## Module 3 Assignment 2

### Underwood, Katie

#install.packages("e1071")  
#install.packages("ROCR")  
library(tidyverse)

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

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

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

library(tidymodels)

## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom 0.7.2 v recipes 0.1.15  
## v dials 0.0.9 v rsample 0.0.8   
## v infer 0.5.4 v tune 0.1.2   
## v modeldata 0.1.0 v workflows 0.2.1   
## v parsnip 0.1.5 v yardstick 0.0.7

## -- Conflicts ----------------------------------------- tidymodels\_conflicts() --  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

library(e1071)

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:tune':  
##   
## tune

library(ROCR)

parole = read\_csv("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"= "1", "female" = "0"))  
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" = "1", "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4"))  
parole = parole %>% mutate(crime= as\_factor(crime)) %>%  
 mutate(crime = fct\_recode(crime, "other" = "1", "larceny" = "2", "drug" = "3", "drive" = "4"))  
parole = parole %>% mutate(multiple.offenses= as\_factor(multiple.offenses)) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "yes" = "1", "no" = "0"))  
parole = parole %>% mutate(violator= as\_factor(violator)) %>%  
 mutate(violator = fct\_recode(violator, "yes" = "1", "no" = "0"))

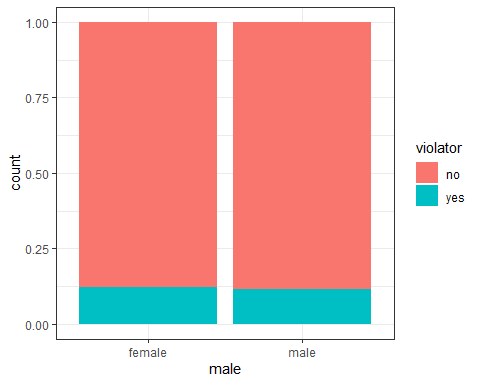
## Task 1

set.seed(12345)  
parole\_split = initial\_split(parole, prop = .7, strata = violator)  
train = training(parole\_split)  
test = testing(parole\_split)

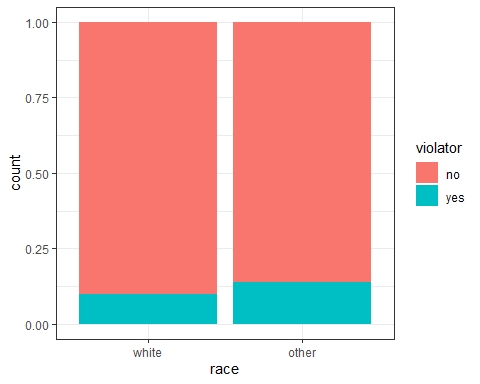
## Task 2

People in Louisiana and people with multiple offenses seem to violate parole more. There may be a weaker relationship between age and likelihood of violation (younger people violate more). People who have a driving offense seem to violate less than other types of crimes.

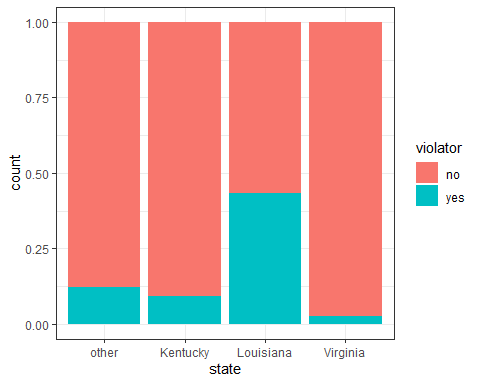
ggplot(train, aes(x=male, fill=violator)) + geom\_bar(position="fill") + theme\_bw()



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



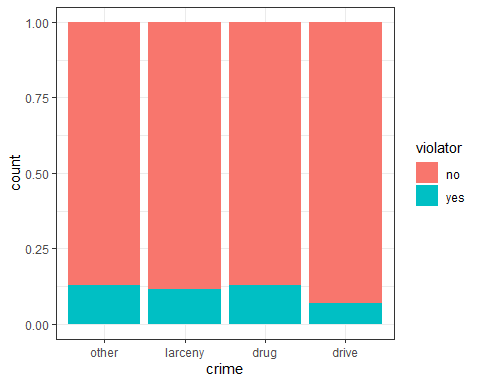
ggplot(train, aes(x=state, fill=violator)) + geom\_bar(position="fill") + theme\_bw()



