#Title : PREDICT THE BURNED AREA OF FOREST FIRES WITH NEURAL NETWORKS

library(neuralnet)

library(nnet)

library(caret)

install.packages("NeuralNetTools")

library(corrplot)

library(NeuralNetTools)

forest <- read.csv(file.choose())

View(forest)

Error in View : object 'forest' not found

> install.packages("NeuralNetTools")

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/TEMP.DESKTOP-LD1KGA9.025/Documents/R/win-library/4.0’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/NeuralNetTools\_1.5.2.zip'

Content type 'application/zip' length 204044 bytes (199 KB)

downloaded 199 KB

package ‘NeuralNetTools’ successfully unpacked and MD5 sums checked

> install.packages("NeuralNetTools")

> library(corrplot)

> library(NeuralNetTools)

Warning message:

package ‘NeuralNetTools’ was built under R version 4.0.3

> library(corrplot)

> library(NeuralNetTools)

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

> View(forest)

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : chr "mar" "oct" "oct" "mar" ...

$ day : chr "fri" "tue" "sat" "fri" ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: chr "small" "small" "small" "small" ...

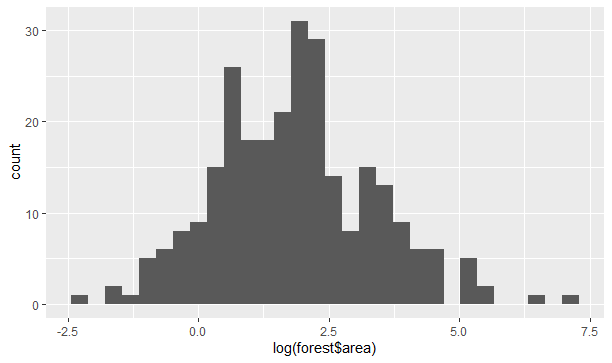
> table(forest$size\_category)

large small

139 378

> ggplot(forest)+geom\_histogram(mapping = aes(x=log(forest$area)))

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



> colnames(forest)

[1] "month" "day" "FFMC" "DMC" "DC"

[6] "ISI" "temp" "RH" "wind" "rain"

[11] "area" "dayfri" "daymon" "daysat" "daysun"

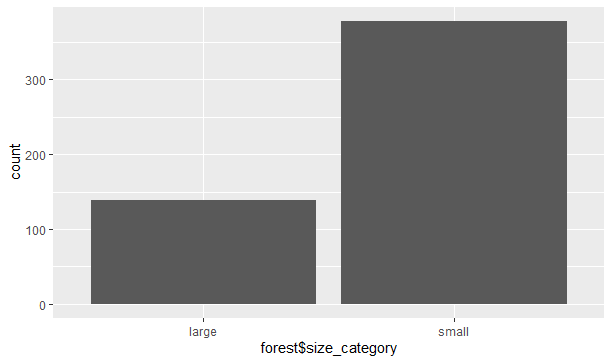
[16] "daythu" "daytue" "daywed" "monthapr" "monthaug"

[21] "monthdec" "monthfeb" "monthjan" "monthjul" "monthjun"

[26] "monthmar" "monthmay" "monthnov" "monthoct" "monthsep"

[31] "size\_category"

> ggplot(data = forest)+geom\_bar(aes(x=forest$size\_category))



forest$month=as.integer(factor(forest$month,levels = c("jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"),labels = c(1,2,3,4,5,6,7,8,9,10,11,12)))

> forest$day=as.integer(factor(forest$day,levels = c("sun","mon","tue","wed","thu","fri","sat"),labels = c(1,2,3,4,5,6,7)))

> forest$size\_category=as.integer(factor(forest$size\_category,levels = c("large","small"),labels = c(1,0)))

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : int 3 10 10 3 3 8 8 8 9 9 ...

$ day : int 6 3 7 6 1 1 2 2 3 7 ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: int 2 2 2 2 2 2 2 2 2 2 ...

