## Stuart Broach

## Neural Networks

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

## Warning: package 'nnet' was built under R version 3.5.2

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

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

## Time difference of 2.205235 secs

nnetBasic

## Neural Network   
##   
## 473 samples  
## 8 predictor  
## 2 classes: 'completed', 'violated'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 427, 425, 426, 425, 425, 426, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8729321 0.2839906  
##   
## Tuning parameter 'size' was held constant at a value of 12  
##   
## Tuning parameter 'decay' was held constant at a value of 0.1

predNetBasic = predict(nnetBasic, train)

confusionMatrix(predNetBasic, train$violator, positive = "violated")

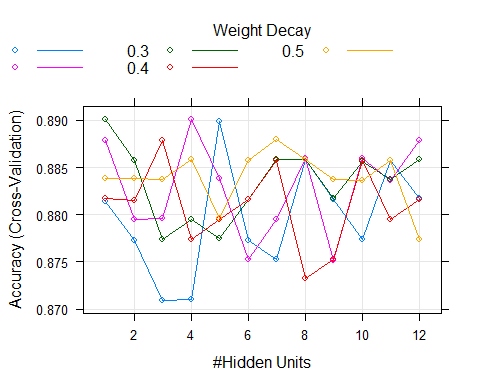
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction completed violated  
## completed 414 16  
## violated 4 39  
##   
## Accuracy : 0.9577   
## 95% CI : (0.9355, 0.974)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 1.25e-08   
##   
## Kappa : 0.7727   
## Mcnemar's Test P-Value : 0.01391   
##   
## Sensitivity : 0.70909   
## Specificity : 0.99043   
## Pos Pred Value : 0.90698   
## Neg Pred Value : 0.96279   
## Prevalence : 0.11628   
## Detection Rate : 0.08245   
## Detection Prevalence : 0.09091   
## Balanced Accuracy : 0.84976   
##   
## 'Positive' Class : violated   
##

This model came in extremly high at .957 accuracy.

start\_time = Sys.time()  
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 ~ .,  
 train,  
 method = "nnet",  
 tuneGrid = nnetGrid,  
 trControl = fitControl,  
 verbose = FALSE,  
 trace = FALSE)  
  
end\_time = Sys.time()  
end\_time-start\_time

## Time difference of 51.70089 secs

plot(nnetFit)



predNet = predict(nnetFit, train)

confusionMatrix(predNet, train$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction completed violated  
## completed 409 29  
## violated 9 26  
##   
## Accuracy : 0.9197   
## 95% CI : (0.8914, 0.9425)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.006833   
##   
## Kappa : 0.5358   
## Mcnemar's Test P-Value : 0.002055   
##   
## Sensitivity : 0.47273   
## Specificity : 0.97847   
## Pos Pred Value : 0.74286   
## Neg Pred Value : 0.93379   
## Prevalence : 0.11628   
## Detection Rate : 0.05497   
## Detection Prevalence : 0.07400   
## Balanced Accuracy : 0.72560   
##   
## 'Positive' Class : violated   
##

This model came in lower wth an accuracy at .919, but still great nonetheless.

start\_time = Sys.time()  
fitControl = trainControl(method = "cv",  
 number = 10)  
  
nnetGrid <- expand.grid(size = 12, decay = 0.1)  
  
set.seed(1234)  
nnetBasic2 = train(violator ~ .,  
 test,  
 method = "nnet",  
 tuneGrid = nnetGrid,  
 trControl = fitControl,  
 verbose = FALSE,  
 trace = FALSE)  
  
end\_time = Sys.time()  
end\_time-start\_time

## Time difference of 1.097686 secs

nnetBasic2

## Neural Network   
##   
## 202 samples  
## 8 predictor  
## 2 classes: 'completed', 'violated'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 182, 182, 182, 182, 181, 182, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8469048 0.05431235  
##   
## Tuning parameter 'size' was held constant at a value of 12  
##   
## Tuning parameter 'decay' was held constant at a value of 0.1

predNetBasic2 = predict(nnetBasic2, test)

confusionMatrix(predNetBasic2, test$violator, positive = "violated")

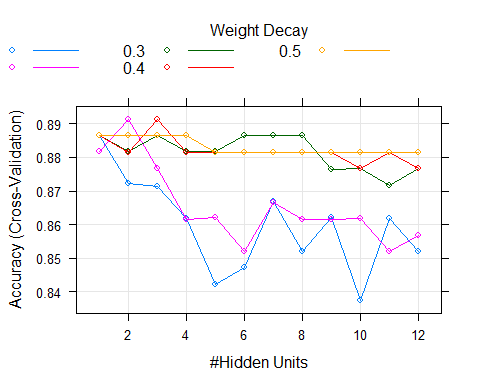
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction completed violated  
## completed 179 11  
## violated 0 12  
##   
## Accuracy : 0.9455   
## 95% CI : (0.9047, 0.9725)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.002921   
##   
## Kappa : 0.6591   
## Mcnemar's Test P-Value : 0.002569   
##   
## Sensitivity : 0.52174   
## Specificity : 1.00000   
## Pos Pred Value : 1.00000   
## Neg Pred Value : 0.94211   
## Prevalence : 0.11386   
## Detection Rate : 0.05941   
## Detection Prevalence : 0.05941   
## Balanced Accuracy : 0.76087   
##   
## 'Positive' Class : violated   
##

This model has a high accuracy of .945 which is of great quality.

start\_time = Sys.time()  
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)  
nnetFit2 = train(violator ~ .,  
 test,  
 method = "nnet",  
 tuneGrid = nnetGrid,  
 trControl = fitControl,  
 verbose = FALSE,  
 trace = FALSE)  
  
end\_time = Sys.time()  
end\_time-start\_time

## Time difference of 24.52789 secs

plot(nnetFit2)



predNet2 = predict(nnetFit2, test)

confusionMatrix(predNet2, test$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction completed violated  
## completed 178 18  
## violated 1 5  
##   
## Accuracy : 0.9059   
## 95% CI : (0.857, 0.9424)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.2224557   
##   
## Kappa : 0.3124   
## Mcnemar's Test P-Value : 0.0002419   
##   
## Sensitivity : 0.21739   
## Specificity : 0.99441   
## Pos Pred Value : 0.83333   
## Neg Pred Value : 0.90816   
## Prevalence : 0.11386   
## Detection Rate : 0.02475   
## Detection Prevalence : 0.02970   
## Balanced Accuracy : 0.60590   
##   
## 'Positive' Class : violated   
##

This model came in with good results too at a .905 accuracy. I don’t think there appears to be overfitting in either model.