Assignment3.rmd

Jordan Whitaker

5/25/2020

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3

## -- Attaching packages ------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tidyr' was built under R version 3.6.3

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'stringr' was built under R version 3.6.3

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

library(MASS)

## Warning: package 'MASS' was built under R version 3.6.3

##   
## Attaching package: 'MASS'

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

library(caret)

## Warning: package 'caret' was built under R version 3.6.3

## 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 3.6.3

parole = read\_csv("parole.csv")

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

head(parole)

## # A tibble: 6 x 9  
## male race age state time.served max.sentence multiple.offens~ crime  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1 33.2 1 5.5 18 0 4  
## 2 0 1 39.7 1 5.4 12 0 3  
## 3 1 2 29.5 1 5.6 12 0 3  
## 4 1 1 22.4 1 5.7 18 0 1  
## 5 1 2 21.6 1 5.4 12 0 1  
## 6 1 2 46.7 1 6 18 0 4  
## # ... with 1 more variable: violator <dbl>

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",  
"other" = "2"))  
  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Other" = "1",  
"Kentucky" = "2",  
"Lousiana" = "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(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"no-violation" = "0",  
"violated-parole" = "1"))  
  
head(parole)

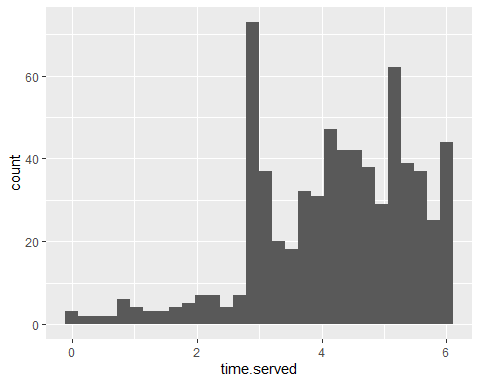
## # A tibble: 6 x 9  
## male race age state time.served max.sentence multiple.offens~ crime  
## <fct> <fct> <dbl> <fct> <dbl> <dbl> <dbl> <fct>  
## 1 male white 33.2 Other 5.5 18 0 driv~  
## 2 fema~ white 39.7 Other 5.4 12 0 drug~  
## 3 male other 29.5 Other 5.6 12 0 drug~  
## 4 male white 22.4 Other 5.7 18 0 Other  
## 5 male other 21.6 Other 5.4 12 0 Other  
## 6 male other 46.7 Other 6 18 0 driv~  
## # ... with 1 more variable: violator <fct>

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

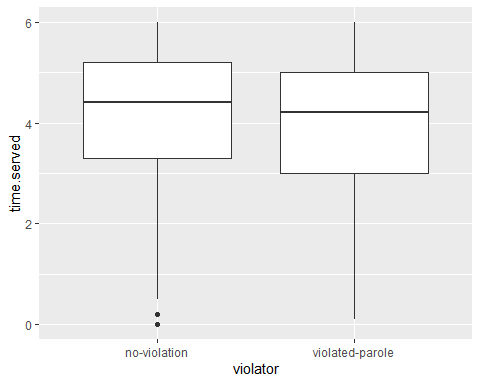
Task 2: My thoughts were to put eyes on most of the variables since there are only 9. Most do not seem to be overwhelmingly indicative of a parole violation other than male. Relative to the other visualizations, if a parole is male, I am going to hypothesize that they will have a higher rate of violating their parole.

ggplot(parole, aes(x=time.served)) + geom\_histogram()

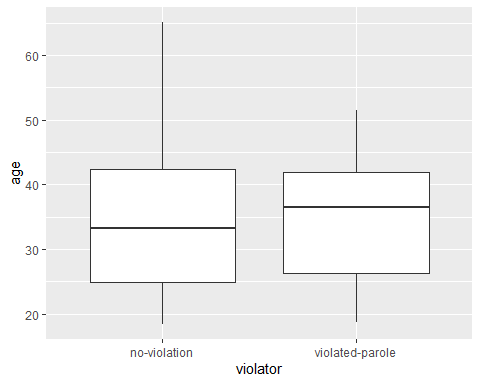
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



