library(tidyverse)

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

## -- Attaching packages -------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ----------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)

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

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(ROCR)

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

## Loading required package: gplots

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

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

parole <- read\_csv("C:/Users/Evan/Desktop/BAN 502/Module 3/Assignment 2/parole.csv")

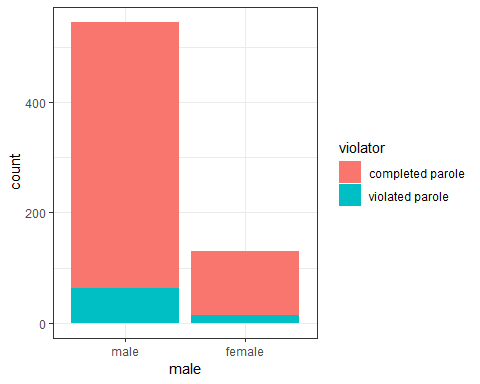
## Parsed with column specification:  
## cols(  
## male = col\_integer(),  
## race = col\_integer(),  
## age = col\_double(),  
## state = col\_integer(),  
## time.served = col\_double(),  
## max.sentence = col\_integer(),  
## multiple.offenses = col\_integer(),  
## crime = col\_integer(),  
## violator = col\_integer()  
## )

View(parole)  
  
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"male" = "1",  
"female" = "0"))  
  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"white" = "1",  
"otherwise" = "2"))  
  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Other" = "1",  
"Kentucky" = "2",  
"Louisana" = "3",  
"Virginia" = "4"))  
  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
mutate(crime = fct\_recode(crime,  
"Other" = "1",  
"larceny" = "2",  
"drug-related" = "3",  
"driving-related" = "4"))  
  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"multiple offenses" = "1",  
"otherwise" = "0"))  
  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"violated parole" = "1",  
"completed parole" = "0"))

set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE) #70% in training  
train = parole[train.rows,]   
test = parole[-train.rows,]

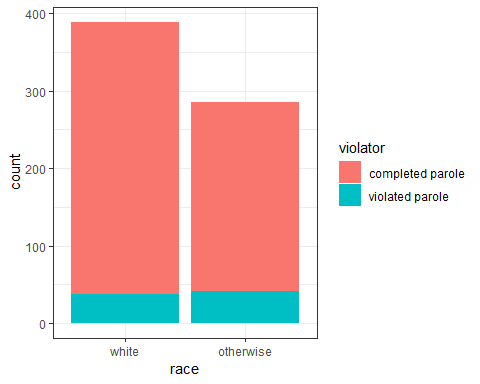
Visuals Gender

ggplot(parole, aes(x=male, fill = violator)) + geom\_bar() + theme\_bw()

 It appears that overall more males than females have been arrested. But that their rate of violating parole is faily similar, males might violate their parole more often than females but not by a large amount. I would not think that this is the best predictor variable.

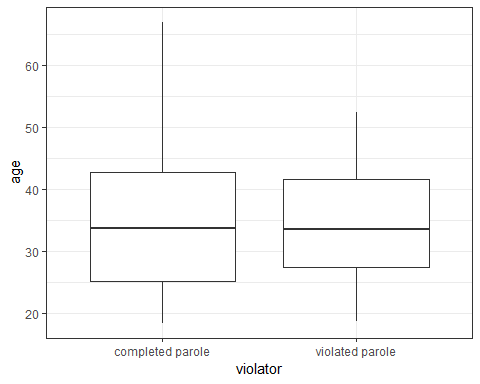
Race

ggplot(parole, aes(x=race, fill = violator)) + geom\_bar() + theme\_bw()

 It appears that non-whites violate their parole at a higher rate than whites.

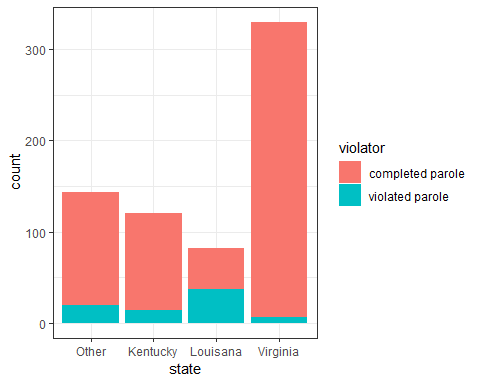
Age

ggplot(parole, aes(x= violator, y = age)) + geom\_boxplot() + theme\_bw()

 Age does not appear to play a significant factor into whether or not someone completes or violates their parole.

