Week3-Assignment2

Engel, Alec

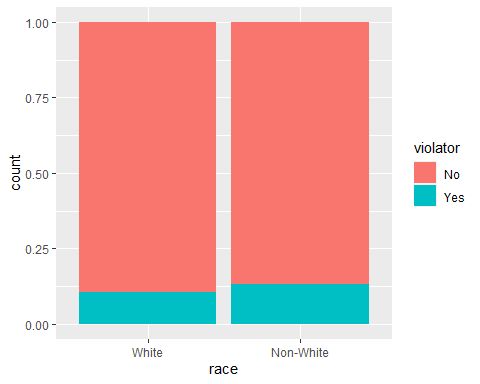
parole = read\_csv("parole.csv")

#str(parole)  
#summary(parole)

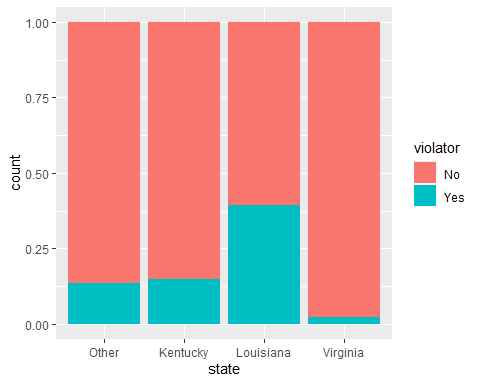
parole = parole %>%  
 mutate(male = as\_factor(male)) %>%  
 mutate(race = as\_factor(race)) %>%  
 mutate(state = as\_factor(state)) %>%  
 mutate(crime = as\_factor(crime)) %>%  
 mutate(multiple.offenses = as\_factor(multiple.offenses)) %>%  
 mutate(violator = as\_factor(violator)) %>%  
 mutate(male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

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

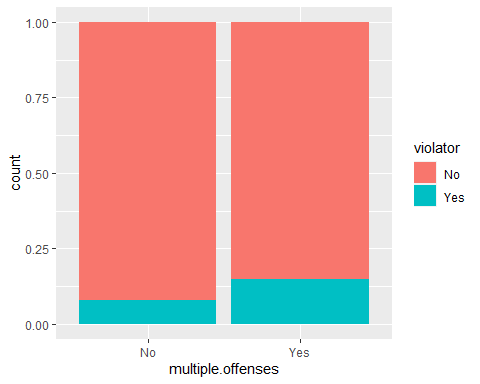
ggplot(train, aes(race, fill = violator)) +  
 geom\_bar(position="fill")



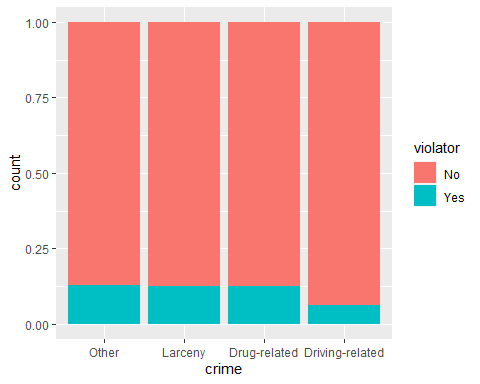
ggplot(train, aes(state, fill = violator)) +  
 geom\_bar(position = "fill")



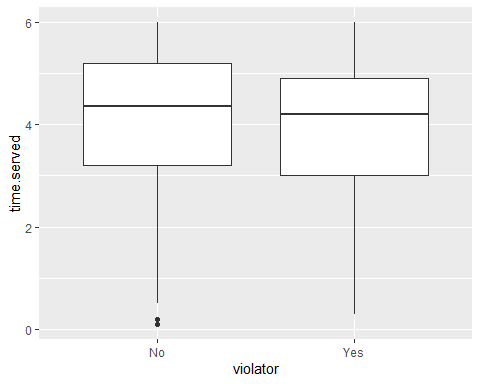
ggplot(train, aes(multiple.offenses, fill = violator)) +  
 geom\_bar(position = "fill")



