HW 09

TUAN BUI

11/23/2021

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.1 ──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.4     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   2.0.1     ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## x dplyr::filter() masks stats::filter()
## x 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
```

```
titanic_data <- read.csv('~/OneDrive - Stony Brook University/SBU/MAT + AMS/F
all 2021/AMS 380/hw/09/Titanic.csv', header = T)
```

Question 01

```
titanic_data <- subset(titanic_data, select = -c(PassengerId, Name, Ticket, C
abin))
```

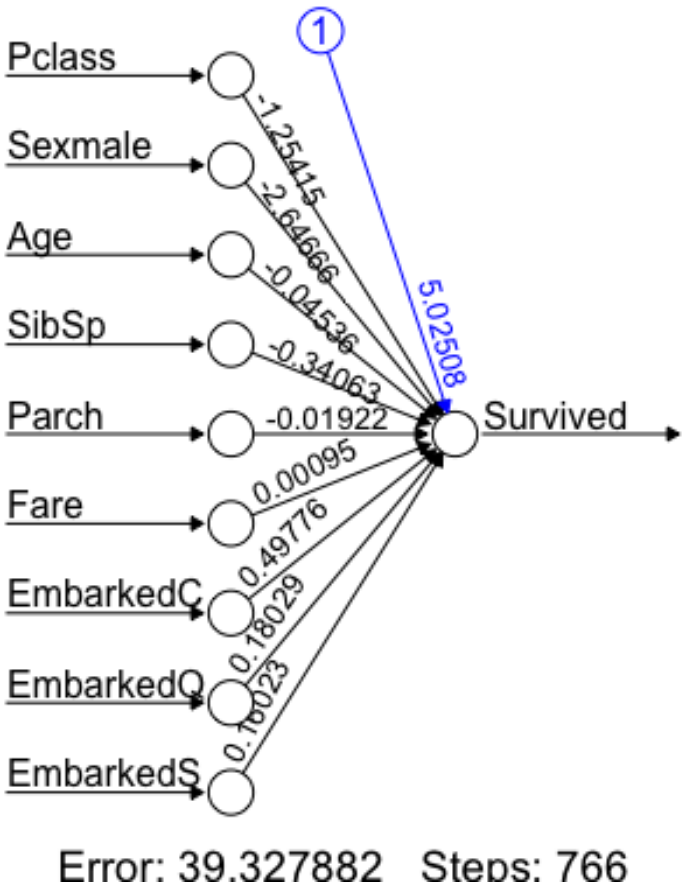
```
titanic_data <- na.omit(titanic_data)
str(titanic_data)
```

```
## 'data.frame':    714 obs. of  8 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     : chr  "male" "female" "female" "female" ...
##  $ 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 ...
##  $ Embarked: chr  "S" "C" "S" "S" ...
##  - attr(*, "na.action")= 'omit' Named int [1:177] 6 18 20 27 29 30 32 33 3
7 43 ...
##   ..- attr(*, "names")= chr [1:177] "6" "18" "20" "27" ...
```

```
# There are 714 observations left after omitting the missing data

x <- model.matrix(Survived ~ ., data = titanic_data)
titanic_data <- cbind(x[,-1], Survived = titanic_data$Survived)
titanic_data <- as.data.frame(titanic_data)

# Generate training and testing data
set.seed(123)
training.samples <- titanic_data$Survived %>%
  createDataPartition(p = 0.75, list = FALSE)
train.data  <- titanic_data[training.samples, ]
test.data <- titanic_data[-training.samples, ]
```

Question 02

```
set.seed(123)
model_02 <- neuralnet(Survived ~ ., data = train.data, hidden = 0, err.fct =
"sse", linear.output = F)

plot(model_02, rep = "best")
```




```
probabilities_02 <- model_02 %>% predict(test.data) %>% as.vector()
predicted.classes_02 <- ifelse(probabilities_02 > 0.5, 1, 0)
confusionMatrix(factor(predicted.classes_02), factor(test.data$Survived), pos
itive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 89 21
##          1  9 59
##
##                Accuracy : 0.8315
##                  95% CI : (0.7682, 0.8833)
##     No Information Rate : 0.5506
##     P-Value [Acc > NIR] : 2.015e-15
##
##                   Kappa : 0.6547
##
##  Mcnemar's Test P-Value : 0.04461
##
##             Sensitivity : 0.7375
##             Specificity : 0.9082
```

