payment,y=status2),data=LoansDT)+  
 geom\_boxplot(color='darkblue') + ggtitle("Plot of status by Payment")  
  
bp2 <- ggplot(aes(x=openAcc,y=status2),data=LoansDT)+  
 geom\_boxplot(color='red') + ggtitle("Plot of status by openAcc")  
  
bp3 <- ggplot(aes(x=income,y=status2),data=LoansDT)+  
 geom\_boxplot(color='darkred') + ggtitle("Plot of status by Income")  
  
bp4 <- ggplot(aes(x=loanID,y=status2),data=LoansDT)+  
 geom\_boxplot(color='green') + ggtitle("Plot of status by loanid")  
  
multiplot(bp1,bp2,cols=2)



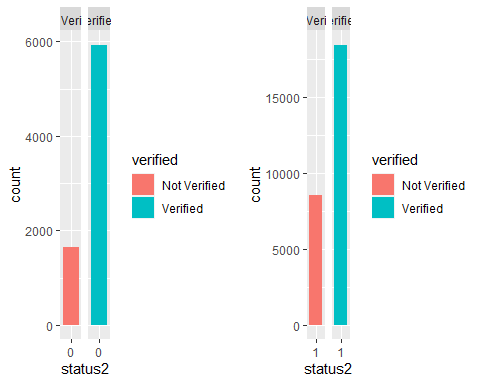


LoansDT$loanID <- NULL

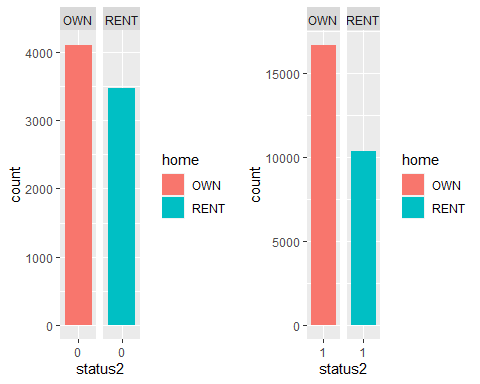
Then I divided up my modified data frame into good and bad loans based on the status2 variable. Then I made bar graphs of some of the categorical variables divided by good and bad loans to see if there were any obvious relationships. I plotted the ‘term’, ‘verified’ and home variable against status2 For the term variable ‘bad’ loans have a greater percentage of loans with 60 month terms than the ‘good’ loans. It appears that there might be some relationship between the ‘verified’ and term variables and status. I will keep this variables for part 5.

loans\_bad <- sqldf("Select \* from LoansDT where status2 = 0")  
loans\_good <- sqldf("Select \* from LoansDT where status2 = 1")  
  
   
p1 <- ggplot(loans\_bad, aes(status2, fill = term)) + facet\_wrap(~term) + geom\_bar()  
p2 <- ggplot(loans\_good, aes(status2, fill = term)) + facet\_wrap(~term) + geom\_bar()  
   
  
p3 <- ggplot(loans\_bad, aes(status2, fill = verified)) + facet\_wrap(~verified) + geom\_bar()  
p4 <- ggplot(loans\_good, aes(status2, fill = verified)) + facet\_wrap(~verified) + geom\_bar()  
   
  
p5 <- ggplot(loans\_bad, aes(status2, fill = home)) + facet\_wrap(~home) + geom\_bar()  
p6 <- ggplot(loans\_good, aes(status2, fill = home)) + facet\_wrap(~home) + geom\_bar()  


multiplot(p3,p4,cols=2)



multiplot(p5,p6,cols=2)

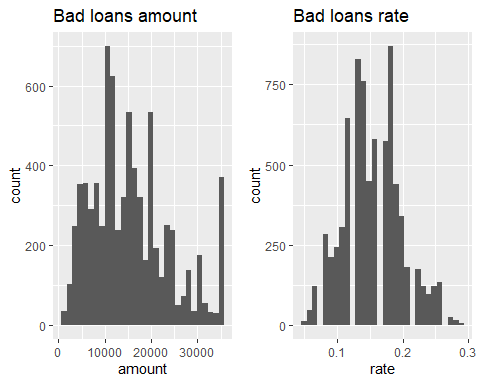


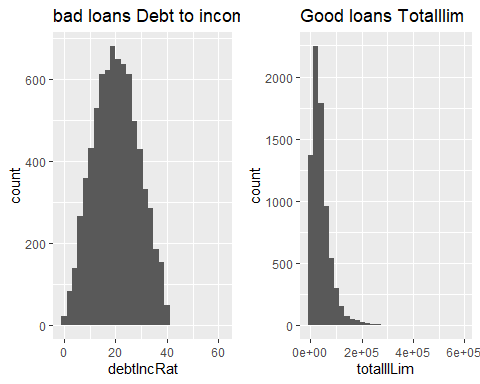
I plotted a histogram of all the variables in the LoansDT dataframe to get an overall sense of of the data is normally distributed or not. Due to space limitations, I did not include this plot. Much of the data looks right skewed.

