**Build a Neural Network model for 50\_startups data to predict profit**

**Ans:**

> library(corrplot)

> library(GGally)

> library(neuralnet)

> library(nnet)

> library(caret)

> library(NeuralNetTools)

> startup <- read.csv(file.choose())

> View(startup)

> summary(startup)

R.D.Spend Administration Marketing.Spend State Profit

Min. : 0 Min. : 51283 Min. : 0 California:17 Min. : 14681

1st Qu.: 39936 1st Qu.:103731 1st Qu.:129300 Florida :16 1st Qu.: 90139

Median : 73051 Median :122700 Median :212716 New York :17 Median :107978

Mean : 73722 Mean :121345 Mean :211025 Mean :112013

3rd Qu.:101603 3rd Qu.:144842 3rd Qu.:299469 3rd Qu.:139766

Max. :165349 Max. :182646 Max. :471784 Max. :192262

> str(startup)

'data.frame': 50 obs. of 5 variables:

$ R.D.Spend : num 165349 162598 153442 144372 142107 ...

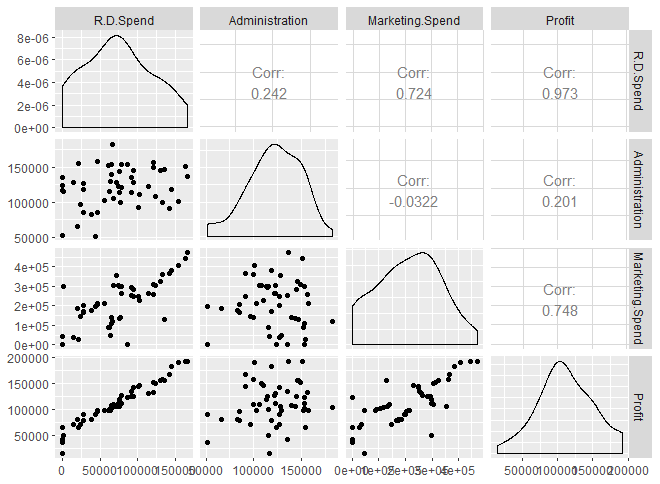
$ Administration : num 136898 151378 101146 118672 91392 ...

$ Marketing.Spend: num 471784 443899 407935 383200 366168 ...

$ State : Factor w/ 3 levels "California","Florida",..: 3 1 2 3 2 3 1 2 3 1 ...

$ Profit : num 192262 191792 191050 182902 166188 ...

> ggpairs(startup[,-4])



> startup$statedummy[startup$State=="New York"] <- 0

> startup$statedummy[startup$State=="California"] <-1

> startup$statedummy[startup$State=="Florida"] <- 2

> startup$statedummy <- as.numeric(startup$statedummy)

> str(startup[,-4])

'data.frame': 50 obs. of 5 variables:

$ R.D.Spend : num 165349 162598 153442 144372 142107 ...

$ Administration : num 136898 151378 101146 118672 91392 ...

$ Marketing.Spend: num 471784 443899 407935 383200 366168 ...

$ Profit : num 192262 191792 191050 182902 166188 ...

$ statedummy : num 0 1 2 0 2 0 1 2 0 1 ...

**Normalising the dataset**

> norm <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

> denorm <- function(x,min,max){

return((max-min)\*x+min)

}

> data <- as.data.frame(lapply(startup[,-4],norm))

> head(data)

R.D.Spend Administration Marketing.Spend Profit statedummy

1 1.0000000 0.6517439 1.0000000 1.0000000 0.0

2 0.9833595 0.7619717 0.9408934 0.9973546 0.5

3 0.9279846 0.3795790 0.8646636 0.9931781 1.0

4 0.8731364 0.5129984 0.8122351 0.9472924 0.0

5 0.8594377 0.3053280 0.7761356 0.8531714 1.0

6 0.7975660 0.3694479 0.7691259 0.8013818 0.0

> attach(data)

The following objects are masked from data (pos = 4):

Administration, Marketing.Spend, Profit, R.D.Spend, statedummy

**Splitting data to train and test data**

> s <- createDataPartition(data$Profit,p=0.8,list = F)

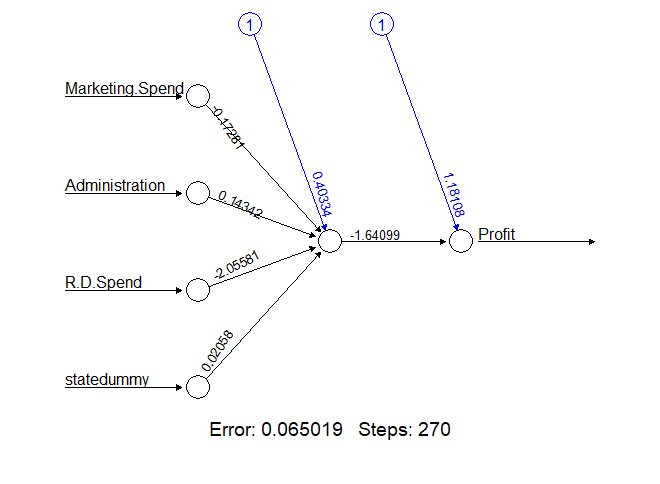
> trn\_s <- data[s,]