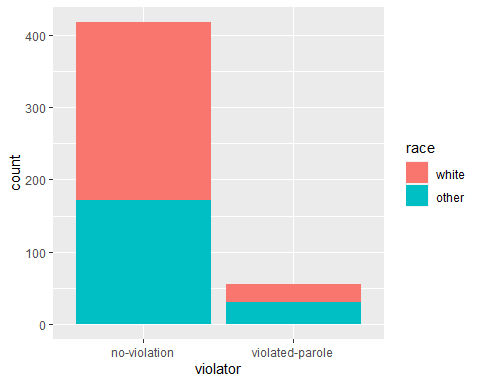
ggplot(train, aes(x=violator, y=time.served)) + geom\_boxplot()



ggplot(train, aes(x=violator, y=age)) + geom\_boxplot()



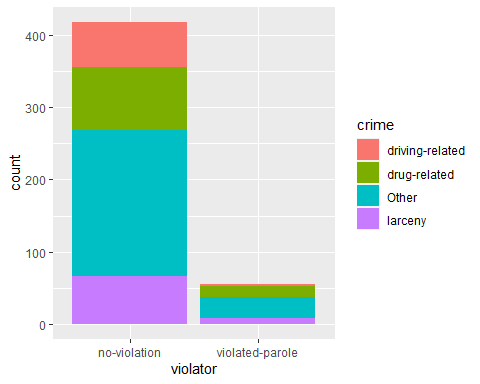
ggplot(train, aes(x=violator, fill=race)) + geom\_bar()



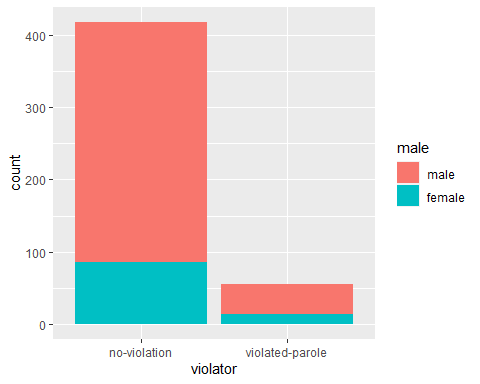
t2 = table(train$violator, train$race)  
prop.table(t2, margin=2)

##   
## white other  
## no-violation 0.90774908 0.85148515  
## violated-parole 0.09225092 0.14851485

ggplot(train, aes(x=violator, fill= crime)) + geom\_bar()



ggplot(train, aes(x=violator, fill= male)) + geom\_bar()



t = table(train$violator, train$state)  
prop.table(t, margin=2)

##   
## Other Kentucky Lousiana Virginia  
## no-violation 0.85263158 0.85542169 0.58620690 0.97890295  
## violated-parole 0.14736842 0.14457831 0.41379310 0.02109705

The variable I chose is not significant :( Has an AIC of 143.89 … lets see how low it can go

test\_mod = glm(violator ~ male, test, family ="binomial")  
summary(test\_mod)

##   
## Call:  
## glm(formula = violator ~ male, family = "binomial", data = test)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5265 -0.5265 -0.5265 -0.2520 2.6328   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.9062 0.2285 -8.342 <2e-16 \*\*\*  
## malefemale -1.5278 1.0414 -1.467 0.142   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 143.22 on 201 degrees of freedom  
## Residual deviance: 139.89 on 200 degrees of freedom  
## AIC: 143.89  
##   
## Number of Fisher Scoring iterations: 6

Making the best model… I think I can I think I can….

allmod = glm(violator ~., train, family ="binomial")  
summary(allmod) #aic of 268.09

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6055 -0.3932 -0.2643 -0.1384 2.9470   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.750397 1.318165 -2.845 0.00444 \*\*   
## malefemale 0.137577 0.411340 0.334 0.73803   
## raceother 1.143719 0.403890 2.832 0.00463 \*\*   
## age 0.005279 0.016910 0.312 0.75490   
## stateKentucky 0.124282 0.492370 0.252 0.80072   
## stateLousiana 0.217202 0.556154 0.391 0.69614   
## stateVirginia -3.801561 0.666733 -5.702 1.19e-08 \*\*\*  
## time.served -0.109344 0.118901 -0.920 0.35777   
## max.sentence 0.065956 0.054593 1.208 0.22700   
## multiple.offenses 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimedrug-related 0.516479 0.739095 0.699 0.48468   
## crimeOther 0.727043 0.690775 1.053 0.29257   
## crimelarceny 1.119953 0.797552 1.404 0.16025   
## ---  
## 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: 242.09 on 460 degrees of freedom  
## AIC: 268.09  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(violator~1, train, family="binomial")  
summary(emptymod) #aic of 342.04

