Logistic Regression on Funded Mortgage Data

Robert Gill, Laurie Harris, Hari Sanadhya

Submitted: August 18, 2017

INTRODUCTION

Mortgage loans are relatively complex lending arrangements and generally require several weeks or months to process. There are several steps in the process beginning with loan origination and ending with a final outcome of either funding or not funding. For a typical home buyer, an interest rate can be locked in once they have an executed purchase agreement. Lock periods are customarily 30-45 days prior to closing. Similarly, an interest rate can be locked in for transactions involving home refinancing.

Generally, as a mortgage application moves through the pipeline, it becomes increasingly more likely that the mortgage will fund. When an interest rate is locked, this creates an obligation for the lender to provide financing to the borrower at the agreed upon rate. Because there are many other factors involved, not all locked in mortgages will complete funding. It is necessary for financial mortgage institutions to closely monitor the loans in the mortgage pipeline to ensure they are prepared to meet funding commitments.

It is customary for mortgage companies to manage hedges. Among other benefits, this allows a mortgage companies to manage their coverage commitments and potential loss. It is for this reason that predicting the number of locks that will fund is important. This allows the company to predict the amount of money it will need to cover and adjust the hedge for potential market fluctuations.

PROBLEM STATEMENT

Develop a logistic regression model that can be used to identify significant attributes that will help predict whether a locked mortgage loan will ultimately fund.

DATA SET DESCRIPTION

The raw data set contains 7,576 observations representing mortgage loan applications for which an interest rate has been locked. There are 48 variables represented, including the response variable which is used to indicate whether the locked loan is ultimately funded. In addition, there are several categorical variables, continuous variables, loan-specific ratios, and milestone dates included in the dataset. Milestones track the progression of a loan through the pipeline. The further along the pipeline that the loan progresses, the greater the likelihood that the loan will fund. A new predictor is introduced into the data set that counts the number of milestones that the loan has passed. This predictor has a significant effect on the final model's ability to predict funding.

Table 1 contains a description of the variables in the complete data set.

Variable Name	Type	Description
Loan.Number	Numeric	Observation Identifier
Funded	Boolean	Response - Funded: True or False
Started	Boolean	Loan Originated: True or False
MD.Start	Date	Date of Loan Origination
Qualification	Boolean	Qualification: True or False
MD.Qual	Date	Date of Borrower Qualification
Processing	Boolean	Processing: True or False
MD.Proc	Date	Date Loan Sent to Processing
submittal	Boolean	Submittal: True or False
MD.Sub	Date	Date Loan Submitted to Underwriting
Cond. Approval	Boolean	Conditional Approval: True or False
MD.CApp	Date	Date of Conditional Approval
Resubmittal	Boolean	Resubmittal: True or False
MD.Resub	Date	Date Loan Resubmitted with Conditions
Approval	Boolean	Approval: True or False
MD.App	Date	Date of Loan Approval
Doc Preparation	Boolean	Document Preparation: True or False
MD.DP	Date	Date of Document Processing
Docs Signing	Boolean	Document Signing: True or False
MD.DS	Date	Date of Document Signing
Total.Loan.Amt	Numeric	Dollar Amount of Loan
Prop.Type	Categorical	Property Type (attached, detached, condo, mfg housing, PUD)
Value	Numeric	Dollar Amount of Property Value
Purpose	Categorical	Loan Purpose (purchase, construction, cash-out or no-cash-out refinance)
Occ	Categorical	Property Occupancy (primary, secondary, investment)
Loan.Type	Categorical	Loan Type (conventional, FHA, VA, FarmersHome)
Prop.Type.2	Categorical	Property Type 2 (single family, mfg housing)
Const	Categorical	New Construction (Yes, No)
Purchase.Price	Numeric	Dollar Value of Contracted Price
Down.Pmt	Percentage	Percentage of Purchase Price used for down payment
Base.Loan.Amt	Numeric	Purchase Price less Downpayment
Note.Rate	Percentage	Locked Interest Rate
APR	percentage	Annual Percentage Rate
Term	Categorical	Loan Term (120, 180, 240, 300, 348, 360 months)