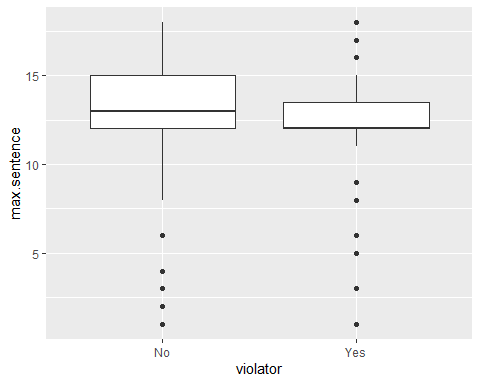
ggplot(train, aes(crime, fill = violator)) +  
 geom\_bar(position = "fill")



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



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



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

##   
## No Yes  
## No 0.4824121 0.3205128  
## Yes 0.5175879 0.6794872

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

##   
## No Yes  
## Other 0.20603015 0.25641026  
## Kentucky 0.17755444 0.17948718  
## Louisiana 0.07537688 0.47435897  
## Virginia 0.54103853 0.08974359

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

##   
## No Yes  
## Other 0.46231156 0.50000000  
## Larceny 0.15577889 0.16666667  
## Drug-related 0.22445561 0.24358974  
## Driving-related 0.15745394 0.08974359

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

##   
## No Yes  
## White 0.5896147 0.4743590  
## Non-White 0.4103853 0.5256410

**After comparing and contrasting various tables and visualizations, a parolees’ race, state, and if they were incarcerated for multiple offenses look to be most predictive of our response variable “violator”. Originally, I was thinking that a parolees’ age or possibly the crime they committed would be better predictors but based on the visualizations and tables, there wasn’t as strong of a correlation. I can see that White’s have a lower probability of violating parole while Louisiana seems to have the highest rate of parole violator states and Virgina the lowest. I believe the strongest predictive variable of a “violator” is whether or not the parolee has been incarcerated for multiple offenses or just one.**

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")  
  
parole\_recipe = recipe(violator ~ multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
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   
## -0.5663 -0.5663 -0.4088 -0.4088 2.2466   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4401 0.2392 -10.201 <2e-16 \*\*\*  
## multiple.offenses\_Yes 0.6909 0.2942 2.348 0.0189 \*   
## ---  
## 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: 358.85 on 505 degrees of freedom  
## AIC: 362.85  
##   
## Number of Fisher Scoring iterations: 5

**We see here an increase in the probability of violation of parole when multiple offenses have been committed. I also am observing what could be a relatively low AIC of 362.85 and a p-value less than .05 proving that this is a significant predictor variable.**

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")  
  
parole\_recipe = recipe(violator ~ state + multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit2 = fit(logreg\_wf, train)  
  
summary(parole\_fit2$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2000 -0.4952 -0.2460 -0.2460 2.6505   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4971 0.3565 -7.005 2.47e-12 \*\*\*  
## state\_Kentucky 0.4601 0.4451 1.034 0.3013   
## state\_Louisiana 0.9181 0.4114 2.231 0.0257 \*   
## state\_Virginia -2.6172 0.5332 -4.908 9.20e-07 \*\*\*  
## multiple.offenses\_Yes 1.6319 0.3663 4.456 8.37e-06 \*\*\*  
## ---  
## 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: 279.41 on 502 degrees of freedom  
## AIC: 289.41  
##   
## Number of Fisher Scoring iterations: 6

**Here I’ve determined the best fit model to be with the two most significant variables of the parolees’ state and if they have committed multiple offenses. While adding in many other variables, they did not appear to be significant in relation to being a violator. We are also seeing what appears to be a strong AIC of 289.41 which has dropped significantly from our original model with an AIC of 362.85. This model is also intuitive when it comes to determining in which states a parolee is likely to violate their parole. We can see that the base is “Other” and if a parolee is in Kentucky(even though this state is not significant based on it’s p-value) or Louisiana, there is a positive coefficient and the probability of violating parole goes up while a negative coefficient and the probability goes down significantly (-2.6172) if the parolee is in Virginia.**