##   
## 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

Forward stepwise

forwardmod = stepAIC(emptymod, direction = "forward", scope=list(upper=allmod, lower=emptymod), trace = TRUE)

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 275.18 283.18  
## + max.sentence 1 331.01 335.01  
## + multiple.offenses 1 335.02 339.02  
## + race 1 336.51 340.51  
## + time.served 1 336.61 340.61  
## <none> 340.04 342.04  
## + crime 3 335.07 343.07  
## + male 1 339.72 343.72  
## + age 1 339.95 343.95  
##   
## Step: AIC=283.18  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 254.96 264.96  
## + race 1 267.66 277.66  
## <none> 275.18 283.18  
## + max.sentence 1 274.27 284.27  
## + time.served 1 274.44 284.44  
## + age 1 275.11 285.11  
## + male 1 275.13 285.13  
## + crime 3 271.72 285.72  
##   
## Step: AIC=264.96  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 246.98 258.98  
## <none> 254.96 264.96  
## + max.sentence 1 253.11 265.11  
## + time.served 1 254.47 266.47  
## + male 1 254.91 266.91  
## + age 1 254.94 266.94  
## + crime 3 252.75 268.75  
##   
## Step: AIC=258.98  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## + max.sentence 1 245.31 259.31  
## + time.served 1 246.33 260.33  
## + male 1 246.78 260.78  
## + age 1 246.98 260.98  
## + crime 3 244.78 262.79

summary(forwardmod) # AIC: 258.98

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLousiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceother 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

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

## Start: AIC=268.09  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - crime 3 244.47 264.47  
## - age 1 242.18 266.18  
## - male 1 242.20 266.20  
## - time.served 1 242.93 266.93  
## - max.sentence 1 243.57 267.57  
## <none> 242.09 268.09  
## - race 1 250.24 274.24  
## - multiple.offenses 1 261.96 285.96  
## - state 3 316.24 336.24  
##   
## Step: AIC=264.47  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - age 1 244.48 262.48  
## - male 1 244.85 262.85  
## - time.served 1 245.04 263.04  
## - max.sentence 1 246.00 264.00  
## <none> 244.47 264.47  
## - race 1 252.62 270.62  
## - multiple.offenses 1 265.46 283.46  
## - state 3 321.69 335.69  
##   
## Step: AIC=262.48  
## violator ~ male + race + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 244.86 260.86  
## - time.served 1 245.04 261.04  
## - max.sentence 1 246.01 262.01  
## <none> 244.48 262.48  
## - race 1 252.65 268.65  
## - multiple.offenses 1 265.52 281.52  
## - state 3 322.14 334.14  
##   
## Step: AIC=260.86  
## violator ~ race + state + time.served + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - time.served 1 245.31 259.31  
## - max.sentence 1 246.33 260.33  
## <none> 244.86 260.86  
## - race 1 252.80 266.80  
## - multiple.offenses 1 265.93 279.93  
## - state 3 322.54 332.54  
##   
## Step: AIC=259.31  
## violator ~ race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - max.sentence 1 246.98 258.98  
## <none> 245.31 259.31  
## - race 1 253.11 265.11  
## - multiple.offenses 1 266.89 278.89  
## - state 3 323.88 331.88  
##   
## Step: AIC=258.98  
## violator ~ race + state + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## - race 1 254.96 264.96  
## - multiple.offenses 1 267.66 277.66  
## - state 3 332.93 338.93

summary(backwardmod) # AIC: 258.98

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## raceother 1.09382 0.38974 2.807 0.00501 \*\*   
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLousiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

Custom Mod. Why not? After a lot of variations.. I ended up with a 260.33 AIC with my custom model. The forward/backward stepwise is the best model.

