ams580\_hw6

Mustafa Yigit Isik

2023-03-21

# Loading data and packages

if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

if (!requireNamespace("neuralnet")) install.packages('neuralnet')

## Loading required namespace: neuralnet

if (!requireNamespace("keras")) install.packages('keras')

## Loading required namespace: keras

if (!requireNamespace("e1071")) install.packages('e1071')

## Loading required namespace: e1071

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(neuralnet)

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

library(keras)  
library(e1071)

# Importing data

data <- read.csv("/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/hw6\_neural\_netw/Titanic2.csv")  
data <- subset(data, select = -c(PassengerId,Name,Ticket,Cabin,Embarked))   
## I took out Embarked as well to read  
data <- subset(data, is.na(Age) == FALSE)  
data$Sex <- ifelse(data$Sex == "male", 1, 0)  
cat('There are', nrow(data), 'observations left.')

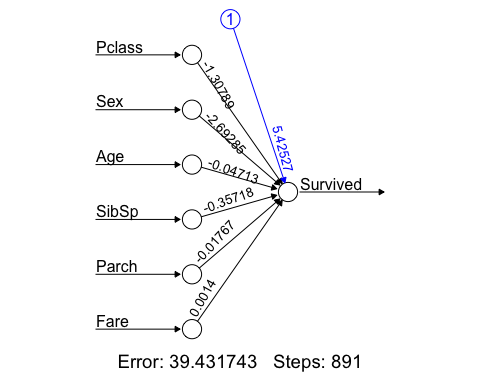
## There are 714 observations left.

str(data)

## 'data.frame': 714 obs. of 7 variables:  
## $ Survived: int 0 1 1 1 0 0 0 1 1 1 ...  
## $ Pclass : int 3 1 3 1 3 1 3 3 2 3 ...  
## $ Sex : num 1 0 0 0 1 1 1 0 0 0 ...  
## $ Age : num 22 38 26 35 35 54 2 27 14 4 ...  
## $ SibSp : int 1 1 0 1 0 0 3 0 1 1 ...  
## $ Parch : int 0 0 0 0 0 0 1 2 0 1 ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...

# Question 1  
set.seed(123)   
split <- data$Survived %>%   
createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[split,]  
test.data <- data[-split,]

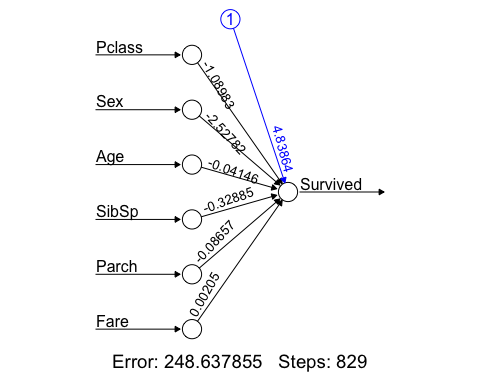
# Question 2 - Part 1  
set.seed(123)   
model <- neuralnet(Survived~., data = train.data, hidden = 0, err.fct = "sse", linear.output = F)  
plot(model, rep = "best")



# Question 2 - Part 2  
probabilities <- model %>% predict(test.data) %>% as.vector()  
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 90 22  
## 1 8 58  
##   
## Accuracy : 0.8315   
## 95% CI : (0.7682, 0.8833)  
## No Information Rate : 0.5506   
## P-Value [Acc > NIR] : 2.015e-15   
##   
## Kappa : 0.6539   
##   
## Mcnemar's Test P-Value : 0.01762   
##   
## Sensitivity : 0.7250   
## Specificity : 0.9184   
## Pos Pred Value : 0.8788   
## Neg Pred Value : 0.8036   
## Prevalence : 0.4494   
## Detection Rate : 0.3258   
## Detection Prevalence : 0.3708   
## Balanced Accuracy : 0.8217   
##   
## 'Positive' Class : 1   
##

# Question 3 - Part 1  
set.seed(123)  
model <- neuralnet(Survived~., data = train.data, hidden = 0, err.fct = "ce", linear.output = F)  
plot(model, rep = "best")



# Question 3 - Part 2  
probabilities <- model %>% predict(test.data)  
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 89 22  
## 1 9 58  
##   
## Accuracy : 0.8258   
## 95% CI : (0.762, 0.8785)  
## No Information Rate : 0.5506   
## P-Value [Acc > NIR] : 7.954e-15   
##   
## Kappa : 0.6428   
##   
## Mcnemar's Test P-Value : 0.03114   
##   
## Sensitivity : 0.7250   
## Specificity : 0.9082   
## Pos Pred Value : 0.8657   
## Neg Pred Value : 0.8018   
## Prevalence : 0.4494   
## Detection Rate : 0.3258   
## Detection Prevalence : 0.3764   
## Balanced Accuracy : 0.8166   
##   
## 'Positive' Class : 1   
##

