Risk Assessment for Bank Loans

Glenn NIles

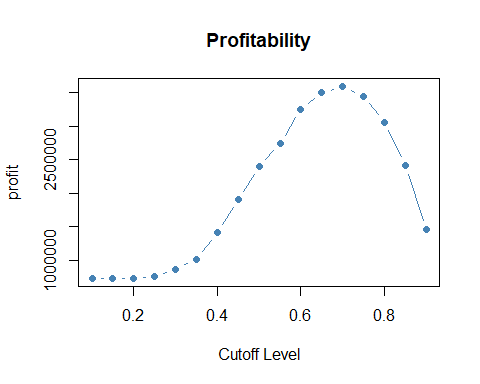
February 24, 2019

## Section 1: Executive Summary

This report provieds an analysis of the evaluation of the banks approach to accepting loans and how to increase the likelihood of giving loans that will have a better chance of being paid back as well as increasing the bank’s profits. Methods of analysis include logistic regression based on loans given out by the bank. The model was tested by accuracy and profitability. All data and computations are included in the rest of the report. Results of the data show that the bank could substantially increase their profitability in by using the approach used here to determine which factors indicate an applicant will successfully pay back their loan. Following the model used to maximize profitability we can see a $3.6 million increase in profits. While an approach that creates a more accurate model for giving out loans that will be successfully paid back it is substantially lower than the profitability model. A profit of $2.4 million is still returned by the model. This would involve giving out less risky loans in return for a lower profit.

It is recommended that the model for maximized profitability is used in the future. Accepting more risky loans is still more profitable than maximizing the amount of good loans given out.

It should be noted that this model is only 73% accurate for the maximized profitability model and 79% accurate for the maximized amount of good loans. While this is very good it is recommended that more data continue to be taken in order to continue to improve the model in order to maximize not only profitability but the likelihood that bad loans will not be given out.



## Warning: package 'gridExtra' was built under R version 3.5.2

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Load Data

loans <- read.csv("loans50k.csv")

## Section 2: Introduction

A .csv file is provided containing 50000 loan applicants, the goal is to predict which of them are likely to default on their loans using logistic regression. The dataset contains 32 variables some of which we can use as predictor variables. The motivation behind this is to help the bank make sounder decisions and have more information to work with when issuing loans. The objective here is to predict the likelihood of default and ass the model’s ability to predict well. We will begin with some data cleaning and exploratory analysis to determine which variables could be useful as predictor variables.

## Section 3: Preparing and Cleaning the Data

A new column was made called ‘Standing’ that was used to categorize loan status’ into simply ‘Good’ and ‘Bad.’ Loans that were current, in a grace period or less than 120 days late were removed from the dataset as they aren’t relevant for this analysis.

The column ‘loanID’ was removed as it has nothing to do with the loan outside of being an identifier.

‘totalPaid’ was removed as it can’t be used as a predictor variable since it can only be determined after a loan is issued

‘state’ was removed since it doesn’t seem like geography, at least at a state level, is an indicator of loan repayment. It was clear that the amount of loans per state was based on population. States with higher populations, in general, had more loans than states with lower populations.

‘employment’ was removed since there were so many different jobs listed that it would be difficult to break them down into groups that would be useful for analysis.

This left me with 29 variables to work with.

##   
## car credit\_card   
## 1 281 7843   
## debt\_consolidation home\_improvement house   
## 21046 2021 126   
## major\_purchase medical moving   
## 619 378 215   
## other renewable\_energy small\_business   
## 1571 27 317   
## vacation wedding   
## 209 2

If the reason for taking the loan was for renewable energy or weddings the data was added to the ‘other’ value since there weren’t enough occurences of either of those to be able to use in analysis.

## [1] 782

## amount term rate payment grade length   
## 1 0 1 1 0 0   
## home income verified reason debtIncRat delinq2yr   
## 0 1 0 0 1 1   
## inq6mth openAcc pubRec revolRatio totalAcc totalPaid   
## 1 1 1 16 1 1   
## totalBal totalRevLim accOpen24 avgBal bcOpen bcRatio   
## 1 1 1 1 361 385   
## totalLim totalRevBal totalBcLim totalIlLim standing   
## 1 1 1 1 1

Seeing as there is 782 NA values in a dataset of 34656 rows, most of these are in bcOpen and bcRatio. I’m going to make the assumption that bcRatio will be similar to revolRatio and bcOpen will be similar to totalBal and totalRevLim. Several of the rows that have missing values have multiple missing values in that row. Because of this I feel comfortable removing all of these rows and will leave me with 34271 rows left in the dataset.