The forward/backward model identifies raceother, stateVirginia, and multiple.offenses as the most significant variables in the model. The AIC ends up being 258.98.

custom\_mod = glm(formula = violator ~ time.served + multiple.offenses + state + race, family = "binomial",   
 data = train)  
  
summary(custom\_mod)

##   
## Call:  
## glm(formula = violator ~ time.served + multiple.offenses + state +   
## race, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4877 -0.4035 -0.2694 -0.1486 3.0160   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.11211 0.60342 -3.500 0.000465 \*\*\*  
## time.served -0.09311 0.11521 -0.808 0.418999   
## multiple.offenses 1.72361 0.39528 4.361 1.30e-05 \*\*\*  
## stateKentucky 0.11020 0.46362 0.238 0.812124   
## stateLousiana 0.03136 0.51018 0.061 0.950984   
## stateVirginia -3.62745 0.64038 -5.664 1.47e-08 \*\*\*  
## raceother 1.10892 0.39155 2.832 0.004624 \*\*   
## ---  
## 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: 246.33 on 466 degrees of freedom  
## AIC: 260.33  
##   
## Number of Fisher Scoring iterations: 6

parolee1 = data.frame(state = "Lousiana", multiple.offenses = 1, race = "white" )  
predict(forwardmod, parolee1, type="response")

## 1   
## 0.3379961

The predicted probability for a parole violation for parolee1 is .34

parolee2 = data.frame(state = "Kentucky", multiple.offenses = 0, race = "other" )  
predict(forwardmod, parolee2, type="response")

## 1   
## 0.2069629

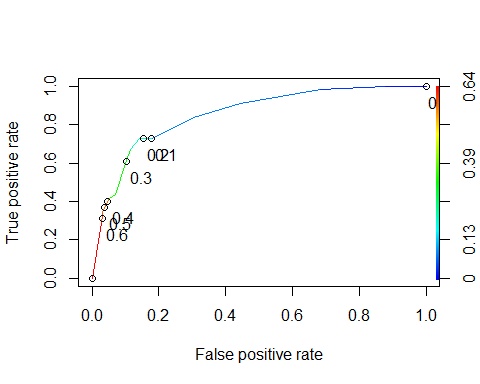
The predicted probability for a parole violation for parolee2 is .21

predictions = predict(forwardmod, type="response")  
head(predictions)

## 1 2 3 4 5 6   
## 0.07509978 0.19512504 0.19512504 0.07509978 0.07509978 0.19512504

Threshold select/ ROC curve

ROCRpred = prediction(predictions, train$violator)  
  
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))



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 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

t1 = table(train$violator, predictions > 0.2069629)  
t1

##   
## FALSE TRUE  
## no-violation 359 59  
## violated-parole 15 40

(t1[1,1]+t1[2,2]/nrow(train)) # this didn't work? so i had to do a manual calculation as you can see below

## [1] 359.0846

train\_accuracy = function(){  
 x = t1[1,1]  
 y = t1[2,2]  
 top = x +y  
 bottom = nrow(train)  
 accuracy = top/bottom  
 print(accuracy)  
}  
  
train\_accuracy()

## [1] 0.8435518

Task 8: Based on the above analysis, the sensitivity, specificity, and accuracy are: sensitivity: 0.7272727 specificity: 0.8588517 accuracy: 0.8435518

Incorrectly classifying someone as a potential parole violator puts them at risk of being jailed/imprisoned for longer than they should be. These people have hopefully been fully rehabilitated and are working towards becoming better versions of themselves.

t1 = table(train$violator, predictions > 0.1)  
t1

##   
## FALSE TRUE  
## no-violation 350 68  
## violated-parole 15 40

train\_accuracy()

## [1] 0.8245243

Task 9: Thresholds of 0.5 and 0.35 have an accuracy of .896 where as a threshold of 0.1 has an accuracy of .825

naive\_accuracy = function(){  
 x = t1[1]  
 top = x  
 bottom = nrow(train)  
 accuracy = top/bottom  
 print(accuracy)  
}  
  
t1 = table(train$violator, predictions > 1)  
t1

##   
## FALSE  
## no-violation 418  
## violated-parole 55

naive\_accuracy()

## [1] 0.8837209

naive has an accuracy of .88

Task 10: I am choosing the threshold of 0.35. The accuracy when applied to the testing data is .90

test\_accuracy = function(){  
 x = t1\_test[1,1]  
 y = t1\_test[2,2]  
 top = x +y  
 bottom = nrow(test)  
 accuracy = top/bottom  
 print(accuracy)  
}  
predict\_test = predict(forwardmod, newdata = test, type= "response")  
  
t1\_test = table(test$violator, predict\_test > 0.35)  
test\_accuracy()

## [1] 0.9009901