Project Final

Ryan Hass

4/27/2021

## Section 1: Executive Summary

This report provides the analysis and evaluation of a model used to predict if a loan applicant will default on a loan.

The methods of analysis include a logistic regression model that was derived from the most relevant variables within the given data set including income, employment status, interest rate, salary, loan term, and overall applicant grade. These categories are visualized below against both good and bad loan applicants. Good and bad applicants were decided based on current loan status. A training data set was created using known information and a secondary data set was tested in order to determine the maximum accuracy of the model. From there, the model was altered to maximize profitability.

All calculations and comparisons can be found in the report below.

The results show that the model optimized for accuracy correctly predicted loan status 57% of the time, while classifying good loans as good 51.0% of the time and bad loans as bad 79.8% of the time.

The results of the model optimized for profitability correctly predict loan status with a percentage of 60%, while classifying good loans as good 56.3% of the time and bad loans as bad 73.4% of the time.

The report finds the models described above should be considered given the understanding that you can increase accuracy or profitability at the expense of either missing good loans or accepting bad loans.

Recommendations include:

* Reevaluate the variables in the given data to determine if there are any other significant predictors.
* Evaluate the comparison between missing good loans or accepting bad loans in respect to profit and/or accuracy.
* Broad categories of loans decided as ‘good’ or ‘bad’ may be reconsidered.

## Section 2: Introduction

This data set includes 30 variables for 50,000 loans. This project is designed to evaluate the variables and create a model that will be used to predict whether a loan applicant is likely to default on their loan.

Thorough graphical evaluation of the variables was completed to determine which variables impacted whether a loan was considered ‘bad’ or ‘good’. ‘Bad’ and ‘good’ classifications are designated based on the current status of the loan.

We acknowledge that some of the variables (loanID, state, employment, verified) are not analyzed for the purpose of identifying the likelihood of defaulting on a loan based on ethics, redundancy, or irrelevance. Additionally, some variables (rate, employment) were modified to allow for more accurate evaluation and/or simpler graphical evaluation.

Once our variable are confirmed, we will use logistic regression to create our predictive model.

## Section 3: Preparing and Cleaning the Data

Loading in the data and necessary packages

library(dplyr)  
library(readr)  
library(ggplot2)  
library(gridExtra)  
library(dlookr)  
library(pROC)  
  
  
loans <- read\_csv("C:/Users/rhass/OneDrive - Green Bay Packaging/Personal/Data Science/DS 705 - Statistical Methods/Project/loans50k.csv", col\_types = cols())

Below are the steps for setting up the response variable and designating loans as ‘good’ or ‘bad’ based on status. Additionally, I cleaned the data by identifying and removing NA’s in status variable. Due to there only being a single NA, I simply removed the row, leaving 34655 rows to evaluate.

Creating the response variable was directed by the project instructions. Some statuses were considered relevant, and some were irrelevant which is reflected below.

# Removing all unnecessary statuses  
  
loans <- loans[!(loans$status == "Late (31-120 days)"),]   
loans <- loans[!(loans$status == "Late (16-30 days)"),]  
loans <- loans[!(loans$status == "Current"),]  
loans <- loans[!(loans$status == "In Grace Period"),]  
  
preTab <- table(is.na(loans$status))  
preTab

##   
## FALSE TRUE   
## 34655 1

# Removing the single NA from statuses  
  
loans <- loans[!is.na(loans$status),]  
  
postTab <- table(is.na(loans$status))  
postTab

##   
## FALSE   
## 34655

loans <- loans %>%  
 mutate(myStatus = case\_when(status %in% c("Fully Paid") ~ "1", status %in% c("Default", "Charged Off") ~ "0"))  
  
loans <- loans %>%  
 mutate(myStatus = as.factor(myStatus))  
  
loans <- loans %>%  
 mutate(myTerm = case\_when(term %in% c("36 months") ~ "1", term %in% c("60 months") ~ "0"))  
  
loans <- loans %>%  
 mutate(grade = as.factor(grade))  
  
loans <- loans %>%  
 mutate(myTerm = as.factor(myTerm))

The next section demonstrates the evaluation of whether or not to remove other NAs from the data set. I chose to mutate the data in ‘employment’ to create a new variable ‘myEmploy’ that signifies if the applicant is employed or not. The assumption that a missing value in the ‘employement’ variable indicates unemployment was made. I chose to ignore the other NAs from he data set based on the low number remaining, knowing that R will ignore them when creating my model.