> tst\_s <- data[-s,]

**Model building**

> model <- neuralnet(Profit~Marketing.Spend+Administration+R.D.Spend+statedummy,data = trn\_s)

> plot(model,rep = "best")



> str(model)

List of 14

$ call : language neuralnet(formula = Profit ~ Marketing.Spend + Administration + R.D.Spend + statedummy, data = trn\_s)

$ response : num [1:42, 1] 0.997 0.993 0.947 0.853 0.801 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. ..$ : chr "Profit"

$ covariate : num [1:42, 1:4] 0.941 0.865 0.812 0.776 0.769 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. ..$ : chr [1:4] "Marketing.Spend" "Administration" "R.D.Spend" "statedummy"

$ model.list :List of 2

..$ response : chr "Profit"

..$ variables: chr [1:4] "Marketing.Spend" "Administration" "R.D.Spend" "statedummy"

$ err.fct :function (x, y)

..- attr(\*, "type")= chr "sse"

$ act.fct :function (x)

..- attr(\*, "type")= chr "logistic"

$ linear.output : logi TRUE

$ data :'data.frame': 42 obs. of 5 variables:

..$ R.D.Spend : num [1:42] 0.983 0.928 0.873 0.859 0.798 ...

..$ Administration : num [1:42] 0.762 0.38 0.513 0.305 0.369 ...

..$ Marketing.Spend: num [1:42] 0.941 0.865 0.812 0.776 0.769 ...

..$ Profit : num [1:42] 0.997 0.993 0.947 0.853 0.801 ...

..$ statedummy : num [1:42] 0.5 1 0 1 0 0.5 1 0 0.5 1 ...

$ exclude : NULL

$ net.result :List of 1

..$ : num [1:42, 1] 0.919 0.901 0.871 0.865 0.834 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

$ weights :List of 1

..$ :List of 2

.. ..$ : num [1:5, 1] 0.4033 -0.1728 0.1434 -2.0558 0.0206

.. ..$ : num [1:2, 1] 1.18 -1.64

$ generalized.weights:List of 1

..$ : num [1:42, 1:4] 0.512 0.449 0.387 0.378 0.342 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

$ startweights :List of 1

..$ :List of 2

.. ..$ : num [1:5, 1] -0.5022 0.1315 -0.0789 0.8868 0.117

.. ..$ : num [1:2, 1] 0.319 -0.582

$ result.matrix : num [1:10, 1] 0.06502 0.00942 270 0.40334 -0.17281 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:10] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

- attr(\*, "class")= chr "nn"

**Model validation**

> modelresults <- compute(model,tst\_s)

> predprofit <- modelresults$net.result

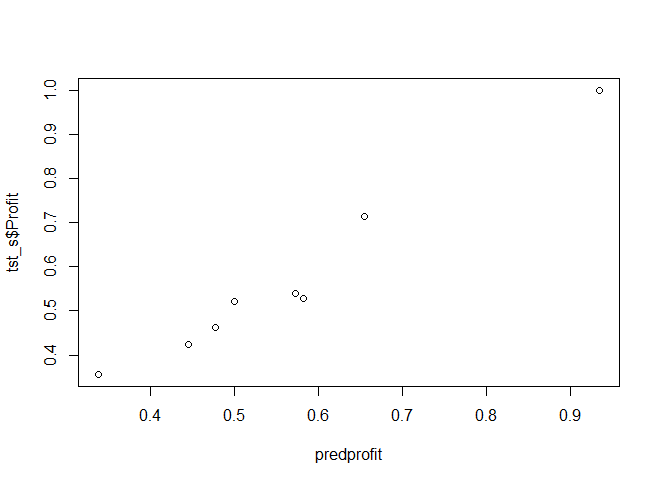
> cor(predprofit,tst\_s$Profit)

[,1]