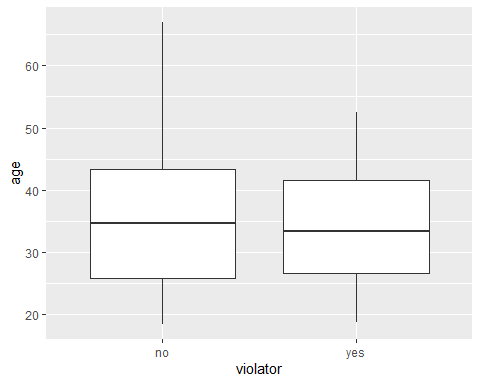
ggplot(train, aes(x=multiple.offenses, fill=violator)) + geom\_bar(position="fill") + theme\_bw()



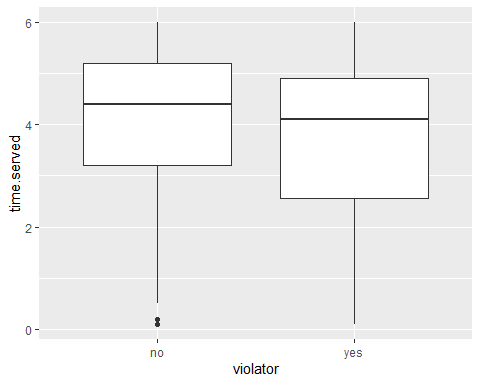
ggplot(train, aes(x=crime, fill=violator)) + geom\_bar(position="fill") + theme\_bw()



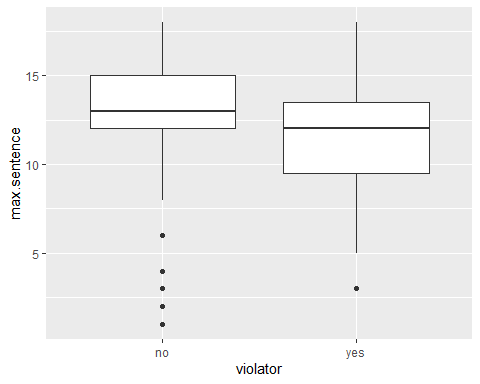
ggplot(train, aes(y=age, x=violator)) + geom\_boxplot()



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



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



## Task 3

I chose multiple.offenses to use as the variable to predict likelihood of violation. The p values are low for this model which means they are statistically significant. The AIC is 332 which is a little high (lower is better).

violator\_model = logistic\_reg() %>%  
 set\_engine ("glm")  
  
predict\_recipe = recipe(violator ~ multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(predict\_recipe) %>%  
 add\_model(violator\_model)  
  
violator\_fit = fit(logreg\_wf, train)  
  
summary(violator\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5911 -0.5911 -0.3566 -0.3566 2.3609   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.7233 0.2863 -9.512 < 2e-16 \*\*\*  
## multiple.offenses\_yes 1.0674 0.3322 3.213 0.00131 \*\*   
## ---  
## 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: 328.29 on 471 degrees of freedom  
## AIC: 332.29  
##   
## Number of Fisher Scoring iterations: 5

## Task 4

Based on AIC, state is actually the best indicator of violation with a value of 277 (compared to the 332 from multiple offenders above)

predict\_recipe2 = recipe(violator ~ state, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
logreg\_wf2 = workflow() %>%  
 add\_recipe(predict\_recipe2) %>%  
 add\_model(violator\_model)  
violator\_fit2 = fit(logreg\_wf2, train)  
summary(violator\_fit2$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0682 -0.5139 -0.2289 -0.2289 2.7037   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.9577 0.3084 -6.348 2.17e-10 \*\*\*  
## state\_Kentucky -0.3159 0.5027 -0.628 0.5298   
## state\_Louisiana 1.6954 0.3925 4.319 1.57e-05 \*\*\*  
## state\_Virginia -1.6710 0.5159 -3.239 0.0012 \*\*   
## ---  
## 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: 269.31 on 469 degrees of freedom  
## AIC: 277.31  
##   
## Number of Fisher Scoring iterations: 6

predict\_recipe3 = recipe(violator ~ male, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
logreg\_wf3 = workflow() %>%  
 add\_recipe(predict\_recipe3) %>%  
 add\_model(violator\_model)  
violator\_fit3 = fit(logreg\_wf3, train)  
summary(violator\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5067 -0.4952 -0.4952 -0.4952 2.0782   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.98787 0.33719 -5.895 3.74e-09 \*\*\*  
## male\_male -0.04901 0.37258 -0.132 0.895   
## ---  
## 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: 340.02 on 471 degrees of freedom  
## AIC: 344.02  
##   
## Number of Fisher Scoring iterations: 4

predict\_recipe4 = recipe(violator ~ race, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
logreg\_wf4 = workflow() %>%  
 add\_recipe(predict\_recipe4) %>%  
 add\_model(violator\_model)  
violator\_fit4 = fit(logreg\_wf4, train)  
summary(violator\_fit4$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5463 -0.5463 -0.4582 -0.4582 2.1477   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.2013 0.2028 -10.855 <2e-16 \*\*\*  
## race\_other 0.3745 0.2874 1.303 0.193   
## ---  
## 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: 338.34 on 471 degrees of freedom  
## AIC: 342.34  
##   
## Number of Fisher Scoring iterations: 4

