# David Wells

## Module 3 Assignment 2

### Read-in Data and Convert

parole <- read\_csv("C:/Users/wdavi/Documents/BAN 502/Module 3/parole.csv")

##   
## -- 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()  
## )

parole = parole %>% mutate(male = as\_factor(male)) %>%   
 mutate(male = fct\_recode(male, "Male" = "0", "Female" = "1" ))   
  
parole = parole %>% mutate(race = as\_factor(race)) %>%   
 mutate(race = fct\_recode(race, "White" = "1", "Other" = "2" ))   
  
parole = parole %>% mutate(state = as\_factor(state)) %>%   
 mutate(state = fct\_recode(state, "Other State" = "1", "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4" ))   
  
parole = parole %>% mutate(multiple.offenses = as\_factor(multiple.offenses)) %>%   
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "No" = "0", "Yes" = "1" ))  
  
parole = parole %>% mutate(crime = as\_factor(crime)) %>%   
 mutate(crime = fct\_recode(crime, "Other Crime" = "1", "Larceny" = "2", "Drug-related" = "3", "Driving-related" = "4" ))  
  
parole = parole %>% mutate(violator = as\_factor(violator)) %>%   
 mutate(violator = fct\_recode(violator, "Completed Parole" = "0", "Violated Parole" = "1" ))   
  
str(parole)

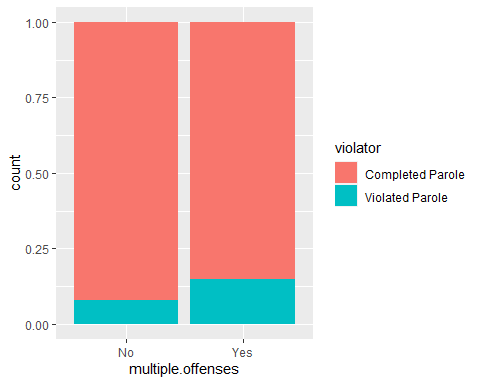
## spec\_tbl\_df [675 x 9] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ male : Factor w/ 2 levels "Male","Female": 2 1 2 2 2 2 2 1 1 2 ...  
## $ race : Factor w/ 2 levels "White","Other": 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num [1:675] 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ state : Factor w/ 4 levels "Other State",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time.served : num [1:675] 5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...  
## $ max.sentence : num [1:675] 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "Other Crime",..: 4 3 3 1 1 4 3 1 3 2 ...  
## $ violator : Factor w/ 2 levels "Completed Parole",..: 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "spec")=  
## .. 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()  
## .. )

### Task 1

set.seed(12345)   
parole\_split = initial\_split(parole, prob = 0.70, strata = violator)  
train = training(parole\_split)  
test = testing(parole\_split)

### Task 2

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



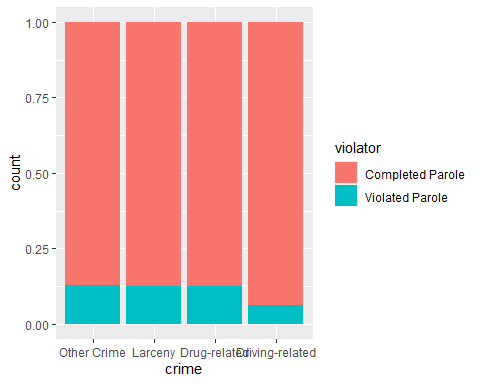
t1 = table(parole$violator,parole$multiple.offenses)  
prop.table(t1, margin = 2)

##   
## No Yes  
## Completed Parole 0.9201278 0.8535912  
## Violated Parole 0.0798722 0.1464088

t1

##   
## No Yes  
## Completed Parole 288 309  
## Violated Parole 25 53

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



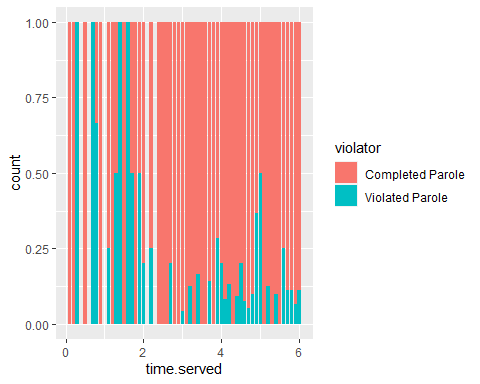
t2 = table(parole$violator,parole$crime)  
prop.table(t2, margin = 2)

##   
## Other Crime Larceny Drug-related Driving-related  
## Completed Parole 0.87619048 0.87735849 0.87581699 0.93069307  
## Violated Parole 0.12380952 0.12264151 0.12418301 0.06930693

t2

##   
## Other Crime Larceny Drug-related Driving-related  
## Completed Parole 276 93 134 94  
## Violated Parole 39 13 19 7