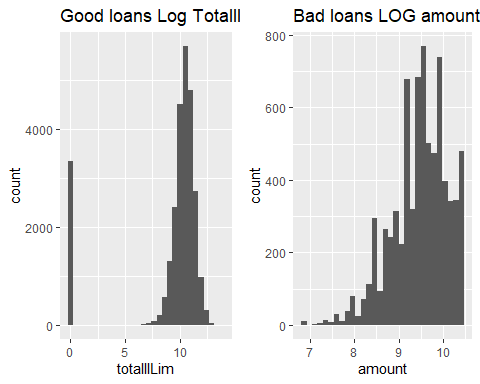
I plotted the states against status for both the good and bad loans but noticed any relationship. As expected the more populous states (New York and California) had the higher number of loans both good and bad. I will be dropping this variable. I am also going to drop the employment field as there are so many different values in there it will slow my regression down as there are over 15,000 distinct values, which is too many for the variable to be significant. Due to space concerns, I will not be displaying the plots.

I plotted variables from each dataframe (‘Good’ and ‘Bad’) to determine if the data is normally distributed or not. There is a definite right skew to many of the numeric variables. “rate”,“Amount”, “debt to income ratio” look normally distributed so I will not transform them. To transform the rest of the numeric variables, I created 2 new dataframes for the good and bad loans and with just the numeric fields IO am going to transform. Then I used the log1p function on the dataframes. I use this function instead of the log function because some of the variables contain zeroes and log has issues with zeroes. I will not transform the following variables as the look normal: “revolRatio” , “accOpen24” and “accOpen24”. I plot the data into a histogram again after the log and the data loos much more normally distributed. There are a few that now look left-skewed. Also after performing it the first time and looking for NA’s i found that it created a total to 384 NAs in bcratio between the 2 groups so I am removing that from the dataframes I apply the log1p to. The debtIncRat and rate fields were also normally distributed so I did not include them in the group of variables to apply the log1p to. I also did not transform “totalPaid” as we are not supposed to use that values as a predictor. I split the categorical variables into a separate dataframe to join with the logged dataframes later. I plotted several of the variables after the log was applied to the to check the results.

par(mfrow=c(4,2))  
  
plt1 <- ggplot(loans\_bad, aes(x=amount)) + geom\_histogram() + ggtitle("Bad loans amount")  
plt2 <- ggplot(loans\_bad, aes(x=rate)) + geom\_histogram() + ggtitle("Bad loans rate")  
plt3 <- ggplot(loans\_bad, aes(x=debtIncRat)) + geom\_histogram() + ggtitle("bad loans Debt to income ratio")  
plt4 <- ggplot(loans\_bad, aes(x=totalIlLim)) + geom\_histogram() + ggtitle("Good loans TotalIlim")  
   
  
nologvars <- subset(LoansDT,select=c(bcRatio,debtIncRat,rate,totalPaid,amount2))  
  
good\_loan\_numeric <- loans\_good [ c("amount" , "payment" , "income" , "openAcc" , "revolRatio" , "totalAcc" ,   
 "totalBal" , "totalRevLim" , "accOpen24" , "avgBal" ,"bcOpen" , "totalLim",   
 "totalRevBal" , "totalBcLim" , "totalIlLim")]  
  
bad\_loan\_numeric <- loans\_bad[ c("amount" , "payment" , "income" , "openAcc" , "revolRatio" , "totalAcc" ,   
 "totalBal" , "totalRevLim" , "accOpen24" , "avgBal" ,"bcOpen" , "totalLim",   
 "totalRevBal" , "totalBcLim" , "totalIlLim")]  
  
good\_loan\_discrete <- loans\_good[ c("grade", "length" ,"home" ,"verified","reason" ,  
 "status2","term" )]  
  
