# Assignment 3.2 Logistic Regression (Classification)

## Module 3

### Task 0

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 2.0.1 v dplyr 0.7.8  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.3.1 v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.5.2

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

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

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

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

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

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

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

## -- 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 3.5.2

##   
## 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("parole.csv")

parole <- parole %>%  
 mutate(male = as.factor(male)) %>%  
 mutate(male = fct\_recode(male, "male" = "1", "female" = "0")) %>%  
 mutate(race = as.factor(race)) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "otherwise" = "2")) %>%  
 mutate(state = as.factor(state)) %>%  
 mutate(state = fct\_recode(state, "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4", "other" = "1")) %>%  
 mutate(crime = as.factor(crime)) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drugrelated" ="3", "drivingrelated" = "4", "other" = "1")) %>%  
 mutate(multiple.offenses = as.factor(multiple.offenses)) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "multipleoffenses" = "1", "otherwise" = "0")) %>%  
 mutate(violator = as.factor(violator)) %>%  
 mutate(violator = fct\_recode(violator, "violation" = "1", "noviolation" = "0"))

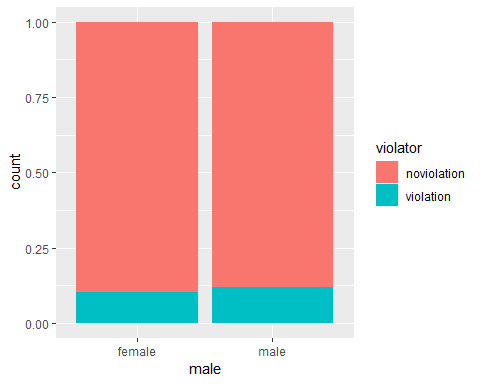
## Warning: package 'bindrcpp' was built under R version 3.5.2

### Task 1

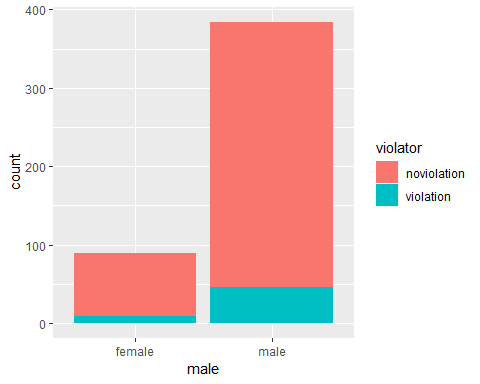
set.seed(12345)  
train.rows = createDataPartition(y=parole$violator, p=0.7, list=FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

### Task 2

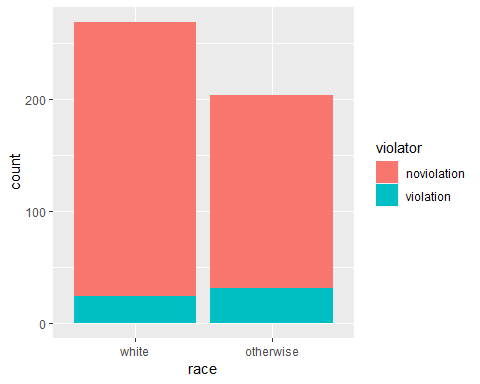
ggplot(train, aes(x=male, fill = violator)) +  
 geom\_bar(position = "fill")



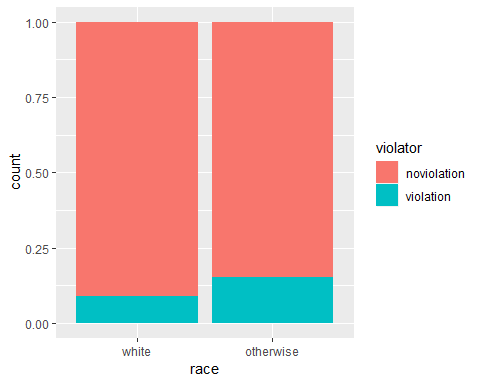
ggplot(train, aes(x=male, fill = violator)) +  
 geom\_bar()

 ####Comparing these two graphs above, we are using whether the inmate was male or female and who was a violator and who wasn’t. When seeing the first graph, we see them as percentages and the second graph we see them as count.

