# Module 5 Assignment 1 Parameter Selection and Neural Networks Assignment

## Gregory Desloge

#install.packages("nnet")  
library("tidyverse")

## -- Attaching packages -------------------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.3.2   
## v tibble 2.1.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 v stringr 1.4.0   
## v readr 1.3.1 v forcats 0.4.0

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

library("caret")

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library("nnet")

parole = read.csv("parole.csv")  
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"female" = "0",  
"male" = "1"))  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"white" = "1",  
"other" = "2"))  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4",  
"Any Other State" = "1"))  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"One Offense" = "0",  
"Multiple Offenses" = "1"))  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"Violated Parole" = "1",  
"No Violations" = "0"))  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
mutate(crime = fct\_recode(crime,  
"larceny" = "2",  
"drug related" = "3",  
"driving relate" = "4",  
"other" = "1"))

set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p = 0.7, list = FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

fitControl = trainControl(method ="cv",  
 number = 10)  
  
nnetGrid <- expand.grid(size= 12, decay = 0.1)  
  
set.seed(1234)  
nnetBasic = train(violator ~.,  
 parole,  
 method = "nnet",  
 tuneGrid = nnetGrid,  
 trControl = fitControl,  
 verbose = FALSE,  
 trace = FALSE)

predNetBasic = predict(nnetBasic, train)  
confusionMatrix(predNetBasic, train$violator, positive="No Violations")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Violations Violated Parole  
## No Violations 409 24  
## Violated Parole 9 31  
##   
## Accuracy : 0.9302   
## 95% CI : (0.9034, 0.9515)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.0005254   
##   
## Kappa : 0.6149   
##   
## Mcnemar's Test P-Value : 0.0148061   
##   
## Sensitivity : 0.9785   
## Specificity : 0.5636   
## Pos Pred Value : 0.9446   
## Neg Pred Value : 0.7750   
## Prevalence : 0.8837   
## Detection Rate : 0.8647   
## Detection Prevalence : 0.9154   
## Balanced Accuracy : 0.7711   
##   
## 'Positive' Class : No Violations   
##

This model is a fairly good model. Accuracy is higher than the naive model, and based on the P-value it is statistically significantly better than the naive model as well. It has a very high sensitivity meanig it can predict No Violations very well. The specificity rate of 0.5636 isn’t great, meaning this model wasn’t very good in determinig who actually violated parole.

fitControl = trainControl(method ="cv",  
 number = 10)  
  
nnetGrid <- expand.grid(size= seq(from = 1, to =12, by = 1),   
 decay = seq(from = 0.1, to = 0.5, by = 0.1))  
  
set.seed(1234)  
nnetFit = train(violator ~.,  
 parole,  
 method = "nnet",  
 tuneGrid = nnetGrid,  
 trControl = fitControl,  
 verbose = FALSE,  
 trace = FALSE)

predNetFit = predict(nnetFit, train)  
confusionMatrix(predNetFit, train$violator, positive="No Violations")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Violations Violated Parole  
## No Violations 411 38  
## Violated Parole 7 17  
##   
## Accuracy : 0.9049   
## 95% CI : (0.8748, 0.9298)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.08373   
##   
## Kappa : 0.3871   
##   
## Mcnemar's Test P-Value : 7.744e-06   
##   
## Sensitivity : 0.9833   
## Specificity : 0.3091   
## Pos Pred Value : 0.9154   
## Neg Pred Value : 0.7083   
## Prevalence : 0.8837   
## Detection Rate : 0.8689   
## Detection Prevalence : 0.9493   
## Balanced Accuracy : 0.6462   
##   
## 'Positive' Class : No Violations   
##

This model isn’t bad, but not as good as the model from task 3 (nnetBasic). The accuracy has dropped and the p value has increased to the point where the model is no longer statiscially significantly better than the naive model. The sensitivity has just slightly increased, but specificity has decreased substantially. This indicates that this model is much worse at predicting whether a person will violate parole.

predNetBasic = predict(nnetBasic, test)  
confusionMatrix(predNetBasic, test$violator, positive="No Violations")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Violations Violated Parole  
## No Violations 178 16  
## Violated Parole 1 7  
##   
## Accuracy : 0.9158   
## 95% CI : (0.8687, 0.9502)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.108358   
##   
## Kappa : 0.4174   
##   
## Mcnemar's Test P-Value : 0.000685   
##   
## Sensitivity : 0.9944   
## Specificity : 0.3043   
## Pos Pred Value : 0.9175   
## Neg Pred Value : 0.8750   
## Prevalence : 0.8861   
## Detection Rate : 0.8812   
## Detection Prevalence : 0.9604   
## Balanced Accuracy : 0.6494   
##   
## 'Positive' Class : No Violations   
##

This model has a very high sensitivity, meaning it was extremely effective in determining if someone would not violate parole. The accuracy is slightly higher than the nnetFit train data, but slightly lower than the train data from netBasic. The specificity is much lower than that of the netBasic, meaning this model isn’t as good at predicting people who will violate parole.

predNetFit = predict(nnetFit, test)  
confusionMatrix(predNetFit, test$violator, positive="No Violations")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Violations Violated Parole  
## No Violations 176 20  
## Violated Parole 3 3  
##   
## Accuracy : 0.8861   
## 95% CI : (0.8341, 0.9264)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.5552509   
##   
## Kappa : 0.1677   
##   
## Mcnemar's Test P-Value : 0.0008492   
##   
## Sensitivity : 0.9832   
## Specificity : 0.1304   
## Pos Pred Value : 0.8980   
## Neg Pred Value : 0.5000   
## Prevalence : 0.8861   
## Detection Rate : 0.8713   
## Detection Prevalence : 0.9703   
## Balanced Accuracy : 0.5568   
##   
## 'Positive' Class : No Violations   
##

This model is not very good. The No information Rate and the accuracy are the exact same, meaning this model is as acccurate as a naive model. The sensitivity is high like all of the other models, but this has the lowest specificity of any of the models. This is the weakest model of the four.

Task 8: The model from Task two does not appear to be overfitting. The accuracy droped slightly but it remainder higher than that of the naive model. The model from Task 4 does appear to be overfitting, while the accuracy only dropped slightly, the test portion of the data had the same accuracy and no information rate which is not good. This could mean the model would not work well with new data.