bad\_loan\_discrete <- loans\_bad[ c("grade","length" ,"home" ,"verified","reason",   
 "status2","term")]  
good\_loan\_log <- log1p(good\_loan\_numeric)  
bad\_loan\_log <- log1p(bad\_loan\_numeric)  
  
plt5 <- ggplot(good\_loan\_log, aes(x=totalIlLim)) + geom\_histogram() + ggtitle("Good loans Log TotalIlim")  
plt6 <- ggplot(bad\_loan\_log, aes(x=amount)) + geom\_histogram() + ggtitle("Bad loans LOG amount")  
  
multiplot(plt1,plt2,cols=2)



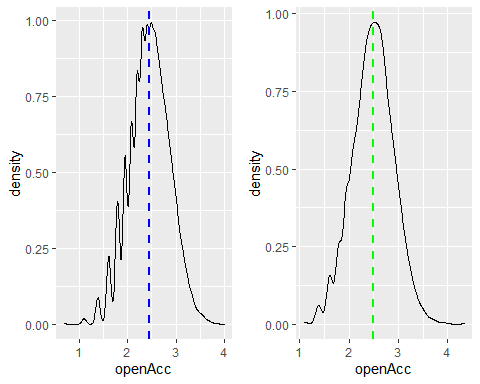


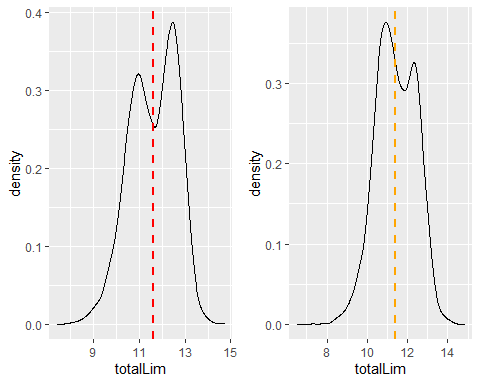


I checked the dataframes resulting from the log1p function for NA’s and found none. I am not showing it to conserve space.

Next I will do some density plots with some of the different logged variables too see if there is a noticeable difference between good and bad loans. I did density plots of openacc and totalLim for good and bad loans. I didn’t see much difference for openacc or totalLim between good or bad loans.I wasn’t able to eliminate many fields in my data exploration but I suspect once I start building my model, that is when I will eliminate more fields.

p1 <- ggplot(good\_loan\_log, aes(x=openAcc)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(openAcc)),  
 color="blue", linetype="dashed", size=1)  
  
p2 <- ggplot(bad\_loan\_log, aes(x=openAcc)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(openAcc)),  
 color="green", linetype="dashed", size=1)  
  
  
p3 <- ggplot(good\_loan\_log, aes(x=totalLim)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(totalLim)),  
 color="red", linetype="dashed", size=1)  
  
p4 <- ggplot(bad\_loan\_log, aes(x=totalLim)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(totalLim)),  
 color="orange", linetype="dashed", size=1)  
  
multiplot(p1,p2,cols=2)





rejoin\_df <-cbind(good\_loan\_discrete , good\_loan\_log)  
rejoin\_bad\_df <- cbind(bad\_loan\_discrete , bad\_loan\_log)   
rejoin\_full <- rbind(rejoin\_df,rejoin\_bad\_df)  
  
Loan\_regrsn <- cbind(rejoin\_full, nologvars)

PART 2 - Section 5 - The Logistic Model For the start of Section 5, I will create 2 datasets from the Loan\_regrsn dataframe from Step 4. One dataset will be my training dataset and will contain 80% of the data, the other will be my test dataset and will contain 20% of the data. I end up 6931 in Loan\_test and 27724 in Loan\_training

training\_size <- 0.8  
set.seed(2112)  
training\_rows <- sample(seq\_len(nrow(Loan\_regrsn)), size = floor(training\_size \* nrow(Loan\_regrsn)))  
Loan\_training <- Loan\_regrsn[training\_rows, ]  
Loan\_test <- Loan\_regrsn[-training\_rows, ]

Next I will run the full model and use the summary function to see which variables are significant and I should include in my model. Based upon the coefficients with significant p values I will keep the following values: grade,verified,reason,term,income,revolRatio,totalAcc,totalRevLim,accOpen24,bcOpen,totalRevBal,totalIlLim. When I run this model the p=values for the following fields are no longer significant:bcOpen,totalRevBal and totalIlLim, so I will drop then and try a third time. A strange thing happened When I ran the third model, The p-values was significant at .013 but it was higher than my second model which had a p-value of .012. The McFadden pseudo R Squared for the third model (.09414) was less than the second model (.09462) but just barely.Looking at the model only 3 of the 12 values of the reason variable were significant so I decided to drop the reason variable and try it again. The results were much better for the 4th time. The p-value is .0006 with a McFadden Pseudo R2 of .0932. This is the model I will move forward with. To save space, I will only show the p-value for the first and second models.