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")  
  
parole\_recipe = recipe(violator ~ state + multiple.offenses + race, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit3 = fit(logreg\_wf, train)  
  
summary(parole\_fit3$fit$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 \*\*\*  
## state\_Kentucky 0.4036 0.4470 0.903 0.367   
## state\_Louisiana 0.7135 0.4481 1.592 0.111   
## state\_Virginia -2.7907 0.5570 -5.010 5.43e-07 \*\*\*  
## multiple.offenses\_Yes 1.5998 0.3684 4.342 1.41e-05 \*\*\*  
## race\_Non.White 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

**With this model, we have added race into the mix. I’ve noticed that Kentucky is still showing as not significant and Louisiana has now jumped into that insignificant predictor bucked with a p-value above .05 along with race of non-white parolees. This leads me to believe that states and race are predictive values that are correlated with each other showing multicollinearity. All other variables are significant and it looks like our AIC value slightly increased from 289.41 to 289.99 with this additional race predictor variable being added. We can also observe positive coefficients of parole violators in Kentucky, Louisiana, and with multiple offenses and non white parolees while we have negative coefficients in the state of Virginia.**

Parolee1 = data.frame(state = "Louisiana", multiple.offenses = "Yes", race = "White")  
predict(parole\_fit3, Parolee1, type="prob")

## # A tibble: 1 x 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.557 0.443

Parolee2 = data.frame(state = "Kentucky", multiple.offenses = "No", race = "Non-White")  
predict(parole\_fit3, Parolee2, type="prob")

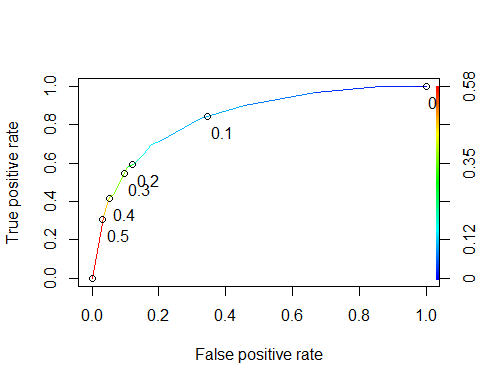
## # A tibble: 1 x 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.848 0.152

**Parolee1 = .4428**  
**Parolee2 = .1521**

predictions = predict(parole\_fit3, train, type="prob")[2]  
head(predictions)

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

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))



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

## [1] 0.834916

## [1] 0.8157421  
#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
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 0.7118644  
## specificity 0.7968750  
## cutoff 0.1070172

**Probability Threshold = 0.107**

t1 = table(train$violator,predictions > 0.1070172)  
t1

##   
## FALSE TRUE  
## No 368 80  
## Yes 18 41

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

## [1] 0.8067061

**accuracy = 0.8067**  
**specificity = 0.7969**  
**sensitivity = 0.7119**  
**In this training dataset, I see that we have incorrectly classified 18 parolees to not violate their parole when they did violate it. The implications here could be criminals being out on the loose with the possibility of committing another crime due to our mis-classification and possibly our lighter survalence because our predictions had us leading to believe this parolee would not violate.**

t2 = table(train$violator,predictions > 0.45)  
t2

##   
## FALSE TRUE  
## No 433 15  
## Yes 40 19

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

## [1] 0.8915187

**Threshold of 0.45 best maximizes accuracy on the training set with an accuracy of 0.8915.**

predictions2 = predict(parole\_fit3, test, type="prob")[2]  
head(predictions2)

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

t3 = table(test$violator,predictions2 > 0.45)  
t3

##   
## FALSE TRUE  
## No 148 1  
## Yes 12 7

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

## [1] 0.922619

**Accuracy of the model on the testing set is 0.9226.**