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

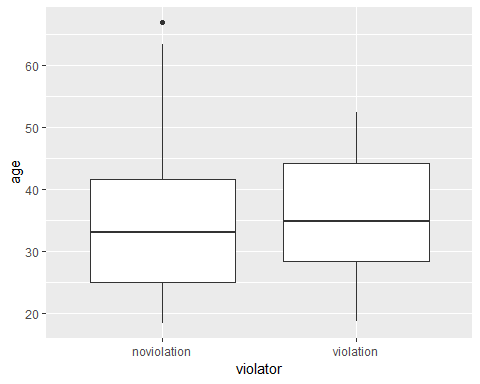


ggplot(train, aes(x=race, fill = violator)) +  
 geom\_bar(position = "fill")

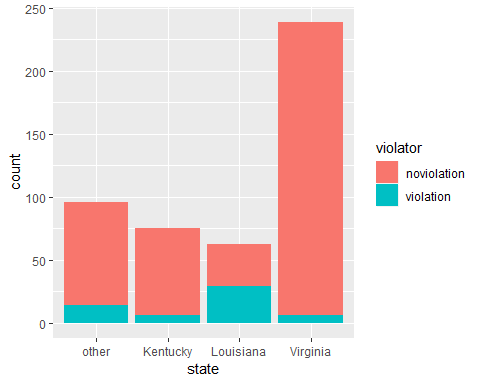


#### We can see with this graph that there is a higher percentage of nonwhites that violated their parole.

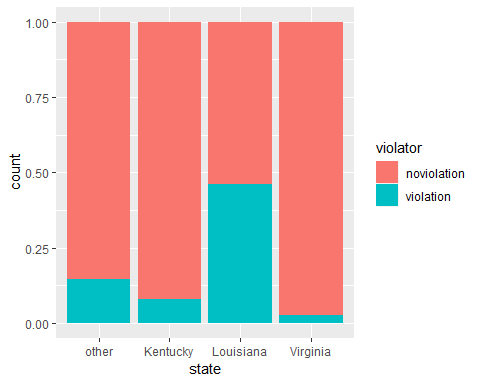
ggplot(train, aes(x=violator, y=age)) +  
 geom\_boxplot()



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

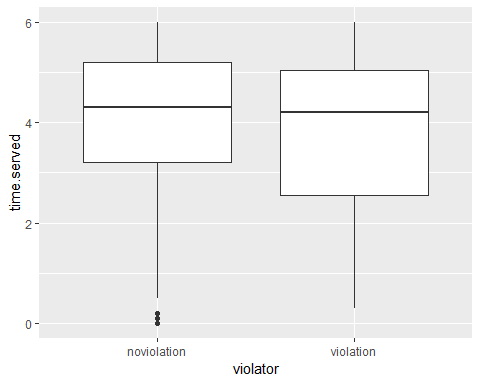


ggplot(train, aes(x=state, fill = violator)) +  
 geom\_bar(position = "fill")

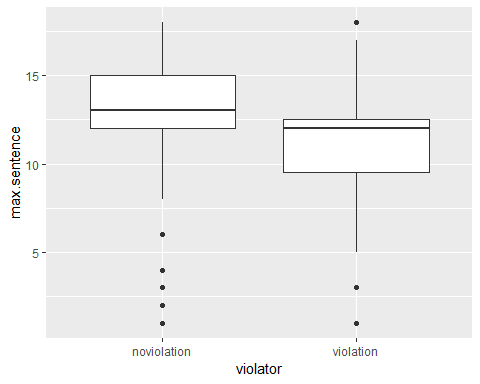


#### We can see here that Louisiana has a higher percentage rate of violators than all the states listed. We see that Virginia has many counts of released criminals but has the lowest percentage of violators.

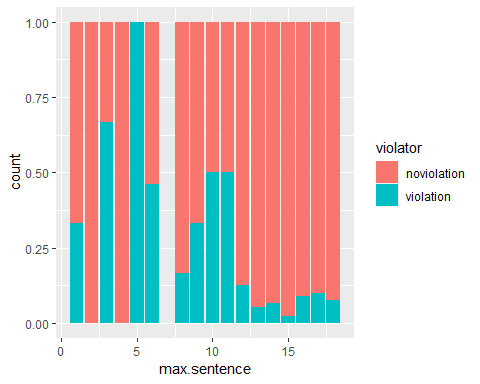
ggplot(train, aes(x=violator, y = time.served)) +  
 geom\_boxplot()



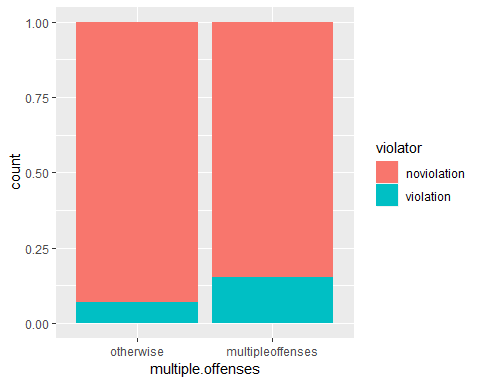
ggplot(train, aes(x=violator, y=max.sentence)) +  
 geom\_boxplot()



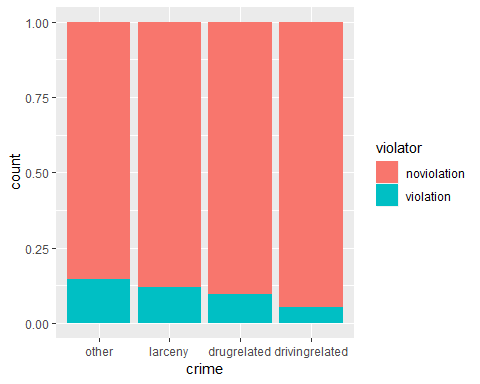
ggplot(train, aes(x=max.sentence, fill = violator)) +  
 geom\_bar(position= "fill")



ggplot(train, aes(x=multiple.offenses, fill = violator)) +  
 geom\_bar(position = "fill")



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



### Task 3

mod1 = glm(violator ~ crime, train, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ crime, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5621 -0.5621 -0.5021 -0.3311 2.4212   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.7654 0.1913 -9.228 <2e-16 \*\*\*  
## crimelarceny -0.2421 0.4033 -0.600 0.5483   
## crimedrugrelated -0.4646 0.3839 -1.210 0.2261   
## crimedrivingrelated -1.1110 0.5483 -2.026 0.0427 \*   
## ---  
## 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: 334.34 on 469 degrees of freedom  
## AIC: 342.34  
##   
## Number of Fisher Scoring iterations: 5

#### Here we see the AIC is 342.34, which is okay.

### Task 4

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

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

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 264.58 272.58  
## + max.sentence 1 321.79 325.79  
## + multiple.offenses 1 331.50 335.50  
## + race 1 335.64 339.64  
## + time.served 1 336.02 340.02  
## <none> 340.04 342.04  
## + age 1 338.27 342.27  
## + crime 3 334.34 342.34  
## + male 1 339.78 343.78  
##   
## Step: AIC=272.58  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 246.88 256.88  
## + race 1 259.14 269.14  
## + age 1 262.48 272.48  
## <none> 264.58 272.58  
## + crime 3 259.43 273.43  
## + male 1 263.58 273.58  
## + time.served 1 264.29 274.29  
## + max.sentence 1 264.49 274.49  
##   
## Step: AIC=256.88  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 240.42 252.42  
## <none> 246.88 256.88  
## + age 1 245.01 257.01  
## + max.sentence 1 245.58 257.58  
## + male 1 246.13 258.13  
## + time.served 1 246.88 258.88  
## + crime 3 242.93 258.93  
##   
## Step: AIC=252.42  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## + age 1 238.31 252.31  
## <none> 240.42 252.42  
## + max.sentence 1 238.81 252.81  
## + male 1 239.85 253.85  
## + time.served 1 240.37 254.37  
## + crime 3 236.69 254.69  
##   
## Step: AIC=252.31  
## violator ~ state + multiple.offenses + race + age  
##   
## Df Deviance AIC  
## + max.sentence 1 236.28 252.28  
## <none> 238.31 252.31  
## + male 1 237.41 253.41  
## + crime 3 233.88 253.88  
## + time.served 1 238.18 254.18  
##   
## Step: AIC=252.28  
## violator ~ state + multiple.offenses + race + age + max.sentence  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## + male 1 235.38 253.38  
## + crime 3 231.56 253.56  
## + time.served 1 236.12 254.12

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race + age +   
## max.sentence, 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 \*\*\*  
## stateKentucky -0.41360 0.54930 -0.753 0.45147   
## stateLouisiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## multiple.offensesmultipleoffenses 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## raceotherwise 1.07386 0.40527 2.650 0.00806 \*\*   
## age 0.02636 0.01660 1.588 0.11224   
## max.sentence 0.07733 0.05475 1.412 0.15788   
## ---  
## 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

backwardmod = 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(backwardmod)

##   
## 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   
## stateLouisiana 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.offensesmultipleoffenses 1.77974 0.41476 4.291 1.78e-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: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

#### Here we see that the AIC is better in both models at 252.28 which is better than the other model.

### Task 5

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

##   
## Call:  
## glm(formula = violator ~ state + race + multiple.offenses, 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   
## stateLouisiana 0.5292 0.4769 1.110 0.2672   
## stateVirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## raceotherwise 1.0024 0.3966 2.528 0.0115 \*   
## multiple.offensesmultipleoffenses 1.6596 0.3985 4.165 3.12e-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: 240.42 on 467 degrees of freedom  
## AIC: 252.42  
##   
## Number of Fisher Scoring iterations: 6

#### This model has a AIC just like the other model we just did with an AIC of 252.42. It’s .14 more from the other model.

### Task 6

#### Parolee 1: Probability of a parolee that lives in Louisiana with multiple offenses whom is white.

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

## 1   
## 0.408682

#### Parolee 2: Probability of a parolee that livs in Kentucky with no multiple offenses and is a race other than white.

newdata1 = data.frame(state = "Kentucky", race = "otherwise", multiple.offenses = "otherwise")  
predict(mod2, newdata1, type = "response")

## 1   
## 0.1153326

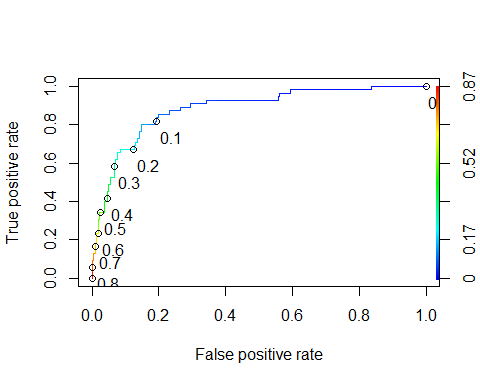
### Task 7

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

## 1 3 4 5 6 7   
## 0.05419344 0.08988548 0.08702778 0.14077504 0.20523702 0.05540271

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))



### Task 8

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 0.85454545  
## specificity 0.80143541  
## cutoff 0.09736171

t1= table(train$violator, predictions > .09736171)  
t1

##   
## FALSE TRUE  
## noviolation 335 83  
## violation 8 47

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

## [1] 0.1162791

### Task 9/10

t2 = table(train$violator, predictions > 0.5)  
t2

##   
## FALSE TRUE  
## noviolation 409 9  
## violation 36 19

(t2[1+1]+t2[2,2])/nrow(train)

## [1] 0.1162791

t3 = table(train$violator, predictions > 0.6)  
t3

##   
## FALSE TRUE  
## noviolation 411 7  
## violation 42 13

(t3[1+1]+t3[2,2])/nrow(train)

## [1] 0.1162791