Loantrain\_one <-glm(status2~.,data=Loan\_training,family="binomial")  
summary(Loantrain\_one)

##   
## Call:  
## glm(formula = status2 ~ ., family = "binomial", data = Loan\_training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.8238 0.3054 0.5147 0.7300 1.5749   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.753e+00 2.391e+00 -0.733 0.463311

## McFadden   
## 0.1031769

Loantrain\_two <- glm(status2~grade+ verified + reason + term + income + revolRatio +totalAcc + totalRevLim + accOpen24 + bcOpen + totalRevBal +totalIlLim ,data=Loan\_training,family="binomial")  
summary(Loantrain\_two)

##   
## Call:  
## glm(formula = status2 ~ grade + verified + reason + term + income +   
## revolRatio + totalAcc + totalRevLim + accOpen24 + bcOpen +   
## totalRevBal + totalIlLim, family = "binomial", data = Loan\_training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7170 0.3205 0.5252 0.7339 1.5311   
##   
## McFadden   
## 0.0946262

Loantrain\_three <- glm(status2~grade+ verified + reason + term + income + revolRatio +totalAcc + totalRevLim + accOpen24 ,data=Loan\_training,family="binomial")  
summary(Loantrain\_three)

##   
## Call:  
## glm(formula = status2 ~ grade + verified + reason + term + income +   
## revolRatio + totalAcc + totalRevLim + accOpen24, family = "binomial",   
## data = Loan\_training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7009 0.3208 0.5260 0.7337 1.5221   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.01291 0.41188 -2.459 0.013924 \*   
## grade1 -0.49536 0.06968 -7.110 1.16e-12 \*\*\*  
## grade2 -0.97236 0.06951 -13.990 < 2e-16 \*\*\*  
## grade3 -1.25334 0.07478 -16.760 < 2e-16 \*\*\*  
## grade4 -1.39934 0.08307 -16.845 < 2e-16 \*\*\*  
## grade5 -1.64692 0.10577 -15.570 < 2e-16 \*\*\*  
## grade6 -1.74163 0.17503 -9.950 < 2e-16 \*\*\*  
## verifiedVerified -0.20631 0.03716 -5.552 2.83e-08 \*\*\*  
## reasoncredit\_card -0.33830 0.19587 -1.727 0.084137 .   
## reasondebt\_consolidation -0.29687 0.19341 -1.535 0.124797   
## reasonhome\_improvement -0.23341 0.20331 -1.148 0.250952   
## reasonhouse -0.15784 0.31684 -0.498 0.618354   
## reasonmajor\_purchase -0.33796 0.22512 -1.501 0.133297   
## reasonmedical -0.48817 0.23633 -2.066 0.038865 \*   
## reasonmoving -0.61920 0.25773 -2.403 0.016283 \*   
## reasonother -0.25935 0.20423 -1.270 0.204122   
## reasonrenewable\_energy -0.55135 0.52666 -1.047 0.295156   
## reasonsmall\_business -0.82574 0.23700 -3.484 0.000494 \*\*\*  
## reasonvacation -0.13705 0.27685 -0.495 0.620574   
## reasonwedding 7.40781 72.46313 0.102 0.918575   
## term60 months -0.65411 0.03797 -17.228 < 2e-16 \*\*\*  
## income 0.34619 0.03474 9.964 < 2e-16 \*\*\*  
## revolRatio -0.62515 0.10952 -5.708 1.14e-08 \*\*\*  
## totalAcc 0.10677 0.03872 2.758 0.005824 \*\*   
## totalRevLim 0.07121 0.02186 3.257 0.001124 \*\*   
## accOpen24 -0.46665 0.03259 -14.320 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 29029 on 27723 degrees of freedom  
## Residual deviance: 26296 on 27698 degrees of freedom  
## AIC: 26348  
##   
## Number of Fisher Scoring iterations: 8

r3 <- PseudoR2(Loantrain\_three)  
r3[1]