## Section 4: Exploring and Transforming the data

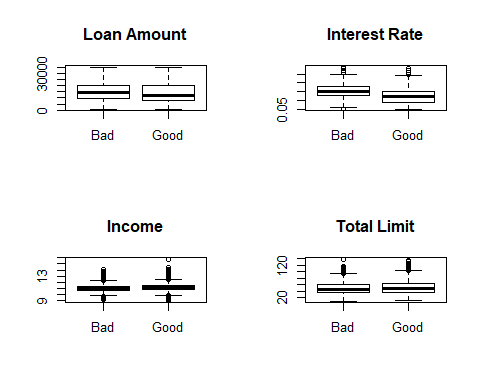
Many of the variables in this dataset are highly skewed. In an attempt to reduce the amount of extreme values and to give the variables a more normal shape the variables were transformed. Through trial and error and examining the histograms I either took the square root, the cube root or the natural logarithm of the variables.

The variables, ‘payment’ and ‘totalAcc’ were transformed by the square root.

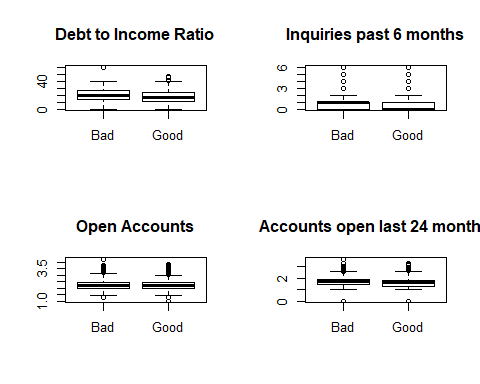
The variables, ‘openAcc,’ ‘totalBal,’ ‘accopen24,’ ‘avgBal,’ ‘bcOpen,’ ‘totalLim,’ ‘totalRevBal,’ and ‘totalILim’ were transformed by the cube root.

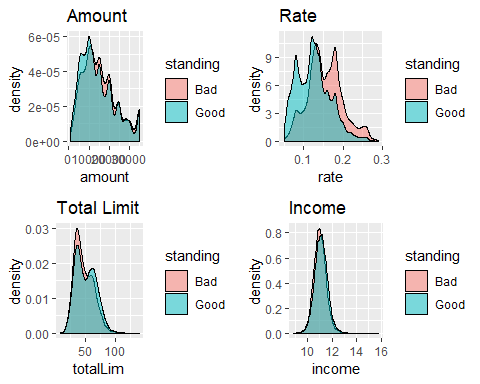
The natural logarithm was used to transform the variables ‘income’

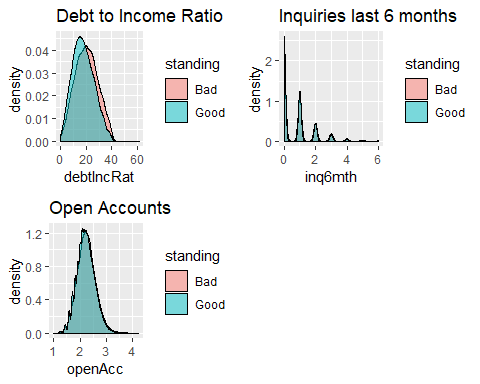
# Print side-by-side boxplots comparing good to bad loans  
par(mfrow=c(2,2))  
boxplot(cleanedLoan$amount~cleanedLoan$standing, main='Loan Amount')  
boxplot(cleanedLoan$rate~cleanedLoan$standing, main='Interest Rate')  
boxplot(cleanedLoan$income~cleanedLoan$standing, main='Income')  
boxplot(cleanedLoan$totalLim~cleanedLoan$standing, main='Total Limit')