> normalize <- function(x){

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

+ }

>

> norm\_foresr <- as.data.frame(lapply(forest,normalize))

> View(norm\_foresr)

> attach(norm\_foresr)

#splitting of data to train and test

> train <- sample(2,nrow(norm\_foresr),prob = c(0.7,0.3),replace = T)

> trn <- norm\_foresr[train==1,]

> tst <- norm\_foresr[train==2,]

>model <- neuralnet(area~.,data = trn,rep=5)

> str(model)

List of 14

$ call : language neuralnet(formula = area ~ ., data = trn, rep = 5)

$ response : num [1:365, 1] 0 0 0 0 0 0 0 0 0 0 ...

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

.. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. ..$ : chr "area"

$ covariate : num [1:365, 1:30] 0.182 0.818 0.182 0.182 0.636 ...

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

.. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. ..$ : chr [1:30] "month" "day" "FFMC" "DMC" ...

$ model.list :List of 2

..$ response : chr "area"

..$ variables: chr [1:30] "month" "day" "FFMC" "DMC" ...

$ err.fct :function (x, y)

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

$ act.fct :function (x)

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

$ linear.output : logi TRUE

$ data :'data.frame': 365 obs. of 31 variables:

..$ month : num [1:365] 0.182 0.818 0.182 0.182 0.636 ...

..$ day : num [1:365] 0.833 1 0.833 0 0 ...

..$ FFMC : num [1:365] 0.871 0.928 0.942 0.911 0.95 ...

..$ DMC : num [1:365] 0.0865 0.1468 0.111 0.173 0.2901 ...

..$ DC : num [1:365] 0.1013 0.7963 0.0816 0.1106 0.563 ...

..$ ISI : num [1:365] 0.0909 0.1194 0.1604 0.1711 0.262 ...

..$ temp : num [1:365] 0.193 0.399 0.196 0.296 0.643 ...

..$ RH : num [1:365] 0.424 0.212 0.965 0.988 0.165 ...

..$ wind : num [1:365] 0.7 0.1 0.4 0.156 0.556 ...

..$ rain : num [1:365] 0 0 0.0312 0 0 ...

..$ area : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ dayfri : num [1:365] 1 0 1 0 0 0 0 0 0 1 ...

..$ daymon : num [1:365] 0 0 0 0 0 1 1 0 0 0 ...

..$ daysat : num [1:365] 0 1 0 0 0 0 0 0 0 0 ...

..$ daysun : num [1:365] 0 0 0 1 1 0 0 0 0 0 ...

..$ daythu : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ daytue : num [1:365] 0 0 0 0 0 0 0 1 0 0 ...

..$ daywed : num [1:365] 0 0 0 0 0 0 0 0 1 0 ...

..$ monthapr : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthaug : num [1:365] 0 0 0 0 1 1 1 0 0 0 ...

..$ monthdec : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthfeb : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjan : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjul : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjun : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthmar : num [1:365] 1 0 1 1 0 0 0 0 0 0 ...

..$ monthmay : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthnov : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthoct : num [1:365] 0 1 0 0 0 0 0 0 0 0 ...

..$ monthsep : num [1:365] 0 0 0 0 0 0 0 1 1 1 ...

..$ size\_category: num [1:365] 1 1 1 1 1 1 1 1 1 1 ...

$ exclude : NULL

$ net.result :List of 5

..$ : num [1:365, 1] 0.00751 0.00751 0.00751 0.00751 0.00751 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1] 0.00602 0.00602 0.00602 0.00602 0.00602 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1] 0.00582 0.00582 0.00582 0.00582 0.00582 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1] -0.004581 -0.002705 0.000206 0.002644 -0.005973 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1] 0.00733 0.00733 0.00733 0.00733 0.00733 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

$ weights :List of 5

..$ :List of 2

.. ..$ : num [1:31, 1] -1.239 -0.315 -0.713 -1.248 -0.335 ...

.. ..$ : num [1:2, 1] 0.00751 3.98916

..$ :List of 2

.. ..$ : num [1:31, 1] -2.311 3.5873 0.3143 -0.0898 -18.0005 ...