## McFadden   
## 0.0941462

Loantrain\_Four <- glm(status2~grade + verified + term + income + revolRatio +totalAcc + totalRevLim + accOpen24 ,data=Loan\_training,family="binomial")  
summary(Loantrain\_Four)

##   
## Call:  
## glm(formula = status2 ~ grade + verified + term + income + revolRatio +   
## totalAcc + totalRevLim + accOpen24, family = "binomial",   
## data = Loan\_training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6806 0.3228 0.5263 0.7347 1.4497   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.25375 0.36781 -3.409 0.000653 \*\*\*  
## grade1 -0.49820 0.06946 -7.172 7.37e-13 \*\*\*  
## grade2 -0.97831 0.06865 -14.250 < 2e-16 \*\*\*  
## grade3 -1.26149 0.07319 -17.235 < 2e-16 \*\*\*  
## grade4 -1.41071 0.08100 -17.417 < 2e-16 \*\*\*  
## grade5 -1.66973 0.10347 -16.137 < 2e-16 \*\*\*  
## grade6 -1.76051 0.17298 -10.177 < 2e-16 \*\*\*  
## verifiedVerified -0.20589 0.03710 -5.550 2.86e-08 \*\*\*  
## term60 months -0.64593 0.03737 -17.286 < 2e-16 \*\*\*  
## income 0.34222 0.03426 9.990 < 2e-16 \*\*\*  
## revolRatio -0.62001 0.10649 -5.822 5.81e-09 \*\*\*  
## totalAcc 0.10785 0.03864 2.791 0.005248 \*\*   
## totalRevLim 0.06733 0.02164 3.112 0.001859 \*\*   
## accOpen24 -0.45874 0.03239 -14.165 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 29029 on 27723 degrees of freedom  
## Residual deviance: 26322 on 27710 degrees of freedom  
## AIC: 26350  
##   
## Number of Fisher Scoring iterations: 5

r4 <- PseudoR2(Loantrain\_Four)  
r4[1]

## McFadden   
## 0.09326458

The 95 percent confidence interval of the predictor values are below:

confint(Loantrain\_Four)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -1.97540259 -0.5335635  
## grade1 -0.63560788 -0.3632417  
## grade2 -1.11425564 -0.8450565  
## grade3 -1.40613865 -1.1191590  
## grade4 -1.57038712 -1.2528217  
## grade5 -1.87307640 -1.4673893  
## grade6 -2.10104448 -1.4221381  
## verifiedVerified -0.27881973 -0.1333908  
## term60 months -0.71916563 -0.5726795  
## income 0.27519524 0.4094780  
## revolRatio -0.82915536 -0.4116982  
## totalAcc 0.03210950 0.1835677  
## totalRevLim 0.02479834 0.1096204  
## accOpen24 -0.52241879 -0.3954641

Below I will use my model created above to predict the status for loans in the test data. After I use my model to predict the status of the loans, I will create a confusion matrix to determine the overall accuracy of the model. The results of the confusion matrix for the model are as follows: I have 5260 loans correctly predicted as good loans and 154 loans correctly predicted as bad loans. So I have a total accuracy percentage of 78%. 98 percent of good loans were correctly predicted as good while only 11 percent of bad loans were correctly predicted as bad. While this model leaves a bit to be desired in predicting the bad loans accurately, it does a much better job accurately predicting the good ones.

test\_mdl <- glm(status2~grade + verified + term + income + revolRatio +totalAcc + totalRevLim + accOpen24 ,data=Loan\_test,family="binomial")  
probabilities <- predict(test\_mdl,newdata=Loan\_test, type="response")  
length(probabilities)

## [1] 6931

t <- .5  
Bad\_or\_Good <- ifelse(probabilities>t,1,0)  
Bad\_or\_Good <- as.factor(Bad\_or\_Good)  
cf <- confusionMatrix(data=Bad\_or\_Good,reference=Loan\_test$status2)  
cf$table

## Reference  
## Prediction 0 1  
## 0 154 116  
## 1 1401 5260

cf$overall[1]

## Accuracy   
## 0.7811283

Section 6 - Optimizing the Threshold for Accuracy.