```
##          Pos Pred Value : 0.8676
##          Neg Pred Value : 0.8091
##              Prevalence : 0.4494
##          Detection Rate : 0.3315
##    Detection Prevalence : 0.3820
##       Balanced Accuracy : 0.8228
##
##        'Positive' Class : 1
##
```

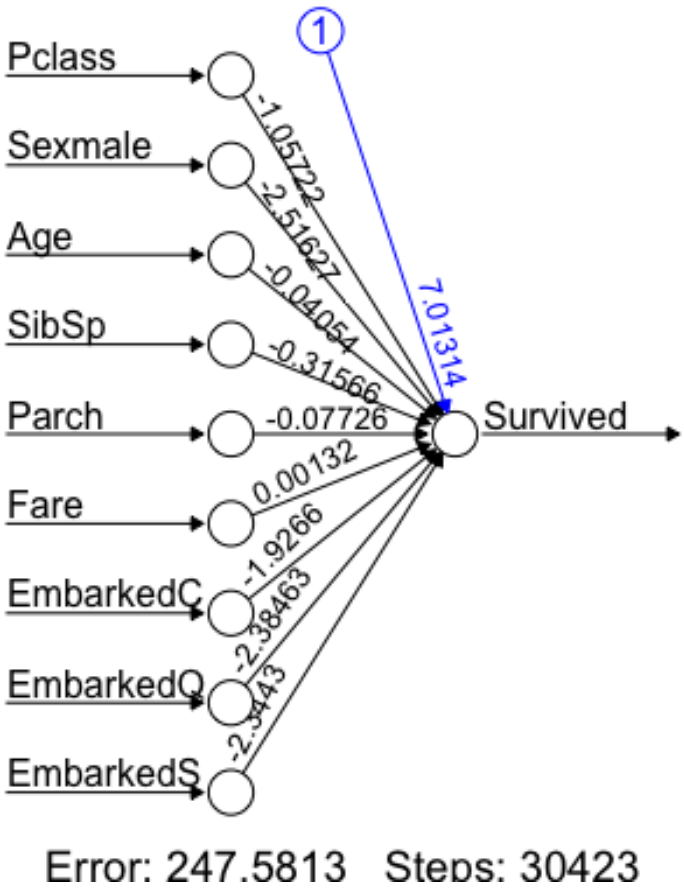
```
# confusion matrix
table(predicted.classes_02, test.data$Survived)
```

```
##
## predicted.classes_02  0  1
##                    0 89 21
##                    1  9 59
```

```
# The overall accuracy of the test data is 0.8315
# The sensitivity of the test data is 0.7375
# The specificity of the test data is 0.9082
```

Question 03

```
set.seed(123)
model_03 <- neuralnet(Survived ~ ., data = train.data, hidden = 0, err.fct =
"ce", linear.output = F)
plot(model_03, rep = "best")
```




```
probabilities_03 <- model_03 %>% predict(test.data) %>% as.vector()
predicted.classes_03 <- ifelse(probabilities_03 > 0.5, 1, 0)
confusionMatrix(factor(predicted.classes_03), factor(test.data$Survived), pos
itive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 89 21
##          1  9 59
##
##                Accuracy : 0.8315
##                  95% CI : (0.7682, 0.8833)
##     No Information Rate : 0.5506
##     P-Value [Acc > NIR] : 2.015e-15
##
##                   Kappa : 0.6547
##
##  Mcnemar's Test P-Value : 0.04461
##
##             Sensitivity : 0.7375
##             Specificity : 0.9082
```

```
##          Pos Pred Value : 0.8676
##          Neg Pred Value : 0.8091
##              Prevalence : 0.4494
##          Detection Rate : 0.3315
##    Detection Prevalence : 0.3820
##       Balanced Accuracy : 0.8228
##
##        'Positive' Class : 1
##
```

```
# confusion matrix
table(predicted.classes_03, test.data$Survived)
```

```
##
## predicted.classes_03  0  1
##                    0 89 21
##                    1  9 59
```

```
# The overall accuracy of the test data is 0.8315
# The sensitivity of the test data is 0.7375
# The specificity of the test data is 0.9082
```

Question 04

```
set.seed(123)
model_04 <- glm(Survived ~ ., family = binomial, data = train.data)
# The fitted logistic regression model coefficients obtained using the traini
ng data:
summary(model_04)$coefficients
```

```
##                 Estimate   Std. Error      z value     Pr(>|z|)
## (Intercept)  16.061380335 5.354114e+02   0.02999820 9.760685e-01
## Pclass       -1.057949923 1.786500e-01  -5.92191475 3.182146e-09
## Sexmale      -2.516036754 2.467277e-01 -10.19762748 2.031750e-24
## Age          -0.040580904 9.053326e-03  -4.48243050 7.379768e-06
## SibSp        -0.315739290 1.437562e-01  -2.19635243 2.806673e-02
## Parch        -0.077049121 1.317589e-01  -0.58477376 5.586999e-01
## Fare          0.001308881 2.558554e-03   0.51157054 6.089516e-01
## EmbarkedC   -10.972030789 5.354113e+02  -0.02049271 9.836503e-01
## EmbarkedQ   -11.429883394 5.354116e+02  -0.02134785 9.829682e-01
## EmbarkedS   -11.389771281 5.354113e+02  -0.02127294 9.830279e-01
```