PITI	Numeric	Borrower('s) monthly payment (loan principal, interest, taxes and insurance)
PI	numeric	Borrower('s) monthly payment (loan principal and interest only)
Income	numeric	Borrower('s) monthly income
FICO	Numeric	Borrower('s) Lower Median Credit Score
LTV	percentage	Loan Amount / Property Value
CLTV	Percentage	Combined Loan(s) / Property Value
Top.Ratio	percentage	Percentage of Borrowers monthly income used for housing expenses (Front-End Debt-to-Income Ratio - DTI)
Bot.Ratio	Percentage	Percentage of Borrowers monthly income used for all debt expenses (Back-End Debt-to-Income Ratio – DTI)
Lien Position	Categorical	Mortgage Lien Position (first, second)
Payment Shock	Numeric	Assessment of payment shock
Change in Prevailing Rate	Percentage	Change in Prevailing Interest Rate
Lock Rate Less Prevailing Rate	Percentage	Difference between Locked Rate and Prevailing Rate
Change in APOR	Percentage	Change in Average Prime Offer Rate
Rate Spread	Percentage	Rate Spread

Table 1: Listing of Dataset Raw Variables

Exploratory Data Analysis

Since the observations have been uniquely identified in our dataset we have decided to remove the loan identification variable due to privacy considerations. In addition, we have elected to create a new variable which counts the number of milestone dates achieved. Because more value can be derived from how many milestones are met versus the exact date of occurrence, we have removed the specific dates from the dataset.

We also need to consider the context of missing values in the dataset. Figure 1 shows an examination of the missing values for each variable as a percentage of total observations.

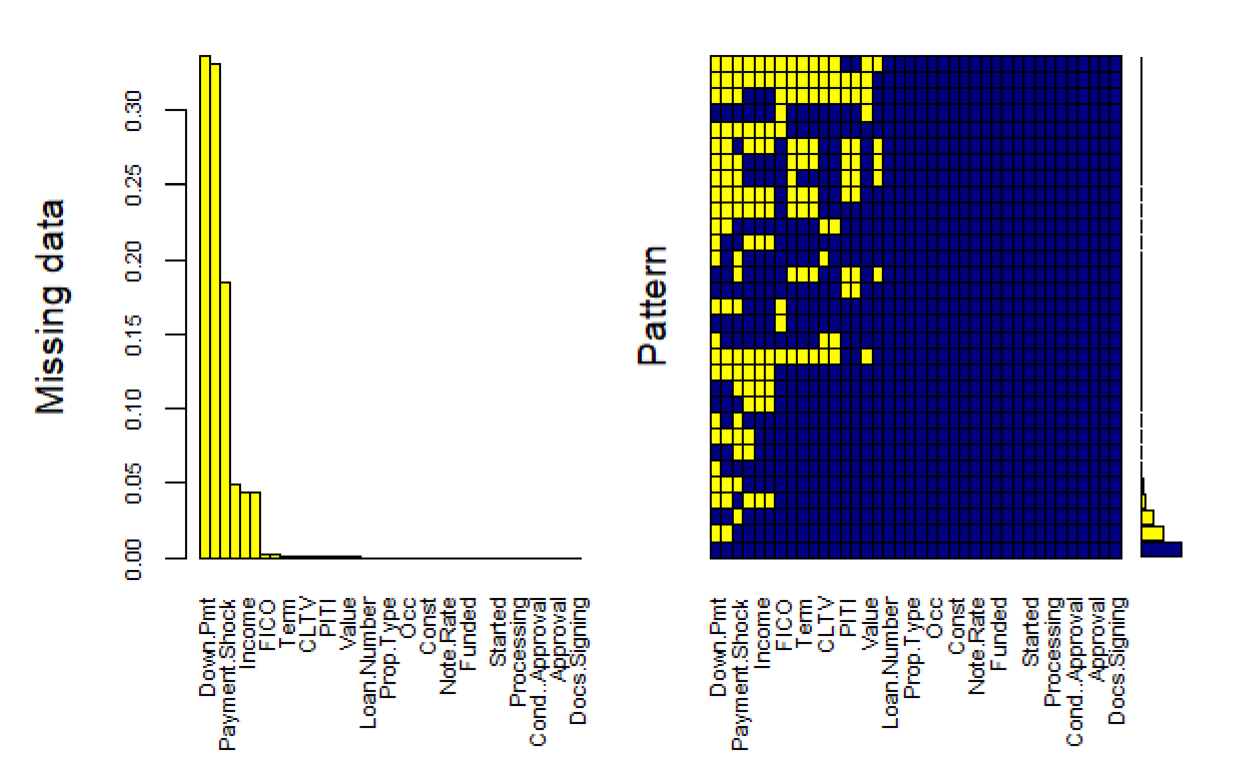


Figure 1 Percentage of Missing Values by Variable

After evaluating the observations for which Purchase Price is missing, we notice that these are related to loans for refinancing. There is no purchase agreement for refinancing loans therefore it is appropriate to have missing values for these observations.