# Removing other NA's from data set  
  
loans <- loans %>%  
 mutate(myEmploy = case\_when(is.na(employment) ~ "Unemployed", !is.na(employment) ~ "Employed"))  
  
loans <- loans %>%  
 mutate(myEmploy = as.factor(myEmploy))  
  
table(is.na(loans))

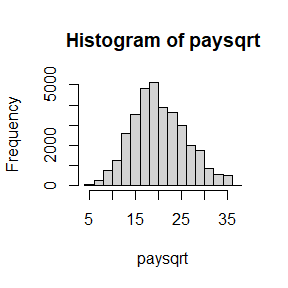
##   
## FALSE TRUE   
## 1210248 2677

## Section 4: Exploring and Transforming the Data

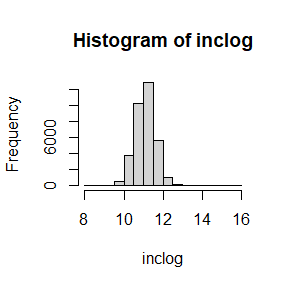
The next step includes creating graphs and visualizations to evaluate the relationship between predictors and loan status. Some data transformations were also conducted to compare against the response variable.

Displayed below, the ‘payment’ and ‘income’ variables were transformed to normalize the data.

n1 <- hist(paysqrt)



loans$myPay <- transform(loans$payment, method = "sqrt")  
  
n2 <- hist(inclog)

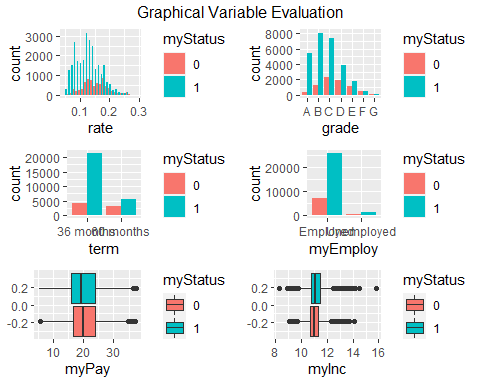


loans$myInc <- transform(loans$income, method = "log")

I’ve graphed 4 instances where good and bad loans appear to have different distributions in respect the the variables they are plotted against. Two quantitative variable were plotted, with one being transformed to determine the graphical relationship between the variable and loan status.

The first plot suggests that there is a difference between rate and its relationship between good and bad loans. The second plot, visualizing grade, also shows slightly different proportions between the dependent and independent variables being compared. The third and fourth plots summarize data in to ‘short’ and ‘long’ term (based on months) loans as well as ‘employed’ and ‘unemployed’ applicants. The final two plots help compare transformed income and payment data against loan status.

p1 <- ggplot(loans, aes(x=rate, fill = myStatus))+  
 geom\_histogram(binwidth = 0.01, position = "dodge")  
  
p2 <- ggplot(loans, aes(x=grade, fill = myStatus))+  
 geom\_bar(position = "dodge")  
  
p3 <- ggplot(loans, aes(x=term, fill = myStatus))+  
 geom\_bar(position = "dodge")  
  
p4 <- ggplot(loans, aes(x=myEmploy, fill = myStatus))+  
 geom\_bar(position = "dodge")  
  
p5 <- ggplot(loans, aes(x=myPay, fill = myStatus))+  
 geom\_boxplot()  
  
p6 <- ggplot(loans, aes(x=myInc, fill = myStatus))+  
 geom\_boxplot()  
  
grid.arrange(p1,p2,p3,p4,p5,p6, nrow=3, top="Graphical Variable Evaluation")



## Section 5: The Logistic Model

In this section, I created two, randomly selected data sets from the cleaned and prepared data from above.

The data in the ‘training’ set was used to create my logistic regression model based on the variables that were determined to have a relationship with loan status. That model was then compared against the ‘test’ data set in order to validate the performance.