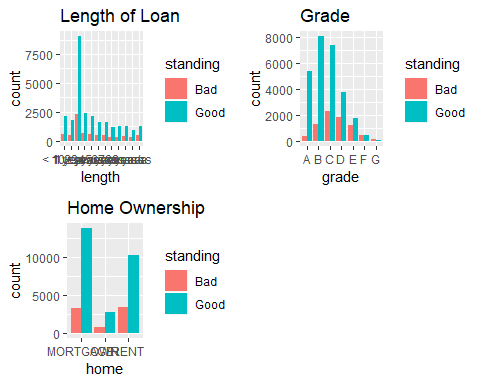


# More boxplots  
par(mfrow=c(2,2))  
boxplot(cleanedLoan$debtIncRat~cleanedLoan$standing, main='Debt to Income Ratio')  
boxplot(cleanedLoan$inq6mth~cleanedLoan$standing, main='Inquiries past 6 months')  
boxplot(cleanedLoan$openAcc~cleanedLoan$standing, main='Open Accounts')  
boxplot(cleanedLoan$accOpen24~cleanedLoan$standing, main='Accounts open last 24 months')









It looks to me like there may be several variables that may be indicative of what might help determine a good loan from a bad loan. Interest rate seems to be the one that sticks out the most but since people with better credit get better interest rates I’m wondering if the interest rate might be dependent on some of the other factors here.

The size of the loan and a person’s income seem to play a large part in determining loan standing.

Total limit is interesting to me because there seems to be a cutoff where lower limits have more bad loans and higher limits having more good standings.

The debt to income ration seems to play a role as well. This makes sense to me.

In the categorical variables it seems like the credit grade is a strong indicator of loan standing.

Its hard to tell and I want to explore it more, but home ownership may play a role in the standing of a loan.

## Section 5: The Logistic Model

The cleaned data will be split into two datasets. 80% of cleaned data will be turned into training data while the remaining 20% will be used as testing data.

# logistic regression model  
loanReg <- glm(trainingLoan$standing~., data=trainingLoan, family='binomial')  
summary(loanReg)