Likewise, many of the missing values for the down payment variable were related to refinanced loans. We notice that the down payment variable can be computed by subtracting the loan value from the purchase price and dividing the result by the purchase price. Since the loan value variable in not missing for any of the observations, we are not concerned with missing values for down payment.

Missing observations for the Payment Shock variable also appear appropriate based on context provided by the mortgage analyst with the company. This is an indicator that is assigned to borrowers with a previous rental or mortgage history. Since these observations do not only represent new home purchasers, missing values are not considered problematic here.

The remaining missing values were for the Combined Loan to Value Ratio and the FICO score. These number of observations with missing values was extremely small, less than 5 observations out of more than 7,000. We considered these to be data error and therefore removed these observations from our analysis.

VARIABLE SCREENING

Based on insights gathered through the exploratory data analysis we have modified some of the raw variables to represent more meaningful attributes upon which to perform our analysis. Following are the new variables we will use in our analysis:

- count_true_flag: This is the number of milestones for which the milestone date is populated. This is different from the count of milestones that are true since the milestones are reset once the rates are locked but the dates of achieving the milestones is not cleaned. The graph below in Figure 2 shows how this flag is affecting the status of funding of the loan with regard to the FICO score variable. The graph suggests that for a loan to be funded, most of the milestones have to be achieved.