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



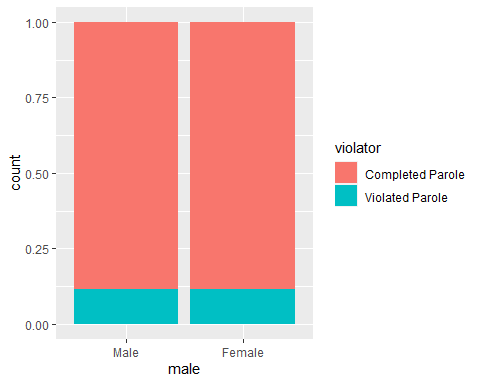
t3 = table(parole$violator,parole$time.served)  
prop.table(t3, margin = 2)

##   
## 0 0.1 0.2 0.3 0.5  
## Completed Parole 1.00000000 0.50000000 1.00000000 0.00000000 1.00000000  
## Violated Parole 0.00000000 0.50000000 0.00000000 1.00000000 0.00000000  
##   
## 0.7 0.8 0.9 1.1 1.2  
## Completed Parole 0.50000000 0.33333333 1.00000000 0.75000000 1.00000000  
## Violated Parole 0.50000000 0.66666667 0.00000000 0.25000000 0.00000000  
##   
## 1.3 1.4 1.5 1.6 1.7  
## Completed Parole 0.50000000 0.00000000 1.00000000 0.00000000 0.33333333  
## Violated Parole 0.50000000 1.00000000 0.00000000 1.00000000 0.66666667  
##   
## 1.8 1.9 2 2.1 2.2  
## Completed Parole 1.00000000 0.50000000 0.66666667 1.00000000 0.66666667  
## Violated Parole 0.00000000 0.50000000 0.33333333 0.00000000 0.33333333  
##   
## 2.3 2.4 2.5 2.6 2.7  
## Completed Parole 1.00000000 0.66666667 1.00000000 1.00000000 0.66666667  
## Violated Parole 0.00000000 0.33333333 0.00000000 0.00000000 0.33333333  
##   
## 2.8 2.9 3 3.1 3.2  
## Completed Parole 1.00000000 1.00000000 0.96721311 1.00000000 0.89473684  
## Violated Parole 0.00000000 0.00000000 0.03278689 0.00000000 0.10526316  
##   
## 3.3 3.4 3.5 3.6 3.7  
## Completed Parole 1.00000000 0.75000000 1.00000000 1.00000000 0.82352941  
## Violated Parole 0.00000000 0.25000000 0.00000000 0.00000000 0.17647059  
##   
## 3.8 3.9 4 4.1 4.2  
## Completed Parole 0.93333333 0.83333333 0.84210526 0.94736842 0.85714286  
## Violated Parole 0.06666667 0.16666667 0.15789474 0.05263158 0.14285714  
##   
## 4.3 4.4 4.5 4.6 4.7  
## Completed Parole 1.00000000 0.89655172 0.85185185 0.93333333 0.95454545  
## Violated Parole 0.00000000 0.10344828 0.14814815 0.06666667 0.04545455  
##   
## 4.8 4.9 5 5.1 5.2  
## Completed Parole 0.93750000 0.68181818 0.71428571 0.96666667 0.87500000  
## Violated Parole 0.06250000 0.31818182 0.28571429 0.03333333 0.12500000  
##   
## 5.3 5.4 5.5 5.6 5.7  
## Completed Parole 0.92307692 0.92307692 1.00000000 0.81250000 0.91666667  
## Violated Parole 0.07692308 0.07692308 0.00000000 0.18750000 0.08333333  
##   
## 5.8 5.9 6  
## Completed Parole 0.92307692 0.88888889 0.84615385  
## Violated Parole 0.07692308 0.11111111 0.15384615

t3

##   
## 0 0.1 0.2 0.3 0.5 0.7 0.8 0.9 1.1 1.2 1.3 1.4 1.5 1.6 1.7  
## Completed Parole 1 1 1 0 2 1 1 3 3 1 1 0 1 0 1  
## Violated Parole 0 1 0 1 0 1 2 0 1 0 1 2 0 1 2  
##   
## 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1 3.2  
## Completed Parole 3 1 4 1 4 1 2 1 1 4 1 11 59 18 17  
## Violated Parole 0 1 2 0 2 0 1 0 0 2 0 0 2 0 2  
##   
## 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4 4.1 4.2 4.3 4.4 4.5 4.6 4.7  
## Completed Parole 12 6 6 12 14 14 10 16 18 24 13 26 23 14 21  
## Violated Parole 0 2 0 0 3 1 2 3 1 4 0 3 4 1 1  
##   
## 4.8 4.9 5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6  
## Completed Parole 15 15 5 29 28 24 12 21 13 11 12 16 22  
## Violated Parole 1 7 2 1 4 2 1 0 3 1 1 2 4