.. ..$ : num [1:2, 1] 1.39 -1.39

..$ :List of 2

.. ..$ : num [1:31, 1] -0.6534 0.1265 0.1702 -0.0741 -18.871 ...

.. ..$ : num [1:2, 1] 1.35 -1.35

..$ :List of 2

.. ..$ : num [1:31, 1] -0.102 0.61 0.302 2.624 -0.299 ...

.. ..$ : num [1:2, 1] -0.478 0.534

..$ :List of 2

.. ..$ : num [1:31, 1] -1.354 -1.85 0.687 -2.395 2.121 ...

.. ..$ : num [1:2, 1] 0.00733 4.21931

$ generalized.weights:List of 5

..$ : num [1:365, 1:30] -4.76e-17 -6.01e-16 -2.66e-19 -6.48e-20 -2.06e-17 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1:30] 0 0 0 0 0 0 0 0 0 0 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1:30] 0 0 0 0 0 0 0 0 0 0 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1:30] -7.07 -11.67 147.25 11.03 -5.53 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

..$ : num [1:365, 1:30] -6.64e-13 -2.18e-11 -4.12e-15 -1.09e-14 -4.01e-12 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

$ startweights :List of 5

..$ :List of 2

.. ..$ : num [1:31, 1] -0.326 0.552 -0.675 0.214 0.311 ...

.. ..$ : num [1:2, 1] 0.279 1.008

..$ :List of 2

.. ..$ : num [1:31, 1] -2.073 1.19 -0.724 0.168 0.92 ...

.. ..$ : num [1:2, 1] 0.423 0.387

..$ :List of 2

.. ..$ : num [1:31, 1] -0.6878 0.1489 -0.0576 -0.0748 1.5099 ...

.. ..$ : num [1:2, 1] 1.365 0.497

..$ :List of 2

.. ..$ : num [1:31, 1] -0.814 0.268 -0.592 2.133 1.173 ...

.. ..$ : num [1:2, 1] -0.783 0.245

..$ :List of 2

.. ..$ : num [1:31, 1] -0.144 -1.609 0.952 -1.819 1.784 ...

.. ..$ : num [1:2, 1] -1.587 -0.927

$ result.matrix : num [1:36, 1:5] 0.31015 0.00882 330 -1.23947 -0.31477 ...

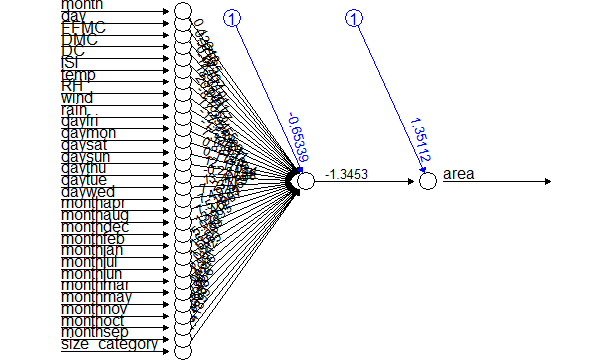
..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:36] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

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

> plot(model,"best")

>

> #prediction

> set.seed(101)

> predict\_model <- compute(model,tst)

> pred <- predict\_model$net.result

> cor(pred,tst$area)

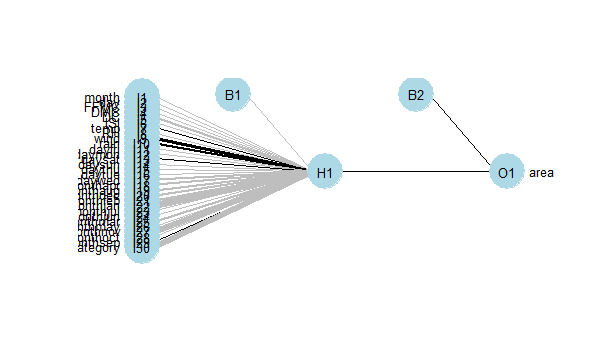
[,1]

[1,] 0.06787342

#visualisation

>

> plotnet(model,cex=0.8)



#improving performance of the modelby including hidden nodes result from bagging method

> set.seed(100)

> model2 <- neuralnet(area~.,hidden = 7,data = trn,) #hidden choosen from bagging method

> str(model2)

List of 14

$ call : language neuralnet(formula = area ~ ., data = trn, hidden = 7)

$ response : num [1:365, 1] 0 0 0 0 0 0 0 0 0 0 ...