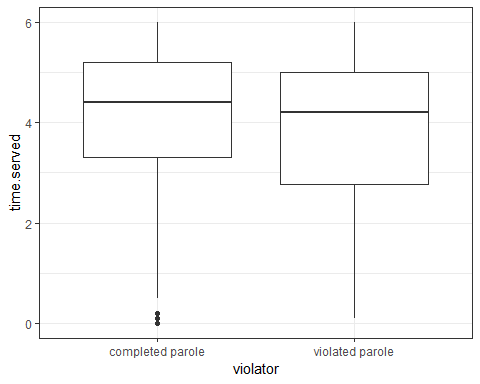
State

ggplot(parole, aes(x=state, fill = violator)) + geom\_bar() + theme\_bw()

 People in Louisana violate their parole at a significanty higher rate than those in Kentucky, Virginia or other states. This so far is the best predictor.

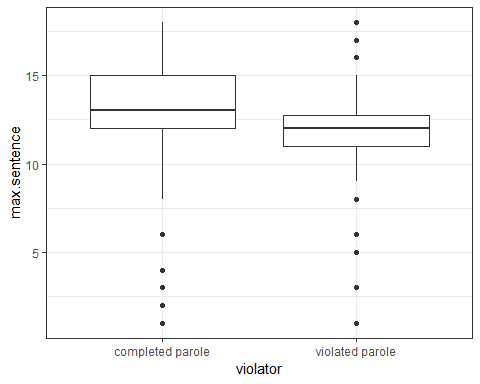
Time Served

ggplot(parole, aes(x= violator, y = time.served)) + geom\_boxplot() + theme\_bw()

 It appears that the longer time served the more likely someone will complete their parole. This makes sense, as it would make logical sense for someone who has spent more time in prison would be less willing to go back, under normal circumstances.

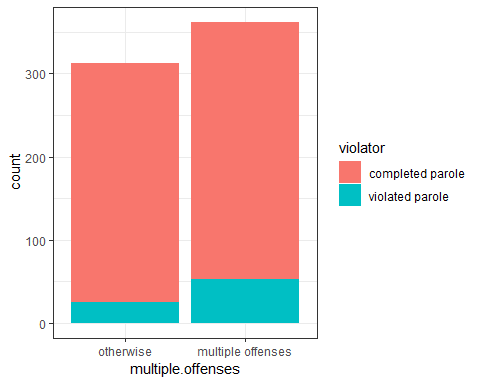
Max Sentence

ggplot(parole, aes(x= violator, y = max.sentence)) + geom\_boxplot() + theme\_bw()

 Another variable that appears to play a significant role. The longer their maximum sentence appears to correlate to a more succesful parole completion rate.

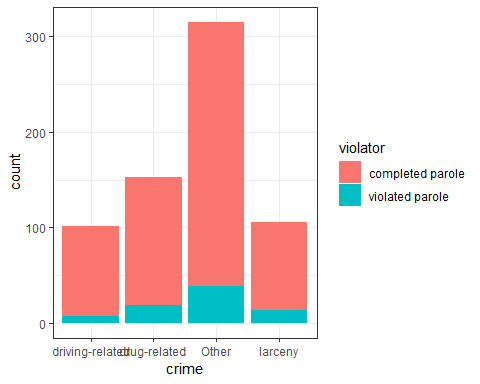
Mutliple Offenses

ggplot(parole, aes(x=multiple.offenses, fill = violator)) + geom\_bar() + theme\_bw()

 Those that have committed multiple offenses are more likely to violate their parole.

Crime

ggplot(parole, aes(x=crime, fill = violator)) + geom\_bar() + theme\_bw()



mod1 = glm(violator ~ max.sentence , parole, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ max.sentence, family = "binomial", data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0615 -0.5137 -0.4780 -0.3841 2.4208   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.12606 0.44516 -0.283 0.777   
## max.sentence -0.15273 0.03576 -4.271 1.94e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 465.68 on 673 degrees of freedom  
## AIC: 469.68  
##   
## Number of Fisher Scoring iterations: 5

I chose to run my model using the maximum sentence variable. The variable is statsitically signifcant and the AIC is 469.68.

