## Stuart Broach

## Logistic Regression

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)

## 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")

## 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,  
"white" = "1",  
"otherwise" = "2"))

parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4",  
"Other" = "1"))

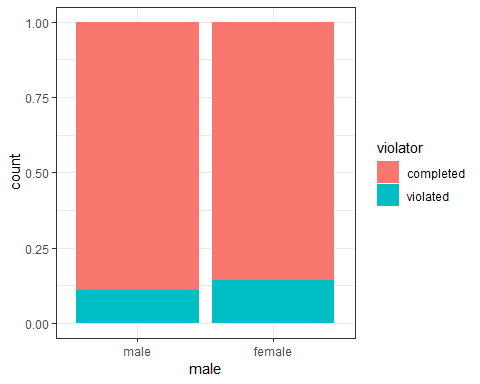
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
mutate(crime = fct\_recode(crime,  
"larceny" = "2",  
"drug-related" = "3",  
"driving-related" = "4",  
"other" = "1"))

parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"multiple-offenses" = "0",  
"single-offense" = "1"))

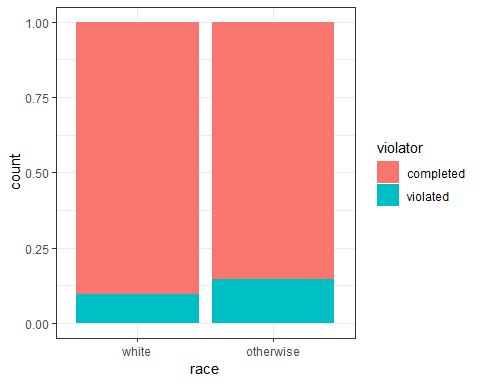
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"violated" = "1",  
"completed" = "0"))

set.seed(1234)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

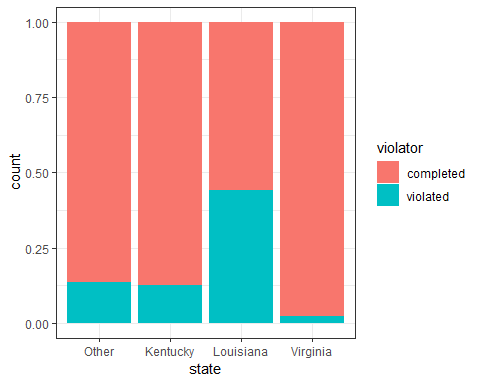
ggplot(train, aes(x=male, fill = violator)) + geom\_bar(position="fill") + theme\_bw()



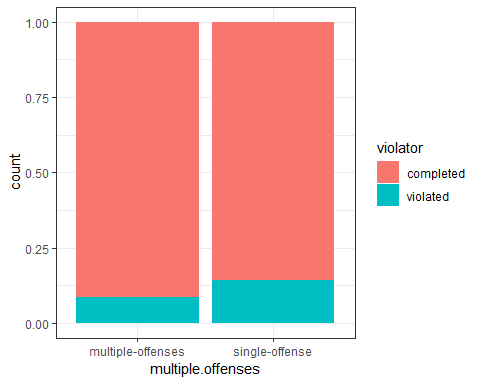
ggplot(train, aes(x=race, fill = violator)) + geom\_bar(position="fill") + theme\_bw()



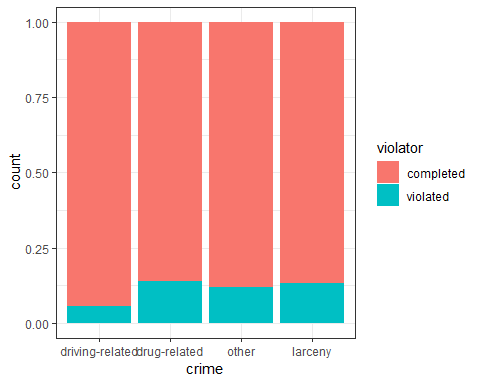
ggplot(train, aes(x=state, fill = violator)) + geom\_bar(position="fill") + theme\_bw()



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



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

 From the graphs, it shows that “State” is the largest predictor of violating parole.

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

##   
## Call:  
## glm(formula = violator ~ state, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0780 -0.5168 -0.2092 -0.2092 2.7687   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.8418 0.2985 -6.170 6.85e-10 \*\*\*  
## stateKentucky -0.1041 0.4393 -0.237 0.812625   
## stateLouisiana 1.6034 0.3973 4.035 5.46e-05 \*\*\*  
## stateVirginia -1.9693 0.5418 -3.635 0.000278 \*\*\*  
## ---  
## 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: 271.34 on 469 degrees of freedom  
## AIC: 279.34  
##   
## Number of Fisher Scoring iterations: 6

Louisiana has the highest coeficiant of the three. The AIC value is 279.34.