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

.. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. ..$ : chr "area"

$ covariate : num [1:365, 1:30] 0.182 0.818 0.182 0.182 0.636 ...

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

.. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. ..$ : chr [1:30] "month" "day" "FFMC" "DMC" ...

$ model.list :List of 2

..$ response : chr "area"

..$ variables: chr [1:30] "month" "day" "FFMC" "DMC" ...

$ err.fct :function (x, y)

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

$ act.fct :function (x)

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

$ linear.output : logi TRUE

$ data :'data.frame': 365 obs. of 31 variables:

..$ month : num [1:365] 0.182 0.818 0.182 0.182 0.636 ...

..$ day : num [1:365] 0.833 1 0.833 0 0 ...

..$ FFMC : num [1:365] 0.871 0.928 0.942 0.911 0.95 ...

..$ DMC : num [1:365] 0.0865 0.1468 0.111 0.173 0.2901 ...

..$ DC : num [1:365] 0.1013 0.7963 0.0816 0.1106 0.563 ...

..$ ISI : num [1:365] 0.0909 0.1194 0.1604 0.1711 0.262 ...

..$ temp : num [1:365] 0.193 0.399 0.196 0.296 0.643 ...

..$ RH : num [1:365] 0.424 0.212 0.965 0.988 0.165 ...

..$ wind : num [1:365] 0.7 0.1 0.4 0.156 0.556 ...

..$ rain : num [1:365] 0 0 0.0312 0 0 ...

..$ area : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ dayfri : num [1:365] 1 0 1 0 0 0 0 0 0 1 ...

..$ daymon : num [1:365] 0 0 0 0 0 1 1 0 0 0 ...

..$ daysat : num [1:365] 0 1 0 0 0 0 0 0 0 0 ...

..$ daysun : num [1:365] 0 0 0 1 1 0 0 0 0 0 ...

..$ daythu : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ daytue : num [1:365] 0 0 0 0 0 0 0 1 0 0 ...

..$ daywed : num [1:365] 0 0 0 0 0 0 0 0 1 0 ...

..$ monthapr : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthaug : num [1:365] 0 0 0 0 1 1 1 0 0 0 ...

..$ monthdec : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthfeb : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjan : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjul : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthjun : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthmar : num [1:365] 1 0 1 1 0 0 0 0 0 0 ...

..$ monthmay : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthnov : num [1:365] 0 0 0 0 0 0 0 0 0 0 ...

..$ monthoct : num [1:365] 0 1 0 0 0 0 0 0 0 0 ...

..$ monthsep : num [1:365] 0 0 0 0 0 0 0 1 1 1 ...

..$ size\_category: num [1:365] 1 1 1 1 1 1 1 1 1 1 ...

$ exclude : NULL

$ net.result :List of 1

..$ : num [1:365, 1] 0.007258 -0.013299 0.000652 -0.009189 0.003285 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

$ weights :List of 1

..$ :List of 2

.. ..$ : num [1:31, 1:7] -0.3907 0.2765 -0.0074 0.9896 0.2765 ...

.. ..$ : num [1:8, 1] -0.889 0.958 -0.899 -0.925 0.245 ...

$ generalized.weights:List of 1

..$ : num [1:365, 1:30] -65.18 2.33 -666.3 12.91 -54.64 ...

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

.. .. ..$ : chr [1:365] "1" "3" "4" "5" ...

.. .. ..$ : NULL

$ startweights :List of 1

..$ :List of 2

.. ..$ : num [1:31, 1:7] -0.5022 0.1315 -0.0789 0.8868 0.117 ...

.. ..$ : num [1:8, 1] -0.9673 0.8794 -0.2536 -1.5179 -0.0239 ...