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



t4 = table(parole$violator,parole$male)  
prop.table(t4, margin = 2)

##   
## Male Female  
## Completed Parole 0.8923077 0.8825688  
## Violated Parole 0.1076923 0.1174312

t4

##   
## Male Female  
## Completed Parole 116 481  
## Violated Parole 14 64

# My process involved thinking logically about what might affect parole violations. I first thought individuals with multiple offenses may be more willing to violate parole, as they've shown a willingness to commit multiple crimes in the first place. I then decided to look at whether the type of crime a person committed would affect parole violations...thinking those who committed a drug-related crime might be more likely to violate conditions of parole (by taking or using drugs, for example). I then decided to see how a person's time served would affect parole violations, thinking a person who spent more time in prison may be less willing to go back and therefore stick to the parole conditions. Finally, as a shot in the dark, I looked to see if gender had any affect on the probability of recidivism.

### Task 3

parole\_model =   
 logistic\_reg(mode = "classification") %>%   
 set\_engine("glm")   
  
parole\_recipe = recipe(violator ~ time.served, train)  
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit = fit(logreg\_wf, train)  
  
summary(parole\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7372 -0.5311 -0.4742 -0.4273 2.2536   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.1421 0.4382 -2.607 0.00915 \*\*  
## time.served -0.2192 0.1064 -2.061 0.03933 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 360.55 on 505 degrees of freedom  
## AIC: 364.55  
##   
## Number of Fisher Scoring iterations: 4

# Looking at the model data, I believe "time.served" to be a decent predictor of "violator". It appears from the model, the less time served the more likely one is to be a parole violator. AIC for this model is 364.55.

### Task 4

#Model 2  
parole\_model2 =   
 logistic\_reg(mode = "classification") %>%   
 set\_engine("glm")   
  
parole\_recipe2 = recipe(violator ~ time.served + crime, train)  
  
logreg\_wf2 = workflow() %>%  
 add\_recipe(parole\_recipe2) %>%   
 add\_model(parole\_model2)  
  
parole\_fit2 = fit(logreg\_wf2, train)  
  
summary(parole\_fit2$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7785 -0.5321 -0.4715 -0.4021 2.4070   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.015813 0.453638 -2.239 0.0251 \*  
## time.served -0.228301 0.105812 -2.158 0.0310 \*  
## crimeLarceny -0.005804 0.380922 -0.015 0.9878   
## crimeDrug-related 0.039345 0.351240 0.112 0.9108   
## crimeDriving-related -0.796945 0.504276 -1.580 0.1140   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 357.22 on 502 degrees of freedom  
## AIC: 367.22  
##   
## Number of Fisher Scoring iterations: 5

# Model 3  
parole\_model3 =   
 logistic\_reg(mode = "classification") %>%   
 set\_engine("glm")   
  
parole\_recipe3 = recipe(violator ~ time.served + crime + multiple.offenses, train)  
  
logreg\_wf3 = workflow() %>%  
 add\_recipe(parole\_recipe3) %>%   
 add\_model(parole\_model3)  
  
parole\_fit3 = fit(logreg\_wf3, train)  
  
summary(parole\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.8566 -0.5580 -0.4403 -0.3728 2.4686   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.61025 0.53840 -2.991 0.00278 \*\*  
## time.served -0.19069 0.10834 -1.760 0.07840 .   
## crimeLarceny -0.02551 0.38265 -0.067 0.94685   
## crimeDrug-related 0.21662 0.36284 0.597 0.55050   
## crimeDriving-related -0.77795 0.50538 -1.539 0.12372   
## multiple.offensesYes 0.67528 0.30924 2.184 0.02899 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 352.22 on 501 degrees of freedom  
## AIC: 364.22  
##   
## Number of Fisher Scoring iterations: 5

# From the model tests, it looks like Model 3 improves on my initial model, but only slightly at an AIC of 364.22 compared to my original Model 1 AIC of 364.55. I assume there is a better model here than what I have selected.

### Task 5

parole\_model4 =   
 logistic\_reg(mode = "classification") %>%   
 set\_engine("glm")   
  
parole\_recipe4 = recipe(violator ~ state + multiple.offenses + race, train)  
  
logreg\_wf4 = workflow() %>%  
 add\_recipe(parole\_recipe4) %>%   
 add\_model(parole\_model4)  
  
parole\_fit4 = fit(logreg\_wf4, train)  
  
summary(parole\_fit4$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2598 -0.4718 -0.2675 -0.2173 2.7414   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5431 0.3579 -7.106 1.20e-12 \*\*\*  
## stateKentucky 0.4036 0.4470 0.903 0.367   
## stateLouisiana 0.7135 0.4481 1.592 0.111   
## stateVirginia -2.7907 0.5570 -5.010 5.43e-07 \*\*\*  
## multiple.offensesYes 1.5998 0.3684 4.342 1.41e-05 \*\*\*  
## raceOther 0.4215 0.3527 1.195 0.232   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 277.99 on 501 degrees of freedom  
## AIC: 289.99  
##   
## Number of Fisher Scoring iterations: 6

# This model better fits the data than my previous models. The AIC for this model is 289.99, whereas my best model had an AIC of 364.22.   
  
