Project

## R Markdown

library(neuralnet)

## Warning: package 'neuralnet' was built under R version 4.1.3

library(readxl)

#Training Data Set

train <- read\_excel("big\_training\_data.xlsx")

#Normalizing my data points

train$gre <- (train$gre - min(train$gre))/(max(train$gre) - min(train$gre))

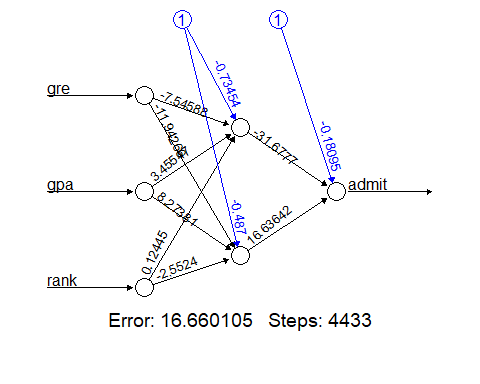
train$gpa <- (train$gpa - min(train$gpa))/(max(train$gpa) - min(train$gpa))

train$rank <- (train$rank - min(train$rank))/(max(train$rank) - min(train$rank))

#Neural Network

network <- neuralnet(admit ~ ., data = train, hidden = 2, err.fct = "sse", act.fct = "logistic", linear.output = FALSE, rep = 5)

plot(network, rep = "best")



network$result.matrix

## [,1] [,2] [,3] [,4]  
## error 1.715751e+01 17.956494130 1.673095e+01 1.666010e+01  
## reached.threshold 7.713846e-03 0.009811837 9.949996e-03 9.961812e-03  
## steps 1.086000e+03 329.000000000 3.997000e+03 4.433000e+03  
## Intercept.to.1layhid1 -3.303754e+00 -1.815419853 7.294716e-01 -7.345410e-01  
## gre.to.1layhid1 -9.903640e-01 1.928969988 1.519398e+01 -7.545882e+00  
## gpa.to.1layhid1 1.375030e+00 4.167603025 -1.188435e+01 3.455474e+00  
## rank.to.1layhid1 8.737332e+00 -6.311638350 4.751881e+00 1.244487e-01  
## Intercept.to.1layhid2 -1.911783e+01 25.531644012 -8.149295e-01 -4.870039e-01  
## gre.to.1layhid2 6.841104e+00 24.270606230 -7.032174e+00 -1.194266e+01  
## gpa.to.1layhid2 2.293086e+01 25.426699641 2.566648e+00 8.273810e+00  
## rank.to.1layhid2 1.207409e+01 10.739355532 7.198971e-01 -2.552397e+00  
## Intercept.to.admit -9.775861e-01 -1.079851681 7.539317e+00 -1.809455e-01  
## 1layhid1.to.admit -3.049571e+00 2.752552531 -7.782575e+00 -3.167770e+01  
## 1layhid2.to.admit 2.496550e+00 -0.811032762 -2.144466e+01 1.663642e+01  
## [,5]  
## error 1.676210e+01  
## reached.threshold 9.791635e-03  
## steps 3.515000e+03  
## Intercept.to.1layhid1 1.695409e+00  
## gre.to.1layhid1 1.936023e+01  
## gpa.to.1layhid1 -1.708203e+01  
## rank.to.1layhid1 7.990756e+00  
## Intercept.to.1layhid2 -1.258323e+00  
## gre.to.1layhid2 -6.818082e+00  
## gpa.to.1layhid2 2.325801e+00  
## rank.to.1layhid2 9.197139e-01  
## Intercept.to.admit 5.990258e+00  
## 1layhid1.to.admit -6.268193e+00  
## 1layhid2.to.admit -2.514971e+01

#Test Data Set

test <- read\_excel("big\_test\_data.xlsx")

#Normalizing Test Data Set

test$gre <- (test$gre - min(test$gre))/(max(test$gre) - min(test$gre))

test$gpa <- (test$gpa - min(test$gpa))/(max(test$gpa) - min(test$gpa))

test$rank <- (test$rank - min(test$rank))/(max(test$rank) - min(test$rank))

#Prediction

Predict = compute(network,test)