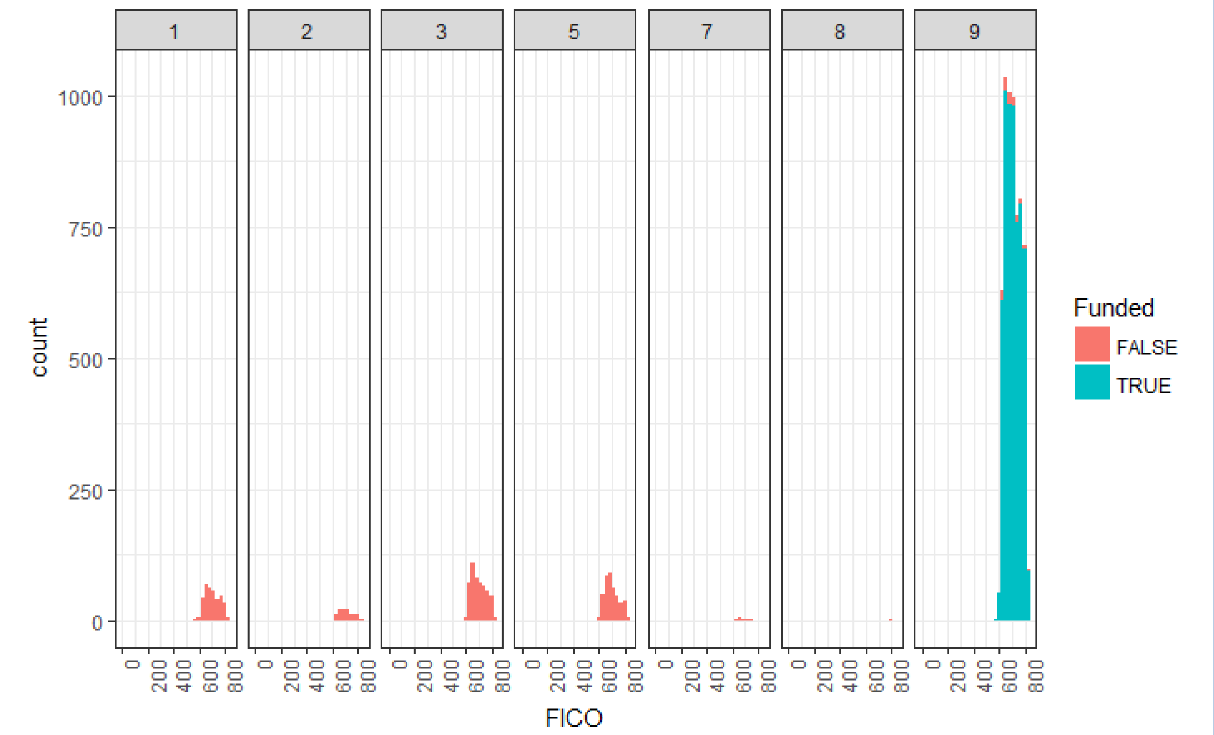


Figure 2: Milestone Count Influence on FICO Credit Score

- Down.Pmt.Present: This flag is set to FALSE if the Down Payment is missing, otherwise it is set to TRUE. Once this value is computed, the missing values for Down Payment variable are set to –1.
- Purchase.Price.Present: This flag is set to FALSE if the Purchase Price is missing, otherwise it is set to TRUE. Once this value is computed, the missing values for Purchase Price are set to –1.

- Payment.Shock: This flag is set to FALSE if the Payment Shock is missing, otherwise it is set to TRUE. Once this value is computed, the missing values for Payment Shock are set to –1.
- FICO_Category: Based on the FICO score recorded on the file (lowest value of median FICO score amongst all applicants). FICO scores were categorized as following (1):
    - 800 and above: Exceptional Credit
    - 740-799: Excellent Credit
    - 670-739: Good Credit
    - 580-669: Fair Credit
    - 579 and below: Poor Credit

MODEL SELECTION

For creating the model, we used the backwards/stepwise model selection algorithm using AIC and BIC with the following interaction terms:

- Income*CLTV: Individual income and CLTV with their interaction
- Top.Ratio*Bot.Ratio: Individual front end and back end DTI with their interaction
- FICO_Category*LTV: Individual FICO_Category and LTV with their interaction
- Payment.Shock.Present:Payment.Shock: Just the interaction term as missing values in the Payment Shock are represented by the Payment.Shock.Present flag.
- Purchase.Price.Present:Purchase.Price: Just the interaction term as missing values in the Payment Shock are represented by the Purchase.Price.Present flag.
- Down.Pmt.Present:Down.Pmt: Just the interaction term as missing values in the Payment Shock are represented by the Down.Pmt.Present flag.
- count_true_flag*(Started + Qualification + Processing + submittal + Cond..Approval + Resubmittal + Approval + Doc.Preparation + Docs.Signing): Individual milestones with the interaction with the count_true_flag as milestones achieved after rate being locked may play significant role in the decision of the deal apart from all the milestones achieved.
- Rate.Spread*Bot.Ratio: Interaction between front end and back end DTI.
- PITI*Bot.Ratio: Interaction between PITI and back end DTI
- Purpose*Bot.Ratio: Interaction between Purpose of the loan and back end Ratio.

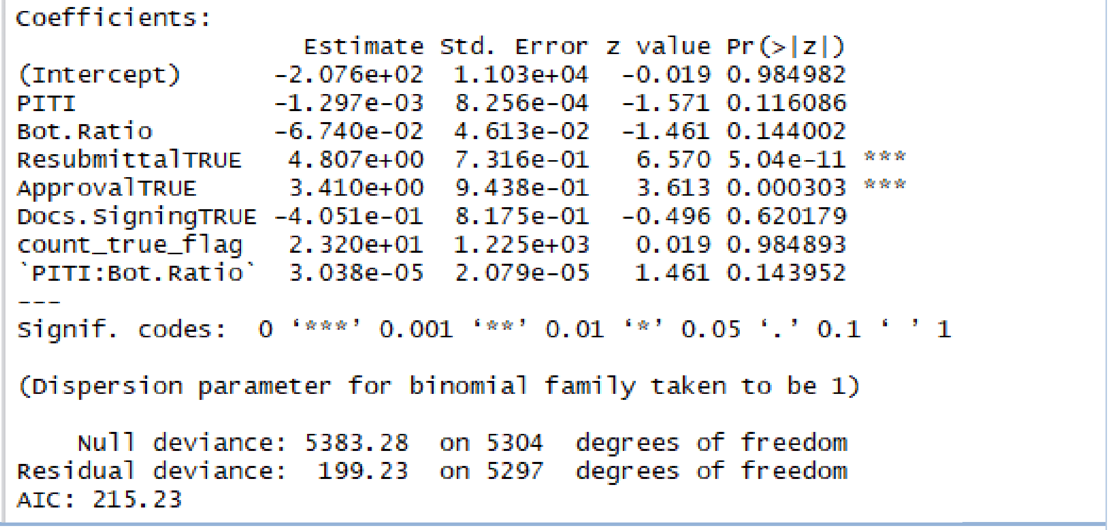
Randomly Split data set into test and train with training set representing 70% of observations (5300 obs) test 30% (2271 obs)

The previously known probability of Funded loans as determined by the population is 79.15%. This percentage value would be used in prediction.