In this section I am going to vary the threshold from .5 to attempt to correctly predict more bad loans. I will create a confusion matrix for each different threshold and then I will graph them all to show accuracy vs. threshold. As you recall from the previous section my accuracy percentage was 78% but was a much better predictor of ‘good’ loans than ‘bad’ loans. I will calculate the accuracy for 5 different thresholds between 0 and 1 to attempt to predict a better percentage of bad loans. Due to space constraints, I will just show the confusion matrix containing the predictions and the accuracy value for each calculation.

# 6 different thresholds  
th1 <- .30  
th2 <- .45  
th3 <- .55  
th4 <- .60  
th5 <- .75   
## Reference  
## Prediction 0 1  
## 0 1 0  
## 1 1554 5376

c1$overall[1]

## Accuracy   
## 0.7757899

## Prediction 0 1  
## 0 72 35  
## 1 1483 5341

c2$overall[1]

## Accuracy   
## 0.780984

## Reference  
## Prediction 0 1  
## 0 262 277  
## 1 1293 5099

c3$overall[1]

## Accuracy   
## 0.7734815

## Reference  
## Prediction 0 1  
## 0 409 469  
## 1 1146 4907

c4$overall[1]

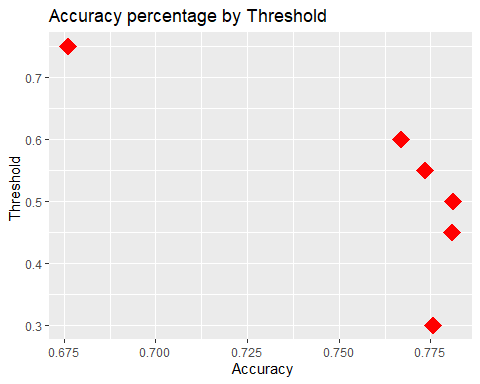
## Accuracy   
## 0.7669889

Bad\_or\_Good5 <- ifelse(probabilities>th5,1,0)  
Bad\_or\_Good5 <- as.factor(Bad\_or\_Good5)  
c5 <- confusionMatrix(data=Bad\_or\_Good5,reference=Loan\_test$status2)  
c5$table

## Reference  
## Prediction 0 1  
## 0 958 1647  
## 1 597 3729

c5$overall[1]

## Accuracy   
## 0.6762372

Given the data in the confusion matrices above, I determine the accuracy vs the threshold of each of them - including the default (.5). So I did a confusion matrix for 5 thresholds in addition to the default of .5. The other values are:.30,.45,.55,60,.75,.85. The highest accuracy level I get is at the .45 and the default threshold. My accuracy percentages is 78.11% for the default threshold and 78.10 for the .45 threshold. The .45 threshold does a little better job at predicting ‘good’ loans with 5,341 correctly predicted as ‘good’ compared to 5,260 for the default threshold, but the .5 threshold does a better job at predicting ‘bad’ loans with 116 ‘bad’ loans predicted correctly. That number is 35 for the .45 threshold. The lower the threshold is ,the better it is at predicting good loans. Unfortunately, the lower the threshold, the worse it is a predicting ‘bad’ loans. The lowest threshold I used, .3, is 77.5% accurate with 5376 out of 5377 loans correctly predicted as good but it incorrectly predicts all ‘bad’ loans as good. As the threshold increases from the default level, the overall accuracy drops as the number of ‘good’ loans predicted correctly drops but the number of ‘bad’ loans predicted correctly increases. for the .55 threshold, the overall accuracy drops slightly to 77.3%. The number of correctly predicted ‘good’ loans drops to 4907 but the number of ‘bad’ loans rises to 469. As the threshold values rises, the number of predicted ‘good’ loans drops and the number of predicted ‘bad’ loans rises. With the threshold set at .65 the number of ‘good’ loans is 4907 while the number of bad loans is 409. From the .65 threshold to the .75 threshold the numbers take a dramatic jump. At the .785 threshold there are 3729 loans that are ‘good’ and 958 ‘bad’ loans. This leaves us with over 2,244 loans that are incorrectly classified as good or bad. The reverse is true as well since as we decrease the threshold from .5 the number of loans classified as good rises and the number of loans classified as bad drops. The accuracy percentage drops as well.  Part 7 - Optimizing the Threshold for Profit In this section I will take the predictions at the different threshold levels from Part 6 and apply them to my test data. Then I will sum the profit for each threshold and determine what the maximum profit increase is for the loans I predicted as good. I add the probabilities calculations from section 5 and the test data I created in section 5. Then I sum the total paid minus the amount for my total profit. According to the calculations, the threshold of .45 is where the biggest profit is - 1,446,039. The lowest profit is my second to last threshold of.65 where the profit is 955,800.1. The second highest profit among my thresholds is at .3 where the profit is 1,418,208. The default threshold profit (.5) is the third highest at 1,414,100. From there the threshold for .55 is 1,233,294 and then for .75 the profit is at 1,066,199.