[1,] 0.9829946

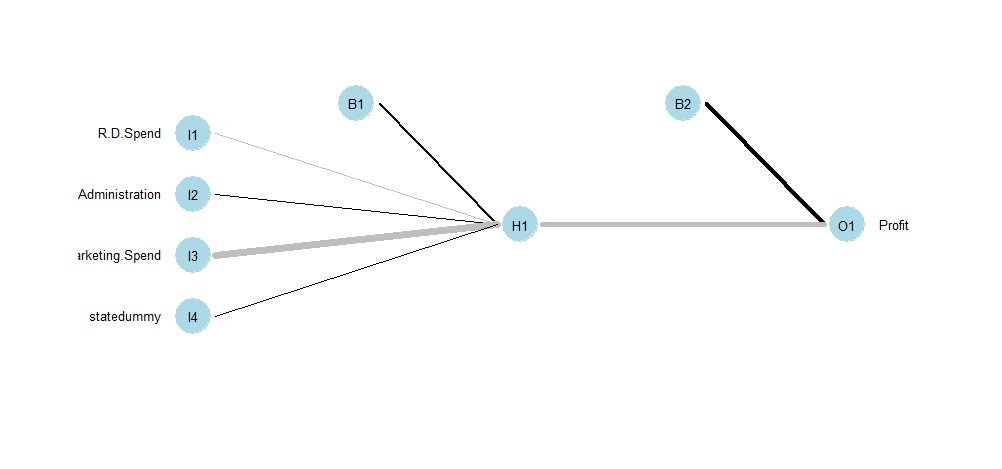
**SSE is higher than previous model**

> plot(predprofit,tst\_s$Profit)



**Visualisation**

> plotnet(model,x\_names = colnames(data[,-4]),y\_names = colnames(data[,4]),cex=0.8)



> max\_s <- max(data$Profit)

> min\_s <- min(data$Profit)

> actualpred\_profit <- denorm(predprofit,min = min\_s,max = max\_s)

> actualpred\_profit

[,1]

1 0.9343421

13 0.6551831

23 0.5727193

25 0.5820840

26 0.5005377

34 0.4779773

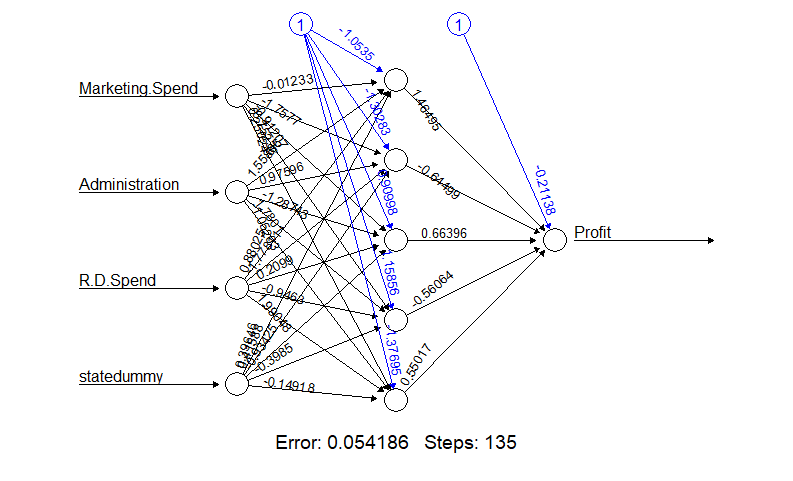
38 0.4449841

42 0.3386893

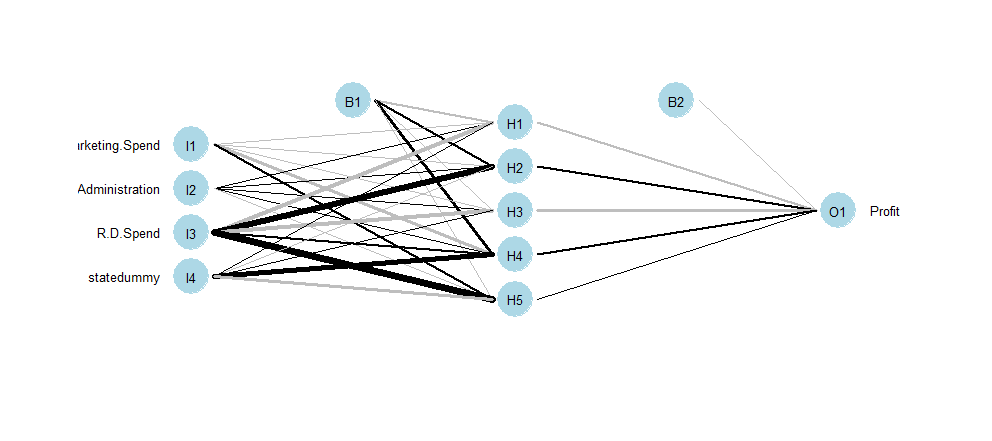
**Improving model performance**

> model2 <- neuralnet(Profit~Marketing.Spend+Administration+R.D.Spend+statedummy,data = trn\_s,hidden = 5,rep = 5)

> plot(model2,"best")



> plotnet(model2,cex=0.8)



> str(model2)