After running selection, the following model was generated:

Funded ~ PITI + Bot.Ratio + Resubmittal + Approval + Docs.Signing + count_true_flag + PITI:Bot.Ratio

This model was then tested through a 10-fold cross-validation. The coefficient estimates generated from this cross-validation are the coefficients used for the final model.



```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)      -2.076e+02  1.103e+04  -0.019 0.984982    
PITI             -1.297e-03  8.256e-04  -1.571 0.116086    
Bot.Ratio        -6.740e-02  4.613e-02  -1.461 0.144002    
ResubmittalTRUE   4.807e+00  7.316e-01   6.570 5.04e-11 ***
ApprovalTRUE      3.410e+00  9.438e-01   3.613 0.000303 ***
Docs.SigningTRUE -4.051e-01  8.175e-01  -0.496 0.620179    
count_true_flag   2.320e+01  1.225e+03   0.019 0.984893    
`PITI:Bot.Ratio`  3.038e-05  2.079e-05   1.461 0.143952    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5383.28  on 5304  degrees of freedom
Residual deviance:  199.23  on 5297  degrees of freedom
AIC: 215.23
```

Figure 3: Predictor estimates

Figure 3 above shows the predictor estimates after the 10-fold cross-validation. The information that is the most important in this figure is the deviance. The deviance for the Null model is 5383.28. The residual deviance for the final model is 199.23. This large drop in deviance demonstrates large improvement in significance of the final model over that of the null model.