Predict$net.result

## [,1]  
## [1,] 0.20500716  
## [2,] 0.09606776  
## [3,] 0.46422767  
## [4,] 0.02083853  
## [5,] 0.17936014  
## [6,] 0.25564293  
## [7,] 0.50519282  
## [8,] 0.16851405  
## [9,] 0.05795960  
## [10,] 0.19407417  
## [11,] 0.50881714  
## [12,] 0.20130344  
## [13,] 0.17901933  
## [14,] 0.17926295  
## [15,] 0.08486275  
## [16,] 0.08167578  
## [17,] 0.17967292  
## [18,] 0.17889285  
## [19,] 0.27411129  
## [20,] 0.16915029  
## [21,] 0.45272902  
## [22,] 0.17686258  
## [23,] 0.10620549  
## [24,] 0.02505185  
## [25,] 0.79303434  
## [26,] 0.55070424  
## [27,] 0.49960866  
## [28,] 0.49043333  
## [29,] 0.17086906  
## [30,] 0.19986936  
## [31,] 0.20524175  
## [32,] 0.24982285  
## [33,] 0.20420204  
## [34,] 0.19651324  
## [35,] 0.78946822  
## [36,] 0.19160963  
## [37,] 0.17983184  
## [38,] 0.25798623  
## [39,] 0.20820004  
## [40,] 0.17933595  
## [41,] 0.02539999  
## [42,] 0.17873460  
## [43,] 0.15038299  
## [44,] 0.20340452  
## [45,] 0.12520148  
## [46,] 0.50936439  
## [47,] 0.46151543  
## [48,] 0.28080299  
## [49,] 0.12454701  
## [50,] 0.49330229  
## [51,] 0.20714279  
## [52,] 0.19968277  
## [53,] 0.54559345  
## [54,] 0.45193144  
## [55,] 0.26176415  
## [56,] 0.55410780  
## [57,] 0.49563388  
## [58,] 0.19905827  
## [59,] 0.20026797  
## [60,] 0.78127131  
## [61,] 0.55725998  
## [62,] 0.40606883  
## [63,] 0.43681869  
## [64,] 0.34416507  
## [65,] 0.02595628  
## [66,] 0.04511436  
## [67,] 0.20257850  
## [68,] 0.78515352  
## [69,] 0.51803072  
## [70,] 0.45353030  
## [71,] 0.20232268  
## [72,] 0.26350519  
## [73,] 0.44421650  
## [74,] 0.45833274  
## [75,] 0.17927525  
## [76,] 0.50213005  
## [77,] 0.49981065  
## [78,] 0.21032217  
## [79,] 0.12007281  
## [80,] 0.52511841  
## [81,] 0.49066254  
## [82,] 0.48961969  
## [83,] 0.78967880  
## [84,] 0.11757541  
## [85,] 0.24873426  
## [86,] 0.50350156  
## [87,] 0.49833392  
## [88,] 0.38519762  
## [89,] 0.52680277  
## [90,] 0.46731225  
## [91,] 0.47420079  
## [92,] 0.20725334  
## [93,] 0.48263111  
## [94,] 0.19166010  
## [95,] 0.44191895  
## [96,] 0.16677420  
## [97,] 0.11454240  
## [98,] 0.52511841  
## [99,] 0.19694696

#Convert probability to binary 0 = male, 1 = female

prob <- Predict$net.result  
pred <- ifelse(prob > 0.5, 1, 0)

pred

## [,1]  
## [1,] 0  
## [2,] 0  
## [3,] 0  
## [4,] 0  
## [5,] 0  
## [6,] 0  
## [7,] 1  
## [8,] 0  
## [9,] 0  
## [10,] 0  
## [11,] 1  
## [12,] 0  
## [13,] 0  
## [14,] 0  
## [15,] 0  
## [16,] 0  
## [17,] 0  
## [18,] 0  
## [19,] 0  
## [20,] 0  
## [21,] 0  
## [22,] 0  
## [23,] 0  
## [24,] 0  
## [25,] 1  
## [26,] 1  
## [27,] 0  
## [28,] 0  
## [29,] 0  
## [30,] 0  
## [31,] 0  
## [32,] 0  
## [33,] 0  
## [34,] 0  
## [35,] 1  
## [36,] 0  
## [37,] 0  
## [38,] 0  
## [39,] 0  
## [40,] 0  
## [41,] 0  
## [42,] 0  
## [43,] 0  
## [44,] 0  
## [45,] 0  
## [46,] 1  
## [47,] 0  
## [48,] 0  
## [49,] 0  
## [50,] 0  
## [51,] 0  
## [52,] 0  
## [53,] 1  
## [54,] 0  
## [55,] 0  
## [56,] 1  
## [57,] 0  
## [58,] 0  
## [59,] 0  
## [60,] 1  
## [61,] 1  
## [62,] 0  
## [63,] 0  
## [64,] 0  
## [65,] 0  
## [66,] 0  
## [67,] 0  
## [68,] 1  
## [69,] 1  
## [70,] 0  
## [71,] 0  
## [72,] 0  
## [73,] 0  
## [74,] 0  
## [75,] 0  
## [76,] 1  
## [77,] 0  
## [78,] 0  
## [79,] 0  
## [80,] 1  
## [81,] 0  
## [82,] 0  
## [83,] 1  
## [84,] 0  
## [85,] 0  
## [86,] 1  
## [87,] 0  
## [88,] 0  
## [89,] 1  
## [90,] 0  
## [91,] 0  
## [92,] 0  
## [93,] 0  
## [94,] 0  
## [95,] 0  
## [96,] 0  
## [97,] 0  
## [98,] 1  
## [99,] 0