$ result.matrix : num [1:228, 1] 1.79e-02 9.67e-03 6.91e+03 -3.91e-01 2.77e-01 ...

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

.. ..$ : chr [1:228] "error" "reached.threshold" "steps" "Intercept.to.1layhid1" ...

.. ..$ : NULL

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

> set.seed(100)

> pred2model <- compute(model2,tst)

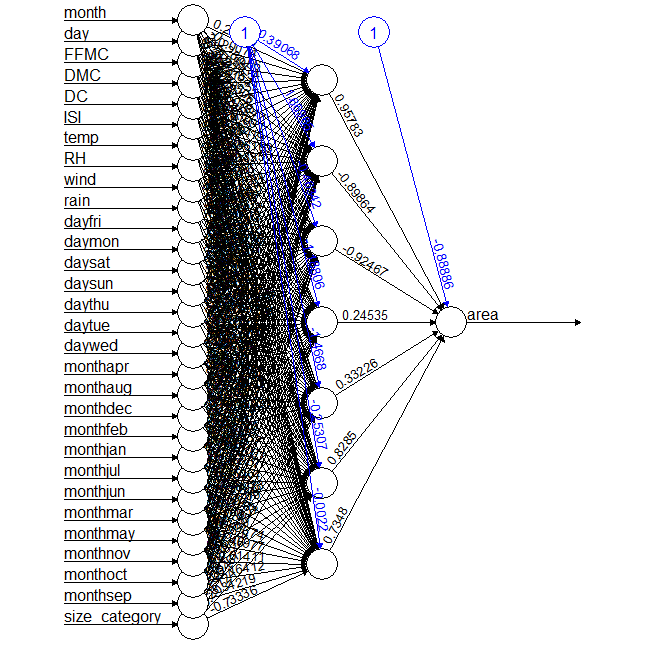
> pred2 <- pred2model$net.result

> cor(pred2,tst$area)

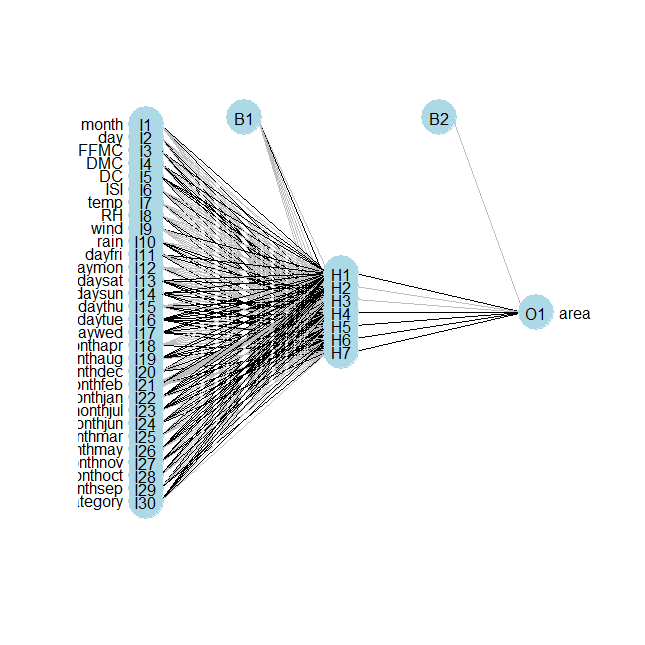
[,1]

[1,] 0.005670043

plot(model2)



plotnet(model2)



> #bagging

> acc <- c()

> for(i in seq(1,10,1)){

+ set.seed(100)

+ model\_bag <- neuralnet(area~.,hidden = i,data = trn)

+ pred\_bag <- compute(model\_bag,tst)

+ acc <- c(acc,cor(pred\_bag$net.result,tst$area))

+ }

acc

[1] 0.063030721 0.149591537 0.061483293 0.072842252 0.002228802 0.087123231

[7] 0.005670043 0.074650621 0.133326600 0.101221659

plot(seq(1,10,1),acc)