allmod = glm(violator ~., train, family = "binomial")  
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9635 -0.3638 -0.2354 -0.1449 2.9869   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -5.33220 1.39750 -3.816 0.000136  
## malefemale -0.53377 0.49107 -1.087 0.277051  
## raceotherwise 1.06698 0.41324 2.582 0.009824  
## age 0.03361 0.01696 1.982 0.047493  
## stateKentucky -0.30132 0.56939 -0.529 0.596665  
## stateLouisana 0.87804 0.52428 1.675 0.093984  
## stateVirginia -3.46523 0.63742 -5.436 5.44e-08  
## time.served -0.03009 0.12159 -0.247 0.804537  
## max.sentence 0.08458 0.05644 1.499 0.133963  
## multiple.offensesmultiple offenses 1.72841 0.41857 4.129 3.64e-05  
## crimedrug-related 0.11232 0.71712 0.157 0.875535  
## crimeOther 0.87795 0.62271 1.410 0.158571  
## crimelarceny 1.06304 0.73146 1.453 0.146139  
##   
## (Intercept) \*\*\*  
## malefemale   
## raceotherwise \*\*   
## age \*   
## stateKentucky   
## stateLouisana .   
## stateVirginia \*\*\*  
## time.served   
## max.sentence   
## multiple.offensesmultiple offenses \*\*\*  
## crimedrug-related   
## crimeOther   
## crimelarceny   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 230.16 on 460 degrees of freedom  
## AIC: 256.16  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(violator ~1, train, family = "binomial")   
summary(emptymod)

##   
## Call:  
## glm(formula = violator ~ 1, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4972 -0.4972 -0.4972 -0.4972 2.0745   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.0281 0.1434 -14.14 <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: 340.04 on 472 degrees of freedom  
## Residual deviance: 340.04 on 472 degrees of freedom  
## AIC: 342.04  
##   
## Number of Fisher Scoring iterations: 4

backmod = stepAIC(allmod, direction = "backward", trace = TRUE)

## Start: AIC=256.16  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - time.served 1 230.22 254.22  
## - crime 3 235.30 255.30  
## - male 1 231.41 255.41  
## <none> 230.16 256.16  
## - max.sentence 1 232.46 256.46  
## - age 1 234.09 258.09  
## - race 1 236.97 260.97  
## - multiple.offenses 1 248.67 272.67  
## - state 3 304.40 324.40  
##   
## Step: AIC=254.22  
## violator ~ male + race + age + state + max.sentence + multiple.offenses +   
## crime  
##   
## Df Deviance AIC  
## - crime 3 235.38 253.38  
## - male 1 231.56 253.56  
## <none> 230.22 254.22  
## - max.sentence 1 232.50 254.50  
## - age 1 234.09 256.09  
## - race 1 236.97 258.98  
## - multiple.offenses 1 249.39 271.39  
## - state 3 304.94 322.95  
##   
## Step: AIC=253.38  
## violator ~ male + race + age + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 236.28 252.28  
## <none> 235.38 253.38  
## - max.sentence 1 237.41 253.41  
## - age 1 238.26 254.26  
## - race 1 242.32 258.32  
## - multiple.offenses 1 255.31 271.31  
## - state 3 309.30 321.30  
##   
## Step: AIC=252.28  
## violator ~ race + age + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## - max.sentence 1 238.31 252.31  
## - age 1 238.81 252.81  
## - race 1 243.44 257.44  
## - multiple.offenses 1 256.39 270.39  
## - state 3 309.81 319.80

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ race + age + state + max.sentence +   
## multiple.offenses, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7414 -0.3643 -0.2668 -0.1502 2.7714   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -4.60426 1.13428 -4.059 4.92e-05  
## raceotherwise 1.07386 0.40527 2.650 0.00806  
## age 0.02636 0.01660 1.588 0.11224  
## stateKentucky -0.41360 0.54930 -0.753 0.45147  
## stateLouisana 0.86000 0.51900 1.657 0.09751  
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08  
## max.sentence 0.07733 0.05475 1.412 0.15788  
## multiple.offensesmultiple offenses 1.77974 0.41476 4.291 1.78e-05  
##   
## (Intercept) \*\*\*  
## raceotherwise \*\*   
## age   
## stateKentucky   
## stateLouisana .   
## stateVirginia \*\*\*  
## max.sentence   
## multiple.offensesmultiple offenses \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

I used a backwards stepwise function to have R derive this model. The final model that it deemed best had an AIC of 252.28 which is an improvement from my initial model that only used the max sentence variable. This model finds that race, state and multiple offenses are statistically significant. The model allows us to see a few trends that were noted above in the visuals section. People in the start of louisana are more likely to violate parole while those in Kentucky and Virginia are not. Those with multiple offenses are also more likely to violate their parole as well as non-whites, which are all observations that were noted previously.