Result

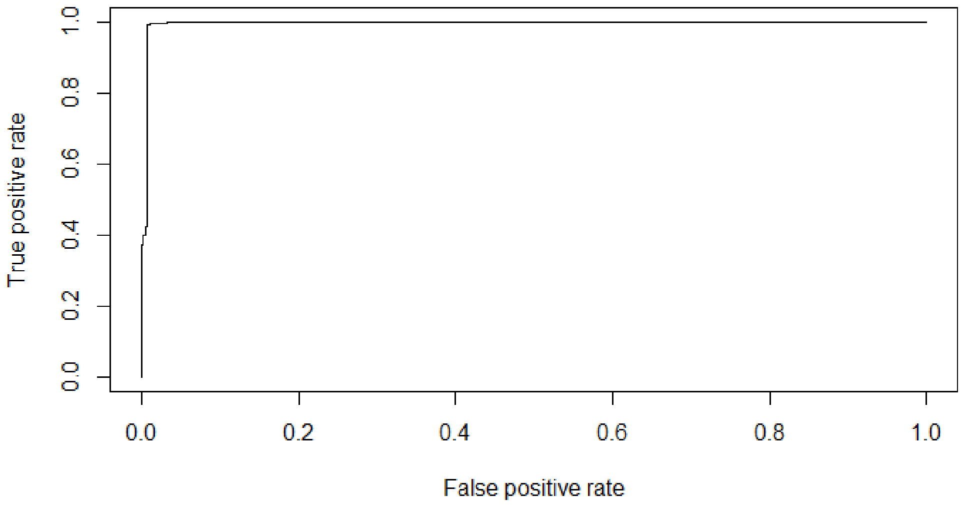
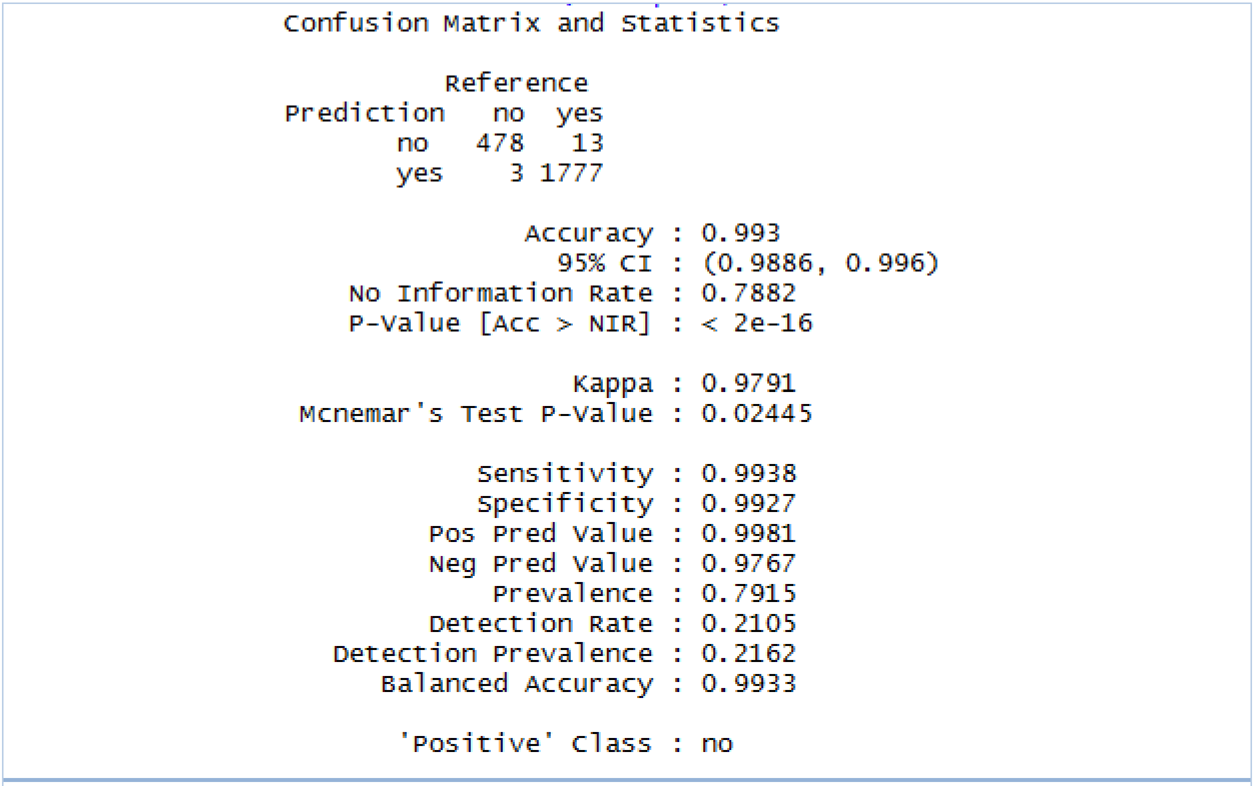


Figure 4: ROC Curve

Figure 4 above is the ROC curve for our final model. The area under the curve(AUC) is 0.99. This area is very close to 1. the AUC is a performance metric that is sometimes referred to as the index of accuracy. The close the value is to 1, the better the prediction power of the model. The ROC curve itself summarizes the true positive rate(sensitivity) and false positive rate(1-specificity). Figure 3 demonstrates that model has both a high sensitivity and high specificity.



```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no   478   13
       yes    3 1777

               Accuracy : 0.993
                 95% CI : (0.9886, 0.996)
    No Information Rate : 0.7882
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9791
 Mcnemar's Test P-Value : 0.02445

            Sensitivity : 0.9938
            Specificity : 0.9927
         Pos Pred Value : 0.9981
         Neg Pred Value : 0.9767
             Prevalence : 0.7915
         Detection Rate : 0.2105
   Detection Prevalence : 0.2162
      Balanced Accuracy : 0.9933

       'Positive' Class : no
```

Figure 5: Confusion Matrix

Figure 5 above is the Confusion matrix for our model's effect on the test dataset. In the figure we can see the confusion matrix that confirms what was demonstrated by the ROC curve: high sensitivity and high specificity. In addition, we are able to see the accuracy of the model as well as the 95% confidence level for the accuracy. This is confirmation of the predictive power of the final model.