List of 14

$ call : language neuralnet(formula = Profit ~ Marketing.Spend + Administration + R.D.Spend + statedummy, data = trn\_s, hidden = 5, rep = 5)

$ response : num [1:42, 1] 0.997 0.993 0.947 0.853 0.801 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. ..$ : chr "Profit"

$ covariate : num [1:42, 1:4] 0.941 0.865 0.812 0.776 0.769 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. ..$ : chr [1:4] "Marketing.Spend" "Administration" "R.D.Spend" "statedummy"

$ model.list :List of 2

..$ response : chr "Profit"

..$ variables: chr [1:4] "Marketing.Spend" "Administration" "R.D.Spend" "statedummy"

$ err.fct :function (x, y)

..- attr(\*, "type")= chr "sse"

$ act.fct :function (x)

..- attr(\*, "type")= chr "logistic"

$ linear.output : logi TRUE

$ data :'data.frame': 42 obs. of 5 variables:

..$ R.D.Spend : num [1:42] 0.983 0.928 0.873 0.859 0.798 ...

..$ Administration : num [1:42] 0.762 0.38 0.513 0.305 0.369 ...

..$ Marketing.Spend: num [1:42] 0.941 0.865 0.812 0.776 0.769 ...

..$ Profit : num [1:42] 0.997 0.993 0.947 0.853 0.801 ...

..$ statedummy : num [1:42] 0.5 1 0 1 0 0.5 1 0 0.5 1 ...

$ exclude : NULL

$ net.result :List of 5

..$ : num [1:42, 1] 0.923 0.909 0.833 0.87 0.799 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1] 0.924 0.961 0.886 0.916 0.858 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1] 0.947 0.935 0.866 0.895 0.812 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1] 0.946 0.957 0.881 0.905 0.83 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1] 0.959 0.945 0.899 0.886 0.849 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

$ weights :List of 5

..$ :List of 2

.. ..$ : num [1:5, 1:5] -0.751 -0.167 0.266 -1.452 0.261 ...

.. ..$ : num [1:6, 1] -0.14 -0.477 0.53 -0.892 0.519 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] 1.7793 -0.074 -0.0936 -0.2618 0.3215 ...

.. ..$ : num [1:6, 1] -1.377 1.847 1.4 -0.815 -0.427 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -0.7272 1.9401 0.0192 -0.1465 0.2553 ...

.. ..$ : num [1:6, 1] 0.477 1.037 -0.355 -1.354 0.971 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -1.0535 -0.0123 1.5587 0.8802 0.3965 ...

.. ..$ : num [1:6, 1] -0.211 1.465 -0.645 0.664 -0.561 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -1.671 -0.2466 -0.2868 1.4384 0.0676 ...

.. ..$ : num [1:6, 1] -0.0978 0.9089 1.472 -0.152 -0.6831 ...

$ generalized.weights:List of 5

..$ : num [1:42, 1:4] 0.374 0.882 -0.108 0.741 -0.058 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1:4] 0.695 3.059 0.857 1.921 0.905 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1:4] -2.667 -2.828 -1.071 -1.498 -0.792 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1:4] 0.8312 1.1916 0.0239 0.5392 0.0344 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:42, 1:4] 0.0246 1.1575 0.2994 0.7221 0.3749 ...

.. ..- attr(\*, "dimnames")=List of 2

.. .. ..$ : chr [1:42] "2" "3" "4" "5" ...

.. .. ..$ : NULL

$ startweights :List of 5

..$ :List of 2

.. ..$ : num [1:5, 1:5] -0.5022 0.1315 -0.0789 0.8868 0.117 ...

.. ..$ : num [1:6, 1] -0.438 -0.72 0.231 -1.158 0.247 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] 1.757 -0.138 -0.111 -0.69 -0.222 ...

.. ..$ : num [1:6, 1] -1.399 1.825 1.381 -0.839 -0.262 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -0.379 2.582 0.13 -0.713 0.638 ...

.. ..$ : num [1:6, 1] 0.817 1.727 -0.104 -0.557 1.428 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -1.158 -0.53 2.446 -0.832 0.414 ...

.. ..$ : num [1:6, 1] -0.303 1.615 -0.774 0.424 -0.584 ...

..$ :List of 2

.. ..$ : num [1:5, 1:5] -1.545 -0.519 -0.28 1.007 -0.47 ...

.. ..$ : num [1:6, 1] 0.1342 1.0347 1.6535 -0.0179 -0.0242 ...

$ result.matrix : num [1:34, 1:5] 0.06438 0.00892 103 -0.75106 -0.1668 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:34] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

- attr(\*, "class")= chr "nn"

> model2result <- compute(model2,tst\_s)

> pred2profit <- model2result$net.result

> cor(pred2profit,tst\_s$Profit)

[,1]

[1,] 0.9713848

**SSE is decreased with improved Accuracy**