##   
## Call:  
## glm(formula = trainingLoan$standing ~ ., family = "binomial",   
## data = trainingLoan)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7850 0.3058 0.5136 0.7280 1.6716   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.243e+00 6.596e-01 4.916 8.81e-07 \*\*\*  
## amount 3.603e-05 9.326e-06 3.864 0.000112 \*\*\*  
## term 60 months -8.833e-01 6.041e-02 -14.621 < 2e-16 \*\*\*  
## rate -1.507e+00 1.379e+00 -1.093 0.274477   
## payment -7.043e-02 1.263e-02 -5.578 2.44e-08 \*\*\*  
## gradeB -3.138e-01 8.226e-02 -3.815 0.000136 \*\*\*  
## gradeC -6.735e-01 1.087e-01 -6.197 5.76e-10 \*\*\*  
## gradeD -8.502e-01 1.495e-01 -5.688 1.28e-08 \*\*\*  
## gradeE -9.122e-01 1.848e-01 -4.935 8.02e-07 \*\*\*  
## gradeF -9.716e-01 2.454e-01 -3.959 7.54e-05 \*\*\*  
## gradeG -9.158e-01 3.110e-01 -2.945 0.003229 \*\*   
## length1 year -3.695e-03 8.145e-02 -0.045 0.963814   
## length10+ years 9.896e-03 6.278e-02 0.158 0.874750   
## length2 years 1.233e-02 7.601e-02 0.162 0.871085   
## length3 years -3.706e-02 7.772e-02 -0.477 0.633508   
## length4 years -5.165e-02 8.377e-02 -0.617 0.537500   
## length5 years -6.531e-02 8.307e-02 -0.786 0.431759   
## length6 years -9.047e-02 9.092e-02 -0.995 0.319723   
## length7 years -4.554e-02 9.075e-02 -0.502 0.615760   
## length8 years -1.977e-02 8.823e-02 -0.224 0.822650   
## length9 years -6.057e-02 9.731e-02 -0.622 0.533657   
## lengthn/a -4.594e-01 8.694e-02 -5.284 1.27e-07 \*\*\*  
## homeOWN -8.946e-02 5.737e-02 -1.559 0.118916   
## homeRENT -2.420e-01 4.486e-02 -5.395 6.86e-08 \*\*\*  
## income 6.851e-02 5.393e-02 1.270 0.203914   
## verifiedSource Verified -4.186e-02 4.100e-02 -1.021 0.307175   
## verifiedVerified -4.833e-02 4.566e-02 -1.058 0.289918   
## reasoncredit\_card -1.432e-01 1.990e-01 -0.720 0.471684   
## reasondebt\_consolidation -1.293e-01 1.965e-01 -0.658 0.510727   
## reasonhome\_improvement -1.980e-01 2.067e-01 -0.958 0.338088   
## reasonhouse -1.741e-01 3.069e-01 -0.567 0.570602   
## reasonmajor\_purchase -2.393e-02 2.304e-01 -0.104 0.917280   
## reasonmedical -4.837e-01 2.395e-01 -2.020 0.043427 \*   
## reasonmoving -6.139e-01 2.630e-01 -2.334 0.019606 \*   
## reasonother -1.638e-01 2.078e-01 -0.788 0.430531   
## reasonsmall\_business -6.003e-01 2.411e-01 -2.490 0.012775 \*   
## reasonvacation -2.812e-01 2.738e-01 -1.027 0.304373   
## debtIncRat -2.331e-02 2.730e-03 -8.536 < 2e-16 \*\*\*  
## delinq2yr -8.868e-02 1.745e-02 -5.082 3.74e-07 \*\*\*  
## inq6mth -5.757e-02 1.616e-02 -3.562 0.000369 \*\*\*  
## openAcc -3.660e-01 1.368e-01 -2.675 0.007478 \*\*   
## pubRec -2.118e-02 2.513e-02 -0.843 0.399480   
## revolRatio -3.383e-01 1.271e-01 -2.662 0.007762 \*\*   
## totalAcc 1.285e-01 2.016e-02 6.375 1.83e-10 \*\*\*  
## totalBal -1.492e-02 9.270e-03 -1.610 0.107432   
## totalRevLim 9.978e-07 1.210e-06 0.825 0.409617   
## accOpen24 -5.209e-01 4.782e-02 -10.894 < 2e-16 \*\*\*  
## avgBal 2.733e-03 1.353e-02 0.202 0.839916   
## bcOpen 2.048e-02 4.904e-03 4.176 2.97e-05 \*\*\*  
## bcRatio 4.208e-03 1.256e-03 3.350 0.000809 \*\*\*  
## totalLim 2.340e-02 7.726e-03 3.028 0.002461 \*\*   
## totalRevBal 5.139e-03 4.477e-03 1.148 0.251014   
## totalBcLim -3.762e-06 2.103e-06 -1.788 0.073718 .   
## totalIlLim 9.765e-04 2.595e-03 0.376 0.706672   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28786 on 27415 degrees of freedom  
## Residual deviance: 25778 on 27362 degrees of freedom  
## AIC: 25886  
##   
## Number of Fisher Scoring iterations: 5

predictLoan <- predict(loanReg, testingLoan, type='response')  
  
threshhold <- 0.5   
predLoan <- cut(predictLoan, breaks=c(-Inf, threshhold, Inf),   
 labels=c("Good", "Bad"))   
cTab <- table(testingLoan$standing, predLoan)   
addmargins(cTab)

## predLoan  
## Good Bad Sum  
## Bad 213 1301 1514  
## Good 117 5224 5341  
## Sum 330 6525 6855

p <- sum(diag(cTab)) / sum(cTab)   
print(paste('Proportion correctly predicted = ', p))

## [1] "Proportion correctly predicted = 0.793143690736689"

I believe this model would be considered adequate. Its not the best as it predicted 79% of the loans correctly. 64% of the of good loans were correctly predicted and 83% of bad loans were correctly predicted. I think this could be improved substantially. Logistic regression will be used on all remaining predictors in the training data. The regression model will be used on the testing data and a contingency table will be produced to assess the accuracy of the model.

## Section 6: Optimizing the Threshold Accuracy

The previous threshold of 0.5 could be adjusted to correctly predict more bad loans. Throughout this section the threshold value will be manipulated in an attempt to maximized to change the proportion of correctly predicted good and bad loans. A graph will be produced to show the accuracies vs the thresholds used.