CONCLUSIONS

```
Final Model: -
Funded ~ PITI + Bot.Ratio + Resubmittal + Approval + Docs.Signing + count_true_flag +
PITI:Bot.Ratio
```

The model developed has almost a perfect prediction ability. This is suspected to be due to the inclusion of the "count_true_flag" predictor. Previous models were built without the creation of this variable and resulted in an AUC of approximately 65%. The final model has an AUC of over 99%. The AUC is a measure of the specificity and sensitivity of the model. This means that the final model has both a high true positive rate and a low false positive rate. The other predictors in the model are also logical. PITI reflects the financial burden that a borrower would experience if they were to undertake the loan. Bot.Ratio is the borrower's debt to income ratio. This is another measure of the borrower's financial ability to fulfill the payments of the loan. Resubmittal, Approval, and Docs.Signing are milestones. These milestones occur late in the pipeline and act to increase the odds that a file will fund if it is further down the pipeline. In addition, there is an interaction term for PITI and Bot.Ratio as both are a measure of the financial burden that the loan places on the borrower. These predictors demonstrate that the model is logical given the objective.

Constraints and Limitations

The analysis was completed on mortgage applications that originated between July 2015 and January 2017. The locks range across the country as they originate from many branches in the corporate structure. Because these data are observational, no causal inferences can be made. It is unknown whether there are additional factors that may that may influence borrowers in selecting this particular company for their mortgage application. Therefore, it would be inappropriate to draw inferences to other populations, specific financial institutions or historical time periods. The data used is financial in nature. This results in the lifetime of the model being subject to the current state of the market. If the market were to have a large fluctuation or a shift over time, this model could become obsolete.

