Neural Network Project

Chakrika Vemireddy

2025-05-06

## Loading & Preparing the Data

## Loading the required dataset  
recruitment.df<-read.csv("RecruitmentData\_Preprocessed.csv",header=T)  
#View(recruitment.df)  
  
# Loading required packages  
library(neuralnet)  
library(caret)

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.4.3

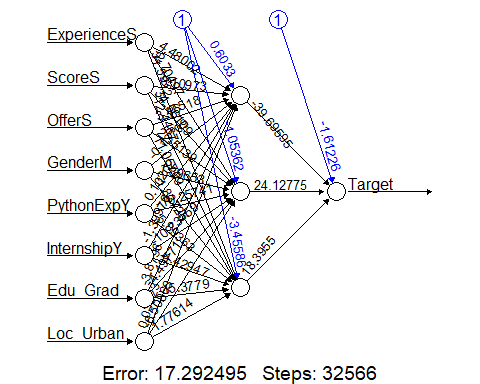
## Loading required package: lattice

library(e1071)  
  
# Creating training and validation sets  
set.seed(123)  
train.rows.r <- sample(rownames(recruitment.df), dim(recruitment.df)[1]\*0.6)  
train.data.r <- recruitment.df[train.rows.r, ]  
valid.rows.r <- setdiff(rownames(recruitment.df), train.rows.r)  
valid.data.r <- recruitment.df[valid.rows.r, ]  
  
# Showing column names for reference  
colnames(recruitment.df)

## [1] "Serial\_no" "ExperienceS" "ScoreS" "OfferS" "GenderM"   
## [6] "PythonExpY" "InternshipY" "Edu\_Grad" "Loc\_Urban" "Target"

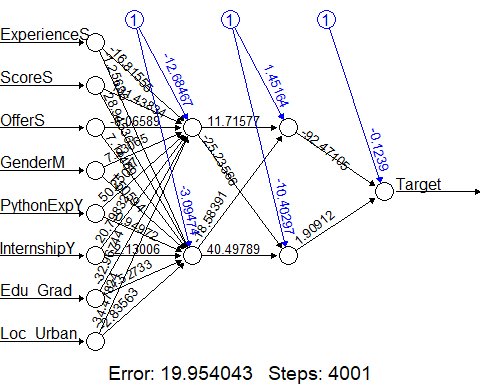
## Neural Network 1 – Single Hidden Layer

# Training neural network with 3 hidden node  
set.seed(123)  
nn.r1<-neuralnet(Target~  
 ExperienceS+ScoreS+OfferS+GenderM+PythonExpY+InternshipY+Edu\_Grad+  
 Loc\_Urban,data=train.data.r,linear.output=F,  
 hidden=3)  
  
# Ploting the neural net  
plot(nn.r1, rep = "best")



## Neural Network 2 – Two Hidden Layers

# Training neural network with 2 hidden layers of 2 nodes each  
set.seed(123)  
nn.r2<-neuralnet(Target~  
 ExperienceS+ScoreS+OfferS+GenderM+PythonExpY+InternshipY+Edu\_Grad+  
 Loc\_Urban,data=train.data.r,linear.output=F,  
 hidden=c(2,2))  
  
# Ploting the neural net  
plot(nn.r2, rep = "best")



## Evaluating Neural Network 1

## Prediction on validation set   
valid.pred.r1<-predict(nn.r1,valid.data.r[,c("ExperienceS", "ScoreS", "OfferS", "GenderM",   
 "PythonExpY", "InternshipY", "Edu\_Grad", "Loc\_Urban")])  
head(valid.pred.r1)

## [,1]  
## 1 7.953208e-01  
## 2 5.080840e-01  
## 3 9.999971e-01  
## 6 9.913823e-01  
## 8 3.074907e-06  
## 9 9.511617e-01

# Converting predictions to binary class  
valid.class.r1=ifelse(valid.pred.r1 > 0.5, 1, 0)  
head(valid.class.r1)

## [,1]  
## 1 1  
## 2 1  
## 3 1  
## 6 1  
## 8 0  
## 9 1

# Displaying performance  
confusionMatrix(as.factor(valid.class.r1), as.factor(valid.data.r$Target))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 27 12  
## 1 38 136  
##   
## Accuracy : 0.7653   
## 95% CI : (0.7025, 0.8205)  
## No Information Rate : 0.6948   
## P-Value [Acc > NIR] : 0.013945   
##   
## Kappa : 0.3765   
##   
## Mcnemar's Test P-Value : 0.000407   
##   
## Sensitivity : 0.4154   
## Specificity : 0.9189   
## Pos Pred Value : 0.6923   
## Neg Pred Value : 0.7816   
## Prevalence : 0.3052   
## Detection Rate : 0.1268   
## Detection Prevalence : 0.1831   
## Balanced Accuracy : 0.6672   
##   
## 'Positive' Class : 0   
##

## Evaluating Neural Network 2

# Prediction on validation set  
valid.pred.r2<-predict(nn.r2,valid.data.r[,c("ExperienceS", "ScoreS", "OfferS", "GenderM",   
 "PythonExpY", "InternshipY", "Edu\_Grad", "Loc\_Urban")])  
head(valid.pred.r2)

## [,1]  
## 1 8.563361e-01  
## 2 8.563392e-01  
## 3 8.532658e-01  
## 6 8.536402e-01  
## 8 1.279951e-28  
## 9 8.563329e-01

# Converting predictions to binary class  
valid.class.r2=ifelse(valid.pred.r2 > 0.5, 1, 0)  
head(valid.class.r2)

## [,1]  
## 1 1  
## 2 1  
## 3 1  
## 6 1  
## 8 0  
## 9 1

# Displaying performance  
confusionMatrix(as.factor(valid.class.r2), as.factor(valid.data.r$Target))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 26 4  
## 1 39 144  
##   
## Accuracy : 0.7981   
## 95% CI : (0.7379, 0.8499)  
## No Information Rate : 0.6948   
## P-Value [Acc > NIR] : 0.0004643   
##   
## Kappa : 0.4393   
##   
## Mcnemar's Test P-Value : 2.161e-07   
##   
## Sensitivity : 0.4000   
## Specificity : 0.9730   
## Pos Pred Value : 0.8667   
## Neg Pred Value : 0.7869   
## Prevalence : 0.3052   
## Detection Rate : 0.1221   
## Detection Prevalence : 0.1408   
## Balanced Accuracy : 0.6865   
##   
## 'Positive' Class : 0   
##