mod2 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial" )  
summary(mod2)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4553 -0.3862 -0.2931 -0.1787 2.8791   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.5582 0.3709 -6.898 5.28e-12  
## stateKentucky -0.4816 0.5417 -0.889 0.3740  
## stateLouisana 0.5292 0.4769 1.110 0.2672  
## stateVirginia -3.2301 0.6028 -5.358 8.39e-08  
## multiple.offensesmultiple offenses 1.6596 0.3985 4.165 3.12e-05  
## raceotherwise 1.0024 0.3966 2.528 0.0115  
##   
## (Intercept) \*\*\*  
## stateKentucky   
## stateLouisana   
## stateVirginia \*\*\*  
## multiple.offensesmultiple offenses \*\*\*  
## raceotherwise \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 240.42 on 467 degrees of freedom  
## AIC: 252.42  
##   
## Number of Fisher Scoring iterations: 6

This model has an AIC value of 252.42 which is very similar to our previous model. Race, and multiple offenses remains significant as well as the state of virginia.

Predictions

newdata = data.frame(state = "Louisana", multiple.offenses = "multiple offenses", race = "white")  
predict(mod2, newdata, type="response")

## 1   
## 0.408682

Another Prediction

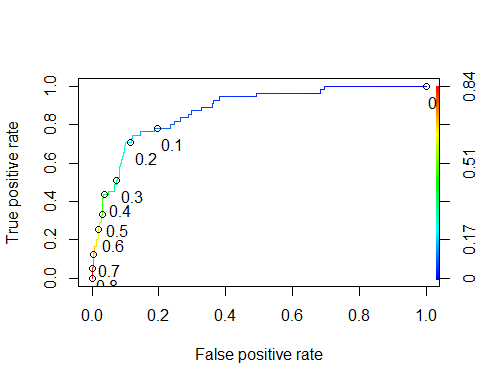
newdata = data.frame(state = "Kentucky", multiple.offenses = "otherwise", race = "otherwise")  
predict(mod2, newdata, type="response")

## 1   
## 0.1153326

predictions = predict(backmod, type="response") #develop predicted probabilities  
head(predictions)

## 1 2 3 4 5 6   
## 0.08809332 0.13886059 0.06774485 0.11577619 0.28753466 0.08354414

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, train$violator)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7636364  
## specificity 0.8540670  
## cutoff 0.1455707

The sensitivity and specificity of the curve are listed above.

#confusion matrix  
t1 = table(train$violator,predictions > 0.1455707)  
t1

##   
## FALSE TRUE  
## completed parole 357 61  
## violated parole 14 41

(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8414376

The accuracy of this model given that we balance sensitivity and specificity is listed above. The risk we run by mis-classifying in this instance is that we may keep people incarcerated that would successfully complete their parole, or may release some people on parole that should not be as they will violate the conditions of their parole.

Trial and error of new thresholds

t1 = table(train$violator,predictions > 0.3)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8773784

t1 = table(train$violator,predictions > 0.4)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.9027484

t1 = table(train$violator,predictions > 0.5)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8942918

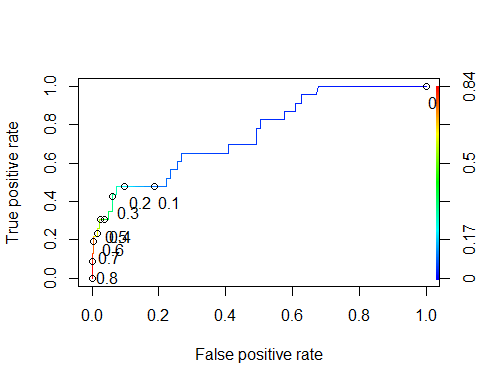
t1 = table(train$violator,predictions > 0.45)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.9006342

predict\_test = predict(backmod, newdata = test, type = "response")  
head(predict\_test)

## 1 2 3 4 5 6   
## 0.06724857 0.48149630 0.07086229 0.04791897 0.09518625 0.03155000

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predict\_test, test$violator)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.65217391  
## specificity 0.73184358  
## cutoff 0.07307531

#confusion matrix  
t1 = table(test$violator,predict\_test > 0.4)  
t1

##   
## FALSE TRUE  
## completed parole 173 6  
## violated parole 16 7

(t1[1,1]+t1[2,2])/nrow(test)

## [1] 0.8910891

The accuracy of the .4 probablity threshold on the test set is .8910891