predict\_recipe5 = recipe(violator ~ crime, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
logreg\_wf5 = workflow() %>%  
 add\_recipe(predict\_recipe5) %>%  
 add\_model(violator\_model)  
violator\_fit5 = fit(logreg\_wf5, train)  
summary(violator\_fit5$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5243 -0.5220 -0.5220 -0.3741 2.3215   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.924519 0.206016 -9.342 <2e-16 \*\*\*  
## crime\_larceny -0.112363 0.409935 -0.274 0.784   
## crime\_drug 0.009699 0.352700 0.028 0.978   
## crime\_drive -0.700150 0.506887 -1.381 0.167   
## ---  
## 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: 337.66 on 469 degrees of freedom  
## AIC: 345.66  
##   
## Number of Fisher Scoring iterations: 5

## Task 5

State and mutiple.offenses are significant variables, race is not a significant variable. The AIC value for this model is 345 which is actually a worse value than a model with state alone.

predict\_recipe6 = recipe(violator ~ race + state + multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
logreg\_wf6 = workflow() %>%  
 add\_recipe(predict\_recipe6) %>%  
 add\_model(violator\_model)  
violator\_fit6 = fit(logreg\_wf6, train)  
summary(violator\_fit6$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3372 -0.3599 -0.2693 -0.2372 2.6777   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.78822 0.40113 -6.951 3.63e-12 \*\*\*  
## race\_other 0.25841 0.38101 0.678 0.4976   
## state\_Kentucky 0.08362 0.53523 0.156 0.8759   
## state\_Louisiana 1.12294 0.46246 2.428 0.0152 \*   
## state\_Virginia -2.54372 0.56630 -4.492 7.06e-06 \*\*\*  
## multiple.offenses\_yes 1.77497 0.39210 4.527 5.99e-06 \*\*\*  
## ---  
## 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: 245.54 on 467 degrees of freedom  
## AIC: 257.54  
##   
## Number of Fisher Scoring iterations: 6

## Task 6

Parolee1 has a 52% chance of violating parole. Parolee2 has an 8% chance of violating parole.

new\_data = data.frame(state = "Louisiana", multiple.offenses = "yes", race = "white")  
predict(violator\_fit6, new\_data, type= "prob")

## # A tibble: 1 x 2  
## .pred\_no .pred\_yes  
## <dbl> <dbl>  
## 1 0.473 0.527

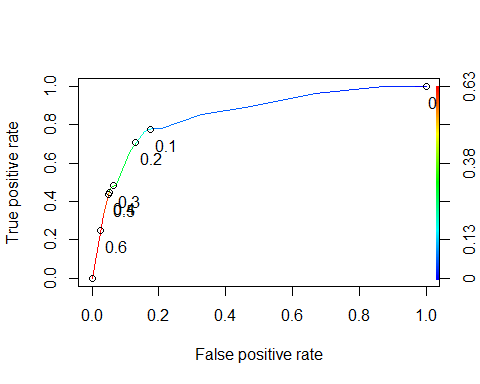
new\_data2 = data.frame(state = "Kentucky", multiple.offenses = "no", race = "other")  
predict(violator\_fit6, new\_data2, type= "prob")

## # A tibble: 1 x 2  
## .pred\_no .pred\_yes  
## <dbl> <dbl>  
## 1 0.920 0.0797

## Task 7

The optimal threshold is .16

predictions = predict(violator\_fit6, train, type= "prob")  
ROCRpred = prediction(predictions[,2], 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 0.7636364  
## specificity 0.8444976  
## cutoff 0.1590554

## Task 8

Sensitivity = .76, Specificity = .84, Accuracy = .85

t1 = table(train$violator, predictions[,2] > 0.1590554)  
t1

##   
## FALSE TRUE  
## no 363 55  
## yes 16 39

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

## [1] 0.8498943

## Task 9

Threshold of .32 will maximize accuracy (at .89 or 89%)

t2 = table(train$violator, predictions[,2] > 0.32)  
(t2[1,1]+t2[2,2])/nrow(train)

## [1] 0.8900634

## Task 10

The accuracy of the model on the testing set is .9 (or 90%) which is slightly higher than the accuracy on the training set.

predictions2 = predict(violator\_fit6, test, type= "prob")  
t3 = table(test$violator, predictions2[,2] > 0.32)  
(t3[1,1]+t3[2,2])/nrow(test)

## [1] 0.9009901