A contingency table was generated to determine the overall accuracy of the model (the percentage of correctly predicted outcomes), the percentage of actually good loans that are predicted as good, and the percentage of actually bad loans that are predicted as bad.

smp\_size <- floor(0.80\*nrow(loans))  
  
set.seed(12345)  
  
train\_ind <- sample(seq\_len(nrow(loans)), size = smp\_size)  
  
train <- loans[train\_ind, ]  
test <- loans[-train\_ind, ]  
  
train$totalPaid <- NULL  
train$employment <- NULL  
  
model <- glm(myStatus ~ rate + grade + myTerm + myEmploy + myPay + myInc, data = train, family = "binomial")  
  
predProb <- predict(model, test, type = "response")  
threshold <- 0.5  
  
predStatus <- cut(predProb, breaks=c(-Inf, threshold, Inf),   
 labels=c("Bad", "Good"))  
  
cTab <- table(test$myStatus, predStatus)   
addmargins(cTab)

## predStatus  
## Bad Good Sum  
## 0 85 1372 1457  
## 1 71 5403 5474  
## Sum 156 6775 6931

p <- round(sum(diag(cTab)) / sum(cTab), 4)  
q <- round((5403/5474) \* 100, 3)  
r <- round((85/1457) \* 100, 3)

## [1] "Proportion correctly predicted = 0.7918"

## [1] "Percentage of actually good loans that are predicted as good = 98.703 %"

## [1] "Percentage of actually bad loans that are predicted as bad = 5.834 %"

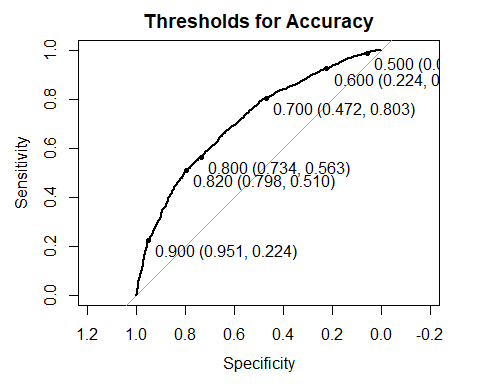
The performance of the model, and its validation against the ‘test’ data set indicate that the model is effective for predicting if a loan will be repaid, 79% of the time at the set threshold as defined above.

## Section 6: Optimizing the Threshold for Accuracy

An analysis of the model was conducted to optimize the threshold for accuracy.

As the threshold varies from 0 to 1, the overall accuracy and proportions of correctly predicted good and bad loans varies. The threshold that results in the best overall accuracy is 0.82 as measured by the area under the curve. As the threshold moves closer to 0, the sensitivity of the model increases, but this is at the expense of the specificity. This means that as the threshold decreases, the risk of classifying a bad loan as good increases. The contrary is true in the other direction, indicating that the risk of classifying a good loan as bad increases. If bad loans are classified as good, the likelihood of repayment is low, costing the bank money. If good loans are classified as bad, the bank will lose profits.

my\_roc <- roc(test$myStatus, predProb, plot = TRUE,main = "Thresholds for Accuracy", print.thres= c(0.5, 0.6, 0.7, 0.8, 0.82, 0.9))



\*Point labels indicate the Specificity (X-Axis) and Sensitivity (Y-Axis) given a prescribed Threshold values.

maxThresh <- coords(my\_roc, "best", ret = "threshold")  
AUC <- auc(test$myStatus, predProb)  
  
print(paste('AUC = ', round(AUC, 3)))

## [1] "AUC = 0.709"

print(paste('Maximum Threshold = ', round(maxThresh, 3)))

## [1] "Maximum Threshold = 0.82"

maximum <- 0.82  
  
predStatusMax <- cut(predProb, breaks=c(-Inf, maximum, Inf),   
 labels=c("Bad", "Good"))  
  
cTabMax <- table(test$myStatus, predStatusMax)   
addmargins(cTabMax)

## predStatusMax  
## Bad Good Sum  
## 0 1162 295 1457  
## 1 2684 2790 5474  
## Sum 3846 3085 6931

pMax <- round(sum(diag(cTabMax)) / sum(cTabMax), 4)  
qMax <- round((2790/5474) \* 100, 3)  
rMax <- round((1162/1457) \* 100, 3)