# The variables "state" and "multiple.offenses" seem to be significant.

### Task 6

newdata1 = data.frame(state = "Louisiana", multiple.offenses = "Yes", race = "White" )  
predict(parole\_fit4, newdata1, type="prob")

## # A tibble: 1 x 2  
## `.pred\_Completed Parole` `.pred\_Violated Parole`  
## <dbl> <dbl>  
## 1 0.557 0.443

# Probability of parole violation = 0.44  
  
newdata2 = data.frame(state = "Kentucky", multiple.offenses = "No", race = "Other" )  
predict(parole\_fit4, newdata2, type="prob")

## # A tibble: 1 x 2  
## `.pred\_Completed Parole` `.pred\_Violated Parole`  
## <dbl> <dbl>  
## 1 0.848 0.152

# Probability of parole violation = 0.15

### Task 7

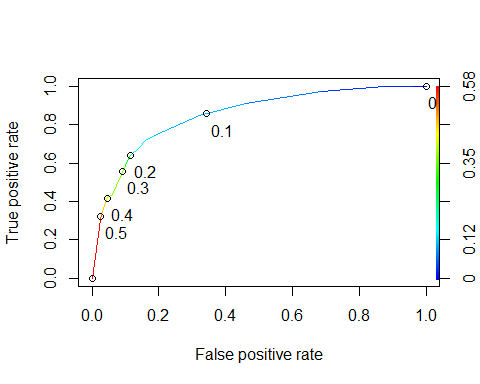
predictions = predict(parole\_fit4, parole, type="prob")  
head(predictions)

## # A tibble: 6 x 2  
## `.pred\_Completed Parole` `.pred\_Violated Parole`  
## <dbl> <dbl>  
## 1 0.927 0.0729  
## 2 0.927 0.0729  
## 3 0.893 0.107   
## 4 0.927 0.0729  
## 5 0.893 0.107   
## 6 0.893 0.107

predictions = predict(parole\_fit4, parole, type="prob")[2]  
head(predictions)

## # A tibble: 6 x 1  
## `.pred\_Violated Parole`  
## <dbl>  
## 1 0.0729  
## 2 0.0729  
## 3 0.107   
## 4 0.0729  
## 5 0.107   
## 6 0.107

ROCRpred = prediction(predictions, parole$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 0.7435897  
## specificity 0.8140704  
## cutoff 0.1070172

### Task 8

t5 = table(parole$violator,predictions > 0.1070172)  
  
t5

##   
## FALSE TRUE  
## Completed Parole 501 96  
## Violated Parole 22 56

# Accuracy  
(t5[1,1]+t5[2,2])/nrow(parole)

## [1] 0.8251852

# Sensitivity  
t5[2,2]/(t5[2,2]+t5[1,2])

## [1] 0.3684211

# Specificity  
t5[1,1]/(t5[1,1]+t5[2,1])

## [1] 0.957935

# Individuals who successfully completed parole may be categorized as not having complete parole.

### Task 9

t6 = table(parole$violator,predictions > 0.5)  
t6

##   
## FALSE TRUE  
## Completed Parole 582 15  
## Violated Parole 54 24

(t6[1,1]+t6[2,2])/nrow(parole)

## [1] 0.8977778

t6 = table(parole$violator,predictions > 0.6)  
t6

##   
## FALSE  
## Completed Parole 597  
## Violated Parole 78

(t6[1,1])/nrow(parole) # I got an error here where I had no TRUE values, so I removed t6[2,2] from the equation.

## [1] 0.8844444

t6 = table(parole$violator,predictions > 0.4)  
t6

##   
## FALSE TRUE  
## Completed Parole 571 26  
## Violated Parole 46 32

(t6[1,1]+t6[2,2])/nrow(parole)

## [1] 0.8933333

# I will pick >0.5

### Task 10

t6 = table(parole$violator,predictions > 0.5)  
t6

##   
## FALSE TRUE  
## Completed Parole 582 15  
## Violated Parole 54 24

(t6[1,1]+t6[2,2])/nrow(parole)

## [1] 0.8977778

# Accuracy  
(t6[1,1]+t6[2,2])/nrow(parole)

## [1] 0.8977778

# Sensitivity  
t6[2,2]/(t6[2,2]+t5[1,2])

## [1] 0.2

# Specificity  
t6[1,1]/(t6[1,1]+t5[2,1])

## [1] 0.9635762