# Low threshold  
threshhold <- 0.1   
predLoan <- cut(predictLoan, breaks=c(-Inf, threshhold, Inf),   
 labels=c("Good", "Bad"))   
cTab <- table(testingLoan$standing, predLoan)   
addmargins(cTab)

## predLoan  
## Good Bad Sum  
## Bad 0 1514 1514  
## Good 0 5341 5341  
## Sum 0 6855 6855

p <- sum(diag(cTab)) / sum(cTab)   
print(paste('Proportion correctly predicted = ', p))

## [1] "Proportion correctly predicted = 0.779139314369074"

At low threshold, 0.1 the proportion of correctly predicted loans drops to 77%

# High threshold  
threshhold <- 0.9   
predLoan <- cut(predictLoan, breaks=c(-Inf, threshhold, Inf),   
 labels=c("Good", "Bad"))   
cTab <- table(testingLoan$standing, predLoan)   
addmargins(cTab)

## predLoan  
## Good Bad Sum  
## Bad 1424 90 1514  
## Good 3925 1416 5341  
## Sum 5349 1506 6855

p <- sum(diag(cTab)) / sum(cTab)  
print(paste('Proportion correctly predicted = ', p))

## [1] "Proportion correctly predicted = 0.414296134208607"

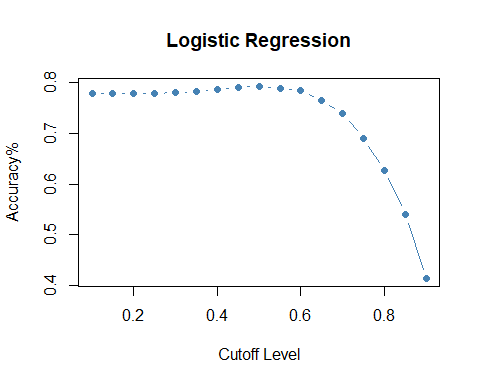
At a high threshold, 0.9, the proportion of correctly predicted loans decreases to 41%

## [1] "Proportion correctly predicted = 0.779139314369074"  
## [2] "Proportion correctly predicted = 0.779139314369074"  
## [3] "Proportion correctly predicted = 0.779139314369074"  
## [4] "Proportion correctly predicted = 0.77928519328957"   
## [5] "Proportion correctly predicted = 0.780889861415025"  
## [6] "Proportion correctly predicted = 0.782494529540481"  
## [7] "Proportion correctly predicted = 0.786870897155361"  
## [8] "Proportion correctly predicted = 0.791684901531729"  
## [9] "Proportion correctly predicted = 0.793143690736689"  
## [10] "Proportion correctly predicted = 0.789059080962801"  
## [11] "Proportion correctly predicted = 0.784099197665937"  
## [12] "Proportion correctly predicted = 0.766301969365427"  
## [13] "Proportion correctly predicted = 0.738730853391685"  
## [14] "Proportion correctly predicted = 0.691320204230489"  
## [15] "Proportion correctly predicted = 0.627862873814734"  
## [16] "Proportion correctly predicted = 0.540481400437637"  
## [17] "Proportion correctly predicted = 0.414296134208607"

The proportion correctly predicted for each tested threshold is as follows:

Proportion 0.10 = 0.779 Proportion 0.15 = 0.779 Proportion 0.20 = 0.779 Proportion 0.25 = 0.779 Proportion 0.30 = 0.780 Proportion 0.35 = 0.782 Proportion 0.40 = 0.787 Proportion 0.45 = 0.792 Proportion 0.50 = 0.793 Proportion 0.55 = 0.789 Proportion 0.60 = 0.784 Proportion 0.65 = 0.766 Proportion 0.70 = 0.739 Proportion 0.75 = 0.691 Proportion 0.80 = 0.628 Proportion 0.85 = 0.540 Proportion 0.90 = 0.414

# Plot thresholds vs accuracy  
plot(thresholds, p, pch=19, type='b', col='steelblue',  
 main='Logistic Regression', xlab='Cutoff Level', ylab='Accuracy%')



# High threshold  
threshhold <- 0.45   
predLoan <- cut(predictLoan, breaks=c(-Inf, threshhold, Inf),   
 labels=c("Good", "Bad"))   
cTab <- table(testingLoan$standing, predLoan)   
addmargins(cTab)

## predLoan  
## Good Bad Sum  
## Bad 134 1380 1514  
## Good 48 5293 5341  
## Sum 182 6673 6855

p <- sum(diag(cTab)) / sum(cTab)  
print(paste('Proportion correctly predicted = ', p))

## [1] "Proportion correctly predicted = 0.791684901531729"

According to the plot it appears that from 0.1 to 0.4 is fairly constant, there is a very minor uptick at 0.5 before it quickly drops off in accuracy after that. 0.45 appears to be the best threshold value for this model having an accuracy of 79%.