## [1] "Proportion correctly predicted = 0.5702"

## [1] "Percentage of actually good loans that are predicted as good = 50.968 %"

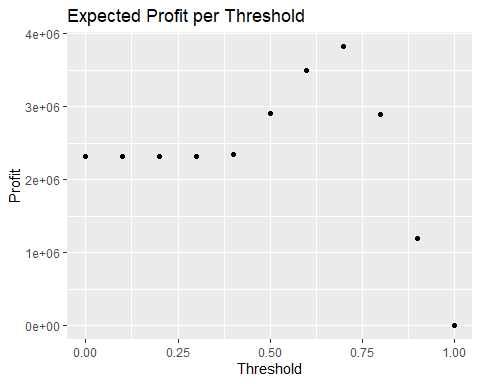
## [1] "Percentage of actually bad loans that are predicted as bad = 79.753 %"

The performance of the model, and its validation against the ‘test’ data set indicate that the model is effective for predicting if a loan will be repaid, 57% of the time at the maximum threshold as defined above.

## Section 7: Optimizing the Threshold for Profit

A similar analysis was done to optimize the threshold for profit as opposed to accuracy. In this case, model is optimized at a threshold of 0.8 in terms of maximizing profit. Similarly to the analysis of accuracy, the further the threshold gets from 0.8, the lower profitability becomes. Thresholds 0.1 - 0.4 stay the same at 2312893, and increase to a maximum of 3826964 at 0.8 before decreasing to 1193352 at 1.0.

test <- test %>%   
 mutate(profit = totalPaid-amount)  
  
test$predprob = predict(model, test, type="response")  
  
  
t0 <- test %>%  
 filter(predprob > 0.0) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t1 <- test %>%  
 filter(predprob > 0.1) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t2 <- test %>%  
 filter(predprob > 0.2) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t3 <- test %>%  
 filter(predprob > 0.3) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t4 <- test %>%  
 filter(predprob > 0.4) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t5 <- test %>%  
 filter(predprob > 0.5) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t6 <- test %>%  
 filter(predprob > 0.6) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t7 <- test %>%  
 filter(predprob > 0.7) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t8 <- test %>%  
 filter(predprob > 0.8) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t9 <- test %>%  
 filter(predprob > 0.9) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
t10 <- test %>%  
 filter(predprob > 1.0) %>%  
 dplyr::summarize(total\_profit = sum(profit))  
  
  
profitData <- matrix(c(2312893, 2312893, 2312893, 2312893, 2347286, 2904028, 3504950, 3826964,2891711, 1193352, 0))  
tVals <- matrix(c(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0))  
  
  
qplot(tVals, profitData) + labs(x="Threshold", y="Profit", title="Expected Profit per Threshold")



Compared to not using the model, the maximum percentage increase in profit that can be expected by deploying the model is 60%.

This increase in profit compares to the increase in profit from a perfect model that denies all of the truly bad loans by 73%

The overall percentage of correctly predicted loans based on profit = 59.9%. The percentage of actually good loans that are predicted as good = 56.3%. The percentage of actually bad loans that are predicted as bad = 73.4%.

The maximum profit threshold (0.80) does in fact coincide with the maximum accuracy threshold (0.82).

## predStatus2  
## Bad Good Sum  
## 0 1070 387 1457  
## 1 2392 3082 5474  
## Sum 3462 3469 6931

## [1] "Proportion correctly predicted = 0.599"

## [1] "Percentage of actually good loans that are predicted as good = 56.303 %"

## [1] "Percentage of actually bad loans that are predicted as bad = 73.439 %"

## Section 8: Results Summary

Overall, the logistic regression model created to predict if a loan applicant is likely to default on their loan is highly effective, accurately predicting loan status 57% of the time.

The final classification threshold for loan status is equal to 0.82 for accuracy based on the maximized AUC and 0.80 for profit, which closely coincide with one another.

The model optimized for accuracy correctly predicts loan status 57% of the time, while classifying good loans as good 51.0% of the time and bad loans as bad 79.8% of the time.

The model optimized for profitability correctly predicts loan status with a percentage of 60%, while classifying good loans as good 56.3% of the time and bad loans as bad 73.4% of the time.