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

##   
## Call:  
## glm(formula = violator ~ male + race + state + multiple.offenses +   
## crime, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5603 -0.4177 -0.2565 -0.1640 2.8884   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.81478 0.59294 -4.747 2.06e-06 \*\*\*  
## malefemale 0.09574 0.40274 0.238 0.812090   
## raceotherwise 0.75798 0.38042 1.992 0.046318 \*   
## stateKentucky 0.02719 0.48663 0.056 0.955440   
## stateLouisiana 0.70968 0.47155 1.505 0.132327   
## stateVirginia -3.06153 0.62058 -4.933 8.08e-07 \*\*\*  
## multiple.offensessingle-offense 1.30104 0.38482 3.381 0.000722 \*\*\*  
## crimedrug-related 0.38446 0.65151 0.590 0.555124   
## crimeother 0.41937 0.60630 0.692 0.489127   
## crimelarceny 0.81648 0.70055 1.165 0.243823   
## ---  
## 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: 252.43 on 463 degrees of freedom  
## AIC: 272.43  
##   
## 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=272.43  
## violator ~ male + race + state + multiple.offenses + crime  
##   
## Df Deviance AIC  
## - crime 3 253.90 267.90  
## - male 1 252.49 270.49  
## <none> 252.43 272.43  
## - race 1 256.36 274.36  
## - multiple.offenses 1 264.29 282.29  
## - state 3 329.63 343.63  
##   
## Step: AIC=267.9  
## violator ~ male + race + state + multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 254.07 266.07  
## <none> 253.90 267.90  
## - race 1 257.96 269.96  
## - multiple.offenses 1 266.90 278.90  
## - state 3 333.90 341.90  
##   
## Step: AIC=266.07  
## violator ~ race + state + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 254.07 266.07  
## - race 1 258.03 268.03  
## - multiple.offenses 1 267.26 277.26  
## - state 3 334.69 340.69

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3667 -0.4317 -0.2643 -0.1829 2.8632   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4626 0.3646 -6.754 1.43e-11 \*\*\*  
## raceotherwise 0.7457 0.3726 2.001 0.045364 \*   
## stateKentucky 0.1362 0.4638 0.294 0.769013   
## stateLouisiana 0.7838 0.4678 1.675 0.093848 .   
## stateVirginia -2.9875 0.6113 -4.887 1.03e-06 \*\*\*  
## multiple.offensessingle-offense 1.3678 0.3830 3.571 0.000355 \*\*\*  
## ---  
## 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: 254.07 on 467 degrees of freedom  
## AIC: 266.07  
##   
## Number of Fisher Scoring iterations: 6

The AIC is down to 266 this time. Thie time, however, multiple.offenses(single-offense) had the highest coefficient with Louisiana coming in second. I don’t think that’s an accurate representation of the data model.

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   
## -1.3667 -0.4317 -0.2643 -0.1829 2.8632   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4626 0.3646 -6.754 1.43e-11 \*\*\*  
## stateKentucky 0.1362 0.4638 0.294 0.769013   
## stateLouisiana 0.7838 0.4678 1.675 0.093848 .   
## stateVirginia -2.9875 0.6113 -4.887 1.03e-06 \*\*\*  
## multiple.offensessingle-offense 1.3678 0.3830 3.571 0.000355 \*\*\*  
## raceotherwise 0.7457 0.3726 2.001 0.045364 \*   
## ---  
## 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: 254.07 on 467 degrees of freedom  
## AIC: 266.07  
##   
## Number of Fisher Scoring iterations: 6

This matches up with the stepwise model in that single offense is the highest coeficient with Louisiana coming in second. Same AIC.

parolee1 = data.frame(state = "Louisiana", race = "white", multiple.offenses = "multiple-offenses")  
predict(backmod, parolee1, type="response")

## 1   
## 0.157258

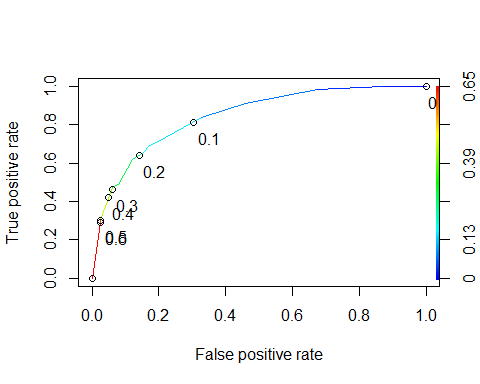
parolee2 = data.frame(state = "Kentucky", race = "otherwise", multiple.offenses = "single-offense")  
predict(backmod, parolee2, type="response")

## 1   
## 0.4469841

Parolee1 has a .15 chance of violating while parolee2 has a .44 chance.

test\_preds = predict(backmod, train, type = "response")

ROCRpred = prediction(test\_preds, 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))



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

## [1] 0.8433667

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]],  
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7090909  
## specificity 0.8062201  
## cutoff 0.1522745  
## cutoff 0.1522745

The senstivity is .70, the specificity is .80.

#t1 = table(train$violator,prediction > 0.1522745)  
#t1