The total profit for all the loans in the test set is 1,418,710. Compared to not using any model the maximum percentage increase that can be expected by deploying my model is 2 percent or $27,329 The profit for a ‘perfect’ model that denies all of the truly bad loans is 750,778.40. The increase in profit from a perfect model to my model is 90 percent. For my best profit threshold, the overall accuracy of correctly predicted bad and good loans is 78.09 percent. This does not quite match up with the maximum accuracy threshold but it is pretty close. The maximum accuracy threshold is 78.11 percent and the accuracy percentage for the top profit is 78.10. For the maximum accuracy threshold there are 5,341 correctly predicted good loans and 72 correctly predicted bad loans. So that is 5,413 correct loans out of 6,931.

Loan\_test\_th1 <- cbind(Loan\_test,probabilities)  
  
thr1 <- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .30"))  
thr2<- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .45 "))  
thrdf <- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .50 "))  
thr3<- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .55 "))  
thr4<- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .65 "))  
thr5<- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where probabilities >= .75 "))  
Prof\_Perfect\_model <- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1 where status2 = 1 "))  
Profit\_No\_Model <- sqldf(c("select sum(totalPaid - amount2) from Loan\_test\_th1"))   
  
Profit\_Margins<- list("profit for threshold of .3",thr1$`sum(totalPaid - amount2)`,"profit for threshold of .45",thr2$`sum(totalPaid - amount2)`,"profit for threshold of .5 - default",thrdf$`sum(totalPaid - amount2)`,"profit for threshold of .55",thr3$`sum(totalPaid - amount2)`,"profit for threshold of .65",thr4$`sum(totalPaid - amount2)`,"profit for threshold of .75",thr5$`sum(totalPaid - amount2)`)  
  
test\_profits <- list("Profit for Perfect Model",Prof\_Perfect\_model$`sum(totalPaid - amount2)`,"Profit For No Model",Profit\_No\_Model$`sum(totalPaid - amount2)`)  
print(Profit\_Margins)

## [1] "profit for threshold of .3"  
## [1] 1418208  
## [1] "profit for threshold of .45"  
## [1] 1446039  
## [1] "profit for threshold of .5 - default"  
## [1] 1414100  
## [1] "profit for threshold of .55"  
## [1] 1233294  
## [1] "profit for threshold of .65"  
## [1] 955800.1  
## [1] "profit for threshold of .75"  
## [1] 1066199

print(test\_profits)

## [1] "Profit for Perfect Model"  
## [1] 750778.4  
  
## [1] "Profit For No Model"  
## [1] 1418710

model\_2\_profit <- thr2$`sum(totalPaid - amount2)`  
no\_model\_profit <- Profit\_No\_Model$`sum(totalPaid - amount2)`  
perfect\_model\_profit <- Prof\_Perfect\_model$`sum(totalPaid - amount2)`  
  
profit\_pct\_No\_model <- (model\_2\_profit - no\_model\_profit)/no\_model\_profit  
## [1] 0.01926338

profit\_perfect\_model\_pct

## [1] 0.9260528

Section 8 - Results Summary

My final classification model ended up with 8 predictor variables: grade, verified ,term, income, revolRatio, totalAcc, totalRevLim and accOpen24. The p-value of my final model is .0006. My final model results in a 2 percent increase in profit, or $27,329 for the bank over their current process. If the bank used a perfect model and denied all the loans in the dataset that are marked as bad, it would result in an 92 percent increase in profit or 695,260.60. The final model had a threshold of .45 and correctly predicted 5,413 out of 6,931 loans correctly. That is 78.1 percent. My model correctly predicted 5,341 out of 5,376 or 99.3 percent ‘good loans’. It correctly predicted 72 out of 1,555 ‘bad’ loans’ or 5 percent. The final model is extremely accurate in predicting good loans.