ANN-for-Prediction-with-State.R

obite

2021-06-29

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.1 v dplyr 1.0.6  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

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

library(neuralnet)

## Warning: package 'neuralnet' was built under R version 4.0.5

##   
## Attaching package: 'neuralnet'

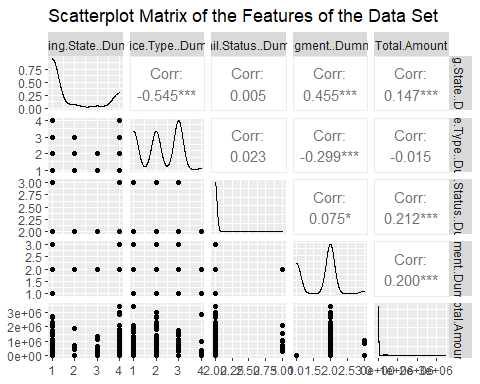
## The following object is masked from 'package:dplyr':  
##   
## compute

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.5

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

# Importing the dataset  
datum= read.csv('Dummy Data With State for ANN.csv')  
  
#Scatterplot matrix  
ggpairs(datum, title = "Scatterplot Matrix of the Features of the Data Set")



# Scale the Data  
scale01 <- function(x){  
 (x - min(x)) / (max(x) - min(x))  
}  
  
datum <- datum %>%  
 mutate\_all(scale01)  
  
set.seed(12345)  
datum\_Train <- sample\_frac(tbl = datum, replace = FALSE, size = 0.60)  
datum\_Test <- anti\_join(datum, datum\_Train)

## Joining, by = c("ï..Billing.State..Dummy.", "Device.Type..Dummy.", "Email.Status..Dummy.", "Segment..Dummy.", "Total.Amount")

#HIDDEN = 2  
  
set.seed(12321)  
train\_NN2 <- neuralnet(Total.Amount ~ ., data = datum\_Train, hidden=2)  
  
# Predict on test data  
pr <- compute(train\_NN2, datum\_Test)  
  
# Compute mean squared error  
pr.nn <- pr$net.result \* (max(datum$Total.Amount) - min(datum$Total.Amount)) + min(datum$Total.Amount)  
test.r <- (datum\_Test$Total.Amount) \* (max(datum$Total.Amount) - min(datum$Total.Amount)) + min(datum$Total.Amount)  
MSE.nn <- sum((test.r - pr.nn)^2) / nrow(datum\_Test)  
  
#mean squared error  
MSE.nn

## [1] 0.01723984

#Root mean squared error  
RMSE <- sqrt(MSE.nn)  
RMSE

## [1] 0.1313006

#To view the diagram of the ANN  
plot(train\_NN2)  
  
# Plot regression line  
plot(datum\_Test$Total.Amount, pr.nn, col = "red",  
 main = 'Real vs Predicted for HIDDEN = 2')  
abline(0, 1, lwd = 2)  
  
  
#HIDDEN = 5  
  
set.seed(12321)  
train\_NN5 <- neuralnet(Total.Amount ~ ., data = datum\_Train, hidden=5)  
  
# Predict on test data  
pr5 <- compute(train\_NN5, datum\_Test)  
  
# Compute mean squared error  
pr.nn5 <- pr5$net.result \* (max(datum$Total.Amount) - min(datum$Total.Amount)) + min(datum$Total.Amount)  
test.r5 <- (datum\_Test$Total.Amount) \* (max(datum$Total.Amount) - min(datum$Total.Amount)) + min(datum$Total.Amount)  
MSE.nn5 <- sum((test.r5 - pr.nn5)^2) / nrow(datum\_Test)  
  
#mean squared error  
MSE.nn5

## [1] 0.01610823

#Root mean squared error  
RMSE5 <- sqrt(MSE.nn5)  
RMSE5

## [1] 0.1269182

#To view the diagram of the ANN  
plot(train\_NN5)  
  
# Plot regression line  
plot(datum\_Test$Total.Amount, pr.nn5, col = "blue",  
 main = 'Real vs Predicted for HIDDEN = 5')  
abline(0, 1, lwd = 2)