Reference

1. http://www.thesimpledollar.com/what-is-a-good-credit-score/

Appendix

R - Code

```
library(glmnet)
library(ROCR)
library(MASS)
library(dplyr)
library(ggplot2)
library(tree)
library(Amelia)
library(reshape2)
library(caret)
library(e1071)
library(mice)
library(VIM)

setwd("C:\\Users\\Neetu
Hari\\Desktop\\SMU\\MSDS 6372
Applied Statistics Inference and
Modeling - 403\\Project3_403")
dat <- read.csv("Input Data File.csv",
header = TRUE, na.strings = c(""))
# Count the number of Milestones
that are set to TRUE
dat$count_true_flag <- 9 -
apply(dat[,c(3:11)],1, function(x)
sum(is.na(x)))
```

```
# FICO categorization
# Source -
http://www.thesimpledollar.com/what
-is-a-good-credit-score/
FICO_Category <-
data.frame(sapply(dat[,c(12)],
 function(x)
 if(is.na(x))
 {"New or Not Found or Special
Case"}
 else if(x >= 800)
 {"Exceptional Credit"}
 else if(x>=740)
 {"Excellent Credit"}
 else if(x>=670)
 {"Good Credit"}
 else if(x>=580)
 {"Fair Credit"}
 else if(x>0)
 {"Poor Credit"}
 else {"New or Not Found or Special
Case"}))
```

```
FICO_Category <- cbind(variable =
row.names(FICO_Category),FICO_
Category)
names(FICO_Category)=c("Variable"
,"FICO_Category")
dat$FICO_Category =
FICO_Category$FICO_Category

# Compute Down.Pmt.Present flag
dat$Down.Pmt.Present <-
apply(dat[,"Down.Pmt",drop=F],1,fun
ction(x) !(is.na(x)))
# Compute Purchase.Price.Present
flag
dat$Purchase.Price.Present <-
apply(dat[,"Purchase.Price",drop=F],
1,function(x) !(is.na(x)))
# Compute Payment.Shock.Present
flag
dat$Payment.Shock.Present <-
apply(dat[,"Payment.Shock",drop=F],
1,function(x) !(is.na(x)))
# Replace missing down payment by
-1
```

```r
dat$Down.Pmt <- apply(dat[,"Down.Pmt",drop=F],1, function(x) if(is.na(x)){-1}else{x})
# Replace missing Purchase Price by -1
 dat$Purchase.Price <- apply(dat[,"Purchase.Price",drop=F], 1, function(x) if(is.na(x)){-1}else{x})
# Replace missing Payment shock by -1
dat$Payment.Shock <- apply(dat[,"Payment.Shock",drop=F], 1, function(x) if(is.na(x)){-1}else{x})
# Remove the unwanted columns which are not required for the analysis
# Remove the milestone date flags and the loan number as they are not model predictors
dat3 <- dat[,-c(1, 3, 4, 5, 6, 7, 8, 9, 10, 11)]

# Randomly divide dataset into train (70%) and test (30%)
dat.train.dataset <- sample_frac(dat3,size = 0.7)
dat.test.dataset <- dat3[-as.numeric(rownames(dat.train.dataset)),]

# Missing data analysis
mice_plot <- aggr(dat3, col=c('navyblue','yellow'), numbers = TRUE, sortVars=TRUE, labels=names(data.raw2),cex.axis=.7,gap=3, ylab=c("Missing data","Pattern"))

# Remove the missing values from the train data
dat.train.dataset <- dat.train.dataset[complete.cases(dat.train.dataset),]

# Fitting Generalized Linear Model
model <- glm(Funded ~ Total.Loan.Amt + FICO + Prop.Type
+ Value + Purpose + Occ +
Loan.Type +  Prop.Type.2 + Const +
Base.Loan.Amt + Note.Rate + APR
+ Term + PITI + PI +  Income*CLTV
+ Top.Ratio*Bot.Ratio +
Lien.Position + Payment.Shock +
Change.in.Prevailing.Rate +
Lock.Rate.Less.Prevailing.Rate +
Change.in.APOR + Rate.Spread +
Started + Qualification + Processing
+  submittal + Cond..Approval +
Resubmittal + Approval +
Doc.Preparation + Docs.Signing +
FICO_Category*LTV +
Payment.Shock.Present:Payment.Shock +
Purchase.Price.Present:Purchase.Price + Down.Pmt.Present:Down.Pmt
+  count_true_flag*(Started +
Qualification + Processing +
submittal +  Cond..Approval +
Resubmittal + Approval +
Doc.Preparation + Docs.Signing) +
Rate.Spread*Bot.Ratio +
PITI*Bot.Ratio + Purpose*Bot.Ratio,
family=binomial(link="logit"),
data=dat.train.dataset)

# Print the model summary
summary(model)

# Automatic model selection using AIC
## Backwards
backwards <- step(model, trace=0)
formula(backwards)
## Stepwise
both <- step(model, trace=0,direction=c("both"))
formula(both)

# Automatic model selection using AIC
## Backwards
backwards_BIC <- step(model, trace=0,k=log(5300))
formula(backwards_BIC)
## Stepwise
both_BIC <- step(model, trace=0,direction=c("both"),k=log(5300))
formula(both_BIC)

# Convert the reference variable values from TRUE/FALSE to yes/no as caret package requires reference variable value to be yes/no
dat.train.dataset$Funded <- apply(as.array(dat.train.dataset$Funded),1, function(x) if (x=='TRUE') {'yes'} else {'no'})
dat.test.dataset$Funded <- apply(as.array(dat.test.dataset$Funded),1, function(x) if (x=='TRUE') {'yes'} else {'no'})

# K-fold (10 fold) cross validation to get the parameter estimates

## For model selected by backward using AIC (stepwise AIC has same formula as backward AIC)
ctrl <- trainControl(method = "repeatedcv", number = 10, p=0.75, savePredictions=T)
mod_fit <- train(formula(backwards), data=dat.test.dataset, method="glm", family="binomial",
 trControl = ctrl, tuneLength = 5)
summary(mod_fit)
library(ROCR)
# ROC
pr <- prediction(mod_fit$finalModel$fitted.values, dat.test.dataset$Funded)
prf <-performance(pr,measure = "tpr", x.measure= "fpr")
plot(prf)
# Compute AUC
 auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
# prediction on the test data
pred <- predict(mod_fit, newdata=dat.test.dataset)
confusionMatrix(data=pred, dat.test.dataset[,c("Funded")], prevalence = 0.7915)

## For model selected by backward_BIC (stepwise_BIC has same formula as backward_BIC)
ctrl <- trainControl(method = "repeatedcv", number = 10, p=0.75, savePredictions=T)
mod_fit <- train(formula(backwards_BIC), data=dat.test.dataset, method="glm", family="binomial",
 trControl = ctrl, tuneLength = 5)
summary(mod_fit)
summary(backwards_BIC)
library(ROCR)
# ROC
pr <- prediction(mod_fit$finalModel$fitted.values, dat.test.dataset$Funded)
prf <-performance(pr,measure = "tpr", x.measure= "fpr")
plot(prf)
# Compute AUC
 auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]

# prediction on the test data
pred <- predict(mod_fit, newdata=dat.test.dataset)
confusionMatrix(data=pred, dat.test.dataset[,c("Funded")], prevalence = 0.7915)
```