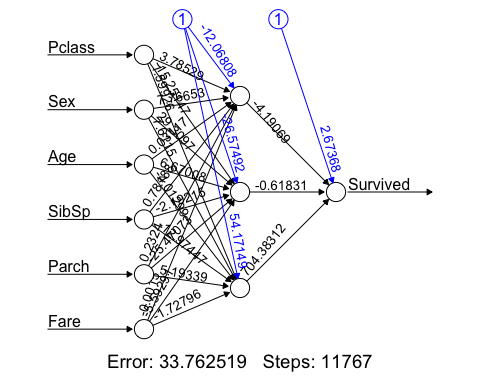
# Question 4 - Part 1  
set.seed(123)  
model <- glm(Survived~., family = binomial, data = train.data)  
model

##   
## Call: glm(formula = Survived ~ ., family = binomial, data = train.data)  
##   
## Coefficients:  
## (Intercept) Pclass Sex Age SibSp Parch   
## 4.839204 -1.089973 -2.527925 -0.041466 -0.328881 -0.086548   
## Fare   
## 0.002053   
##   
## Degrees of Freedom: 535 Total (i.e. Null); 529 Residual  
## Null Deviance: 717.7   
## Residual Deviance: 497.3 AIC: 511.3

# Question 4 - Part 2  
probabilities <- model %>% predict(test.data, type = 'response')  
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 89 22  
## 1 9 58  
##   
## Accuracy : 0.8258   
## 95% CI : (0.762, 0.8785)  
## No Information Rate : 0.5506   
## P-Value [Acc > NIR] : 7.954e-15   
##   
## Kappa : 0.6428   
##   
## Mcnemar's Test P-Value : 0.03114   
##   
## Sensitivity : 0.7250   
## Specificity : 0.9082   
## Pos Pred Value : 0.8657   
## Neg Pred Value : 0.8018   
## Prevalence : 0.4494   
## Detection Rate : 0.3258   
## Detection Prevalence : 0.3764   
## Balanced Accuracy : 0.8166   
##   
## 'Positive' Class : 1   
##

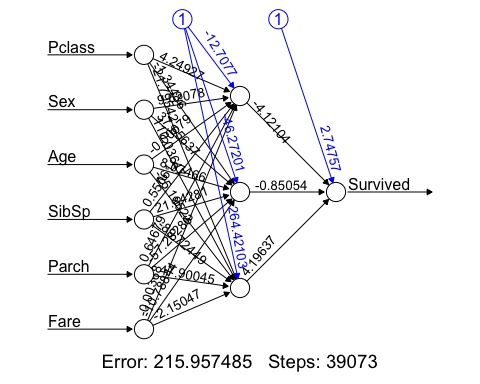
# Question 5 - Part 1  
set.seed(123)  
model <- neuralnet(Survived~., data = train.data, hidden = 3, err.fct = "sse", linear.output = F)  
plot(model, rep = "best")



# Question 5 - Part 2  
probabilities <- model %>% predict(test.data)  
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 94 26  
## 1 4 54  
##   
## Accuracy : 0.8315   
## 95% CI : (0.7682, 0.8833)  
## No Information Rate : 0.5506   
## P-Value [Acc > NIR] : 2.015e-15   
##   
## Kappa : 0.6506   
##   
## Mcnemar's Test P-Value : 0.000126   
##   
## Sensitivity : 0.6750   
## Specificity : 0.9592   
## Pos Pred Value : 0.9310   
## Neg Pred Value : 0.7833   
## Prevalence : 0.4494   
## Detection Rate : 0.3034   
## Detection Prevalence : 0.3258   
## Balanced Accuracy : 0.8171   
##   
## 'Positive' Class : 1   
##

# Question 6 - Part 1  
set.seed(123)  
model <- neuralnet(Survived~., data = train.data, hidden = 3, err.fct = "ce", linear.output = F)  
plot(model, rep = "best")



# Question 6 - Part 2  
probabilities <- model %>% predict(test.data)  
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 97 27  
## 1 1 53  
##   
## Accuracy : 0.8427   
## 95% CI : (0.7807, 0.8929)  
## No Information Rate : 0.5506   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6724   
##   
## Mcnemar's Test P-Value : 2.306e-06   
##   
## Sensitivity : 0.6625   
## Specificity : 0.9898   
## Pos Pred Value : 0.9815   
## Neg Pred Value : 0.7823   
## Prevalence : 0.4494   
## Detection Rate : 0.2978   
## Detection Prevalence : 0.3034   
## Balanced Accuracy : 0.8261   
##   
## 'Positive' Class : 1   
##

#- Steps in q6 takes much more compared to q3 but the accuracy is ~.8427 compared  
#to q3 where accuracy is ~.82  
#- sensitivty in q6 is worse off but the specificity is the clear winner with .98  
#compared to .90 in q3