## Section 7: Optimizing the Threshold for Profit

From the banks perspective the most important feature of the model is how it changes the overall profit. The threshold analysis will be repeated again but this time focusing on maximizing the profit. The profit will be computed and the assumption will be made that all bad loans predicted will be rejected.

# Create profit column  
testingLoan$profit <- (testingLoan$totalPaid - testingLoan$amount)  
sum(testingLoan$profit)

## [1] 733522.5

threshold <- 0.5   
predLoan <- cut(predictLoan, breaks=c(-Inf, threshold, Inf),   
 labels=c("Bad", "Good"))   
testingLoan$Profitable <- predLoan  
sum(testingLoan[testingLoan$Profitable=="Good",]$profit)

## [1] 2403660

At a threshold of 0.5 the profits would be $2403660.00.

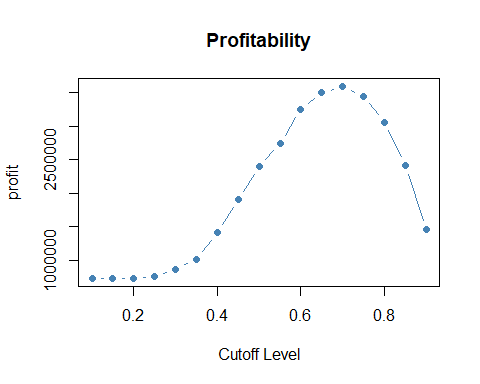
## [1] 733522.5

At a threshold of 0.1 the profits would be $733522.50

## [1] 1459445

At a threshold of 0.9 the profits would be $1459445.00.

thresholds <- seq(0.1,0.9,0.05)   
prof <- NULL  
for(i in seq(along=thresholds)){  
 predLoan <- cut(predictLoan, breaks=c(-Inf, thresholds[i], Inf),   
 labels=c("Bad", "Good"))   
 testingLoan$Profitable <- predLoan  
 prof[i] <- sum(testingLoan[testingLoan$Profitable=="Good",]$profit)  
}



max(prof)

## [1] 3590864

threshhold <- 0.7  
predLoan <- cut(predictLoan, breaks=c(-Inf, threshhold, Inf),   
 labels=c("Bad", "Good"))   
cTab <- table(testingLoan$standing, predLoan)   
addmargins(cTab)

## predLoan  
## Bad Good Sum  
## Bad 738 776 1514  
## Good 1015 4326 5341  
## Sum 1753 5102 6855

p <- sum(diag(cTab)) / sum(cTab)  
print(paste('Proportion correctly predicted = ', p))

## [1] "Proportion correctly predicted = 0.738730853391685"

Profitability has a large swing upwards near 0.4 seems to maximize at 0.7 before falling off sharply again. The maximum profits are $3,590,864.00. The proportion correctly predicted is 73%.

## Section 8: Results Summary

5816517

Applying a logistic regression to the data set I was able to create a model that predicted correctly 79% of the time. This seems rather low and I believe it could be improved with some more work to become more accurate. After running through thresholds 0.1 to 0.9 I was able to determine that a threshold of 0.5 was the most accurate in determining good and bad loans.

Using the same methods but applied towards profitability I was able to determine the that threshold of 0.7 is the most profitable giving 73% correctly predicteed values. While this value is higher than all other thresholds it also involves in giving out riskier loans that may not be paid back based on the accuracy from the previous section. The bank should take this into consideration as there may be an increased risk using the more profitable model than the more accurate one. But the profits of using the higher threshold for profits is substantially greater($3,590,864.00. compared to $2,403,660.00). The modelled profits are greater than the unmodelled profits. The unmodelled profit is $733,522.50 which shows the model is significantly better than the umodelled data.

I believe it would be wiser to take the more profitable approach here rather than the more accurate approach as there is only a 6% difference in accuracy while the profit is significantly higher.