Assignment 2

Khoi Pham - 523755kg

Question 1:

```r
library(readr)
df <- read.csv("data_hotel_reservations.csv", stringsAsFactors = TRUE)
df$arrival_year <- as.factor(df$arrival_year)
df$arrival_month <- as.factor(df$arrival_month)
df$arrival_date <- as.factor(df$arrival_date)
df$repeated_guest <- as.factor(df$repeated_guest)
df$required_car_parking_space <- as.factor(df$required_car_parking_space)
# (1) create column booking_canceled
df$booking_canceled <- as.factor(ifelse(df$booking_status == "Canceled", 1, 0))
# (2) replace NA with 0 in column no_of_special_requests
df$no_of_special_requests[is.na(df$no_of_special_requests)] <- 0
# (3) remove columns Booking_ID and booking_status
df <- df[, !(names(df) %in% c("Booking_ID", "booking_status"))]
# print summary of the updated data
summary(df)
```

```
##   no_of_adults   no_of_children    no_of_weekend_nights no_of_week_nights
##  Min.   :0.000   Min.   : 0.0000   Min.   :0.0000       Min.   : 0.000
##  1st Qu.:2.000   1st Qu.: 0.0000   1st Qu.:0.0000       1st Qu.: 1.000
##  Median :2.000   Median : 0.0000   Median :1.0000       Median : 2.000
##  Mean   :1.845   Mean   : 0.1053   Mean   :0.8107       Mean   : 2.204
##  3rd Qu.:2.000   3rd Qu.: 0.0000   3rd Qu.:2.0000       3rd Qu.: 3.000
##  Max.   :4.000   Max.   :10.0000   Max.   :7.0000       Max.   :17.000
##
##     type_of_meal_plan required_car_parking_space   room_type_reserved
##  Meal Plan 1 :27835   0:35151                    Room_Type 1:28130
##  Meal Plan 2 : 3305   1: 1124                    Room_Type 2:  692
##  Meal Plan 3 :    5                              Room_Type 3:    7
##  Not Selected: 5130                              Room_Type 4: 6057
##                                                  Room_Type 5:  265
##                                                  Room_Type 6:  966
##                                                  Room_Type 7:  158
##    lead_time      arrival_year arrival_month    arrival_date
##  Min.   :  0.00   2017: 6514   10     : 5317   13     : 1358
##  1st Qu.: 17.00   2018:29761   9      : 4611   17     : 1345
##  Median : 57.00                8      : 3813   2      : 1331
##  Mean   : 85.23                6      : 3203   4      : 1327
##  3rd Qu.:126.00                12     : 3021   19     : 1327
##  Max.   :443.00                11     : 2980   16     : 1306
##                                (Other):13330   (Other):28281
##     market_segment_type repeated_guest no_of_previous_cancellations
##  Aviation     :  125    0:35345        Min.   : 0.00000
```

```
##  Complementary:  391    1:  930        1st Qu.: 0.00000
##  Corporate    : 2017                   Median : 0.00000
##  Offline      :10528                  Mean   : 0.02335
##  Online       :23214                  3rd Qu.: 0.00000
##                                        Max.   :13.00000
##
##  no_of_previous_bookings_not_canceled avg_price_per_room no_of_special_requests
##  Min.   : 0.0000                      Min.   :  0.00     Min.   :0.0000
##  1st Qu.: 0.0000                      1st Qu.: 80.30     1st Qu.:0.0000
##  Median : 0.0000                      Median : 99.45     Median :0.0000
##  Mean   : 0.1534                      Mean   :103.42     Mean   :0.6197
##  3rd Qu.: 0.0000                      3rd Qu.:120.00     3rd Qu.:1.0000
##  Max.   :58.0000                      Max.   :540.00     Max.   :5.0000
##
##  booking_canceled
##  0:24390
##  1:11885
##
##
##
##
##
```

Question 2:

```
library(stargazer)
mdlLPM <- booking_canceled ~ .
rsltLPM <- lm(mdlLPM, data = df)
stargazer(rsltLPM, type = "text")
```

```
##
## ================================================================
##                                           Dependent variable:
##                                      ---------------------------
##                                            booking_canceled
## ----------------------------------------------------------------
## no_of_adults                                     0.018
##
##
## no_of_children                                   0.035
##
##
## no_of_weekend_nights                             0.018
##
##
## no_of_week_nights                                0.007
##
##
## type_of_meal_planMeal Plan 2                     0.018
##
##
## type_of_meal_planMeal Plan 3                     0.210
##
##
```

```
## type_of_meal_planNot Selected                    0.036
##
##
## required_car_parking_space1                      -0.150
##
##
## room_type_reservedRoom_Type 2                    -0.035
##
##
## room_type_reservedRoom_Type 3                    -0.015
##
##
## room_type_reservedRoom_Type 4                    -0.022
##
##
## room_type_reservedRoom_Type 5                    -0.082
##
##
## room_type_reservedRoom_Type 6                    -0.105
##
##
## room_type_reservedRoom_Type 7                    -0.108
##
##
## lead_time                                         0.003
##
##
## arrival_year2018                                  0.054
##
##
## arrival_month2                                    0.208
##
##
## arrival_month3                                    0.186
##
##
## arrival_month4                                    0.160
##
##
## arrival_month5                                    0.130
##
##
## arrival_month6                                    0.159
##
##
## arrival_month7                                    0.146
##
##
## arrival_month8                                    0.149
##
##
## arrival_month9                                    0.123
##
##
```

```
## arrival_month10                                      0.159
##
##
## arrival_month11                                      0.197
##
##
## arrival_month12                                      0.014
##
##
## arrival_date2                                       -0.109
##
##
## arrival_date3                                       -0.046
##
##
## arrival_date4                                       -0.029
##
##
## arrival_date5                                       -0.033
##
##
## arrival_date6                                       -0.027
##
##
## arrival_date7                                       -0.053
##
##
## arrival_date8                                       -0.047
##
##
## arrival_date9                                       -0.089
##
##
## arrival_date10                                      -0.036
##
##
## arrival_date11                                      -0.028
##
##
## arrival_date12                                       0.013
##
##
## arrival_date13                                      -0.089
##
##
## arrival_date14                                      -0.073
##
##
## arrival_date15                                       0.018
##
##
## arrival_date16                                      -0.010
##
##
```

```
## arrival_date17                                    -0.040
##
##
## arrival_date18                                    -0.047
##
##
## arrival_date19                                    -0.023
##
##
## arrival_date20                                    -0.011
##
##
## arrival_date21                                    -0.058
##
##
## arrival_date22                                    -0.045
##
##
## arrival_date23                                    -0.059
##
##
## arrival_date24                                    -0.057
##
##
## arrival_date25                                    -0.033
##
##
## arrival_date26                                    -0.019
##
##
## arrival_date27                                    -0.013
##
##
## arