## Classification & Regression

## Adam Samstag

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

library(readr)  
parole <- read\_csv("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()  
## )

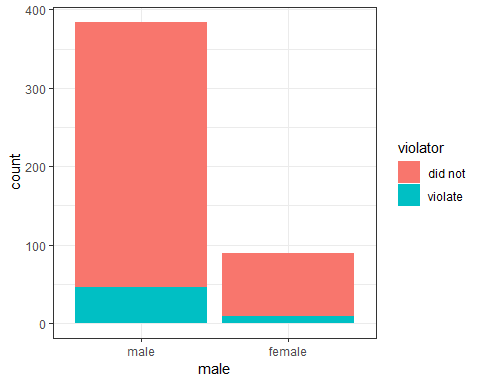
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>% mutate(male = fct\_recode(male, "male" = "1", "female" = "0"))   
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>% mutate(race = fct\_recode(race, "Other" = "2", "White" = "1"))  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>% mutate(state = fct\_recode(state, "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4", "Other" = "1"))  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>% mutate(crime = fct\_recode(crime, "larceny" = "2", "drug-related" = "3", "driving-related" = "4", "other" = "1"))  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>% mutate(multiple.offenses = fct\_recode(multiple.offenses, "SingleOffense" = "0", "MultipleOffense" = "1"))  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>% mutate(violator = fct\_recode(violator, "did not" = "0", "violate" = "1"))

#### Task 1

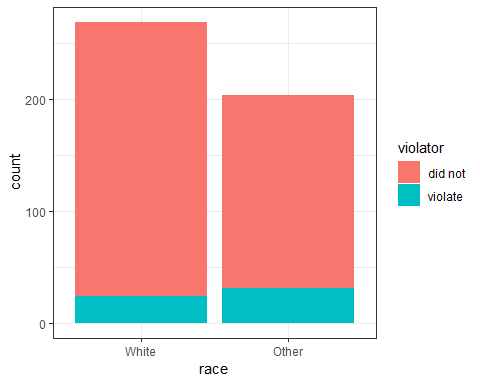
set.seed(12345)  
train.rows = createDataPartition(y= parole$violator, p=.7, list = FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

#### Task 2

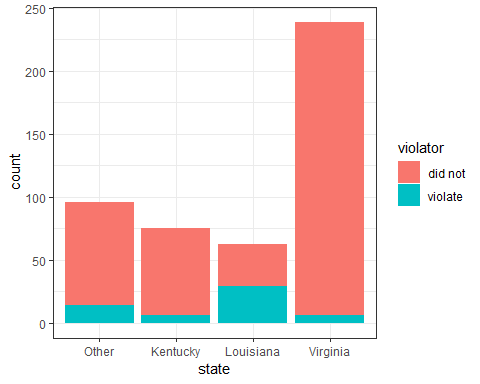
ggplot(train, aes(x=male, fill = violator)) + geom\_bar() + theme\_bw()



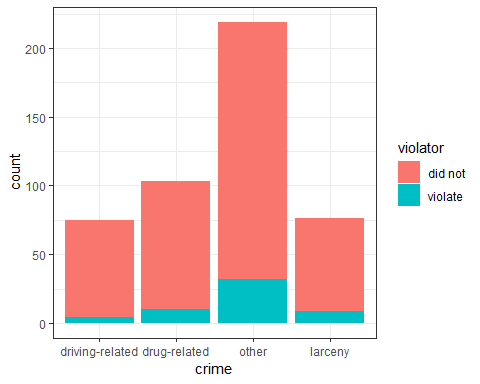
ggplot(train, aes(x=race, fill = violator)) + geom\_bar() + theme\_bw()



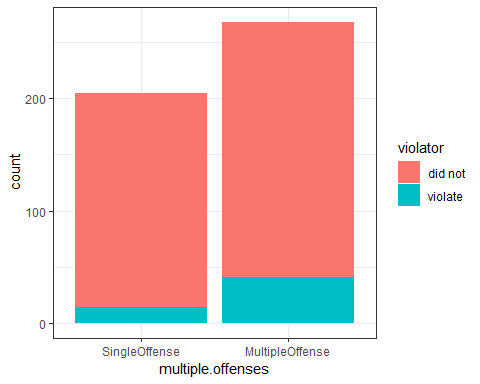
ggplot(train, aes(x=state, fill = violator)) + geom\_bar() + theme\_bw()



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



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



Based on these visualizations you can see that male and race are not good predictors for parole violation however, state, crime and multiple.offenses are good predictors of parole violation or you can at least see a significant difference between at least one of the variables in each one.

#### Task 3

mod1 = glm(violator ~ multiple.offenses, parole, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5627 -0.5627 -0.4080 -0.4080 2.2483   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4441 0.2085 -11.722 < 2e-16 \*\*\*  
## multiple.offensesMultipleOffense 0.6810 0.2561 2.659 0.00783 \*\*   
## ---  
## 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: 475.81 on 673 degrees of freedom  
## AIC: 479.81  
##   
## Number of Fisher Scoring iterations: 5

Based on the visualization in Task 2 I think that multiple.offenses is the best predictor of parole violation. This could be considered a good model because it has small p-value and a low AIC.

#### 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

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

The best model contains: race, age, state, multiple.offenses and max.sentence. The most significant variables are state and multiple.offenses.

#### Task 5

mod3 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
summary(mod3)

##   
## 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   
## stateLouisiana 0.5292 0.4769 1.110 0.2672   
## stateVirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## multiple.offensesMultipleOffense 1.6596 0.3985 4.165 3.12e-05 \*\*\*  
## raceOther 1.0024 0.3966 2.528 0.0115 \*   
## ---  
## 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

In this model each variable has at least one factor that is a good predictor of a parole violator. Those factors include: the state of Virginia, other states and mulitple.offenses multipleoffense.

#### Task 6

parolee1 = data.frame(state = "Louisiana", multiple.offenses = "MultipleOffense", race = "White")  
predict(mod3, parolee1, type = "response")

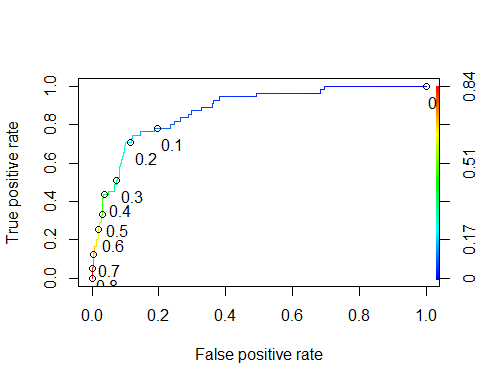
## 1   
## 0.408682

parolee2 = data.frame(state = "Kentucky", multiple.offenses = "SingleOffense", race = "Other")  
predict(mod3, parolee2, type = "response")

## 1   
## 0.1153326

#### Task 7

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

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.8755763

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

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

##   
## FALSE TRUE  
## did not 357 61  
## violate 14 41

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

##   
## FALSE TRUE  
## did not 357 61  
## violate 14 41

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

## [1] 0.8414376

#### Task 9

t2 = table(train$violator, predictions>0.4)  
t2

##   
## FALSE TRUE  
## did not 403 15  
## violate 31 24

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

## [1] 0.9027484

#### Task 10

t3 = table(train$violator, predictions>0.4)  
t3

##   
## FALSE TRUE  
## did not 403 15  
## violate 31 24

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

## [1] 2.113861