```
# The results of the CE loss neutral network without hidden layer are similar
with the logistic regression model.

probabilities_04 <- model_04 %>% predict(test.data, type = "response")
predicted.classes_04 <- ifelse(probabilities_04 > 0.5, 1, 0)
confusionMatrix(factor(predicted.classes_04), factor(test.data$Survived), pos
itive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 89 21
##          1  9 59
##
##                Accuracy : 0.8315
##                  95% CI : (0.7682, 0.8833)
##     No Information Rate : 0.5506
##     P-Value [Acc > NIR] : 2.015e-15
##
##                   Kappa : 0.6547
##
##  Mcnemar's Test P-Value : 0.04461
##
##             Sensitivity : 0.7375
##             Specificity : 0.9082
##          Pos Pred Value : 0.8676
##          Neg Pred Value : 0.8091
##              Prevalence : 0.4494
##          Detection Rate : 0.3315
##    Detection Prevalence : 0.3820
##       Balanced Accuracy : 0.8228
##
##        'Positive' Class : 1
##
```

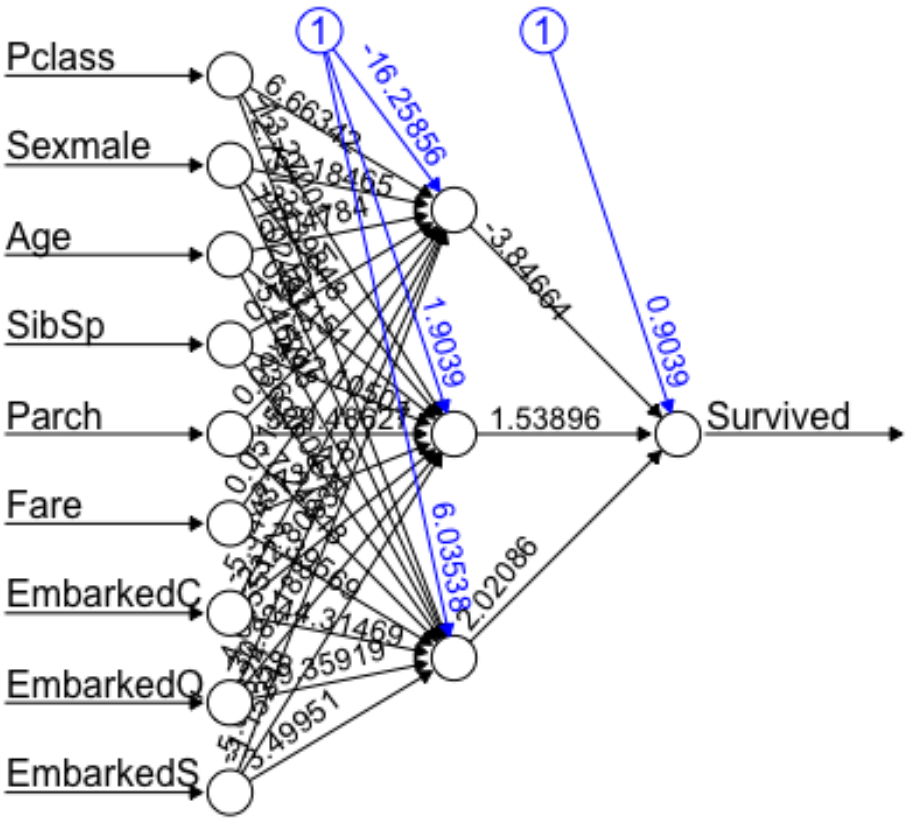
```
# confusion matrix
table(predicted.classes_04, test.data$Survived)
```

```
##
## predicted.classes_04  0  1
##                    0 89 21
##                    1  9 59
```

```
# The overall accuracy of the test data is 0.8315
# The sensitivity of the test data is 0.7375
# The specificity of the test data is 0.9082
```

Question 05

```
model_05 <- neuralnet(Survived ~ ., data = train.data, hidden = 3, err.fct =
"sse", linear.output = F)
plot(model_05, rep = "best")
```




```
probabilities_05 <- model_05 %>% predict(test.data) %>% as.vector()
predicted.classes_05 <- ifelse(probabilities_05 > 0.5, 1, 0)
confusionMatrix(factor(predicted.classes_05), factor(test.data$Survived), pos
itive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 91 20
##          1  7 60
##
##                Accuracy : 0.8483
##                  95% CI : (0.787, 0.8976)
##     No Information Rate : 0.5506
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6889
##
##  Mcnemar's Test P-Value : 0.02092
##
##             Sensitivity : 0.7500
##             Specificity : 0.9286
```

```
##          Pos Pred Value : 0.8955
##          Neg Pred Value : 0.8198
##              Prevalence : 0.4494
##          Detection Rate : 0.3371
##    Detection Prevalence : 0.3764
##       Balanced Accuracy : 0.8393
##
##        'Positive' Class : 1
##
```

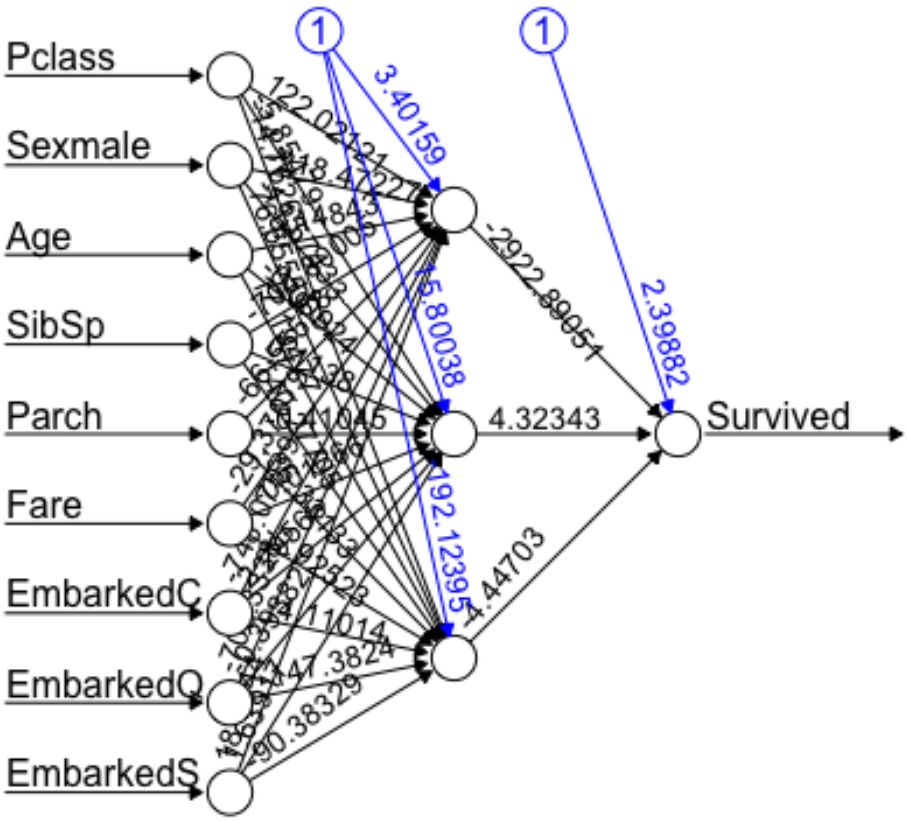
```
# confusion matrix
table(predicted.classes_05, test.data$Survived)
```

```
##
## predicted.classes_05  0  1
##                    0 91 20
##                    1  7 60
```

```
# The overall accuracy of the test data is 0.8258
# The sensitivity of the test data is 0.6625
# The specificity of the test data is 0.9592

# The prediction with hidden layer is similar performance with no hidden laye
r
```

Question 06

```
model_06 <- neuralnet(Survived ~ ., data = train.data, hidden = 3, err.fct =
"ce", linear.output = F)
plot(model_06, rep = "best")
```




```
probabilities_06 <- model_06 %>% predict(test.data) %>% as.vector()
predicted.classes_06 <- ifelse(probabilities_06 > 0.5, 1, 0)
confusionMatrix(factor(predicted.classes_06), factor(test.data$Survived), pos
itive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 96 27
##          1  2 53
##
##                Accuracy : 0.8371
##                  95% CI : (0.7745, 0.8881)
##     No Information Rate : 0.5506
##     P-Value [Acc > NIR] : 4.907e-16
##
##                   Kappa : 0.6611
##
##  Mcnemar's Test P-Value : 8.324e-06
##
##             Sensitivity : 0.6625
##             Specificity : 0.9796
```

```
##          Pos Pred Value : 0.9636
##          Neg Pred Value : 0.7805
##              Prevalence : 0.4494
##          Detection Rate : 0.2978
##    Detection Prevalence : 0.3090
##       Balanced Accuracy : 0.8210
##
##        'Positive' Class : 1
##
```

```
# confusion matrix
table(predicted.classes_06, test.data$Survived)
```

```
##
## predicted.classes_06  0  1
##                    0 96 27
##                    1  2 53
```

```
# The overall accuracy of the test data is 0.8371
# The sensitivity of the test data is 0.6625
# The specificity of the test data is 0.9796

# The prediction with hidden layer is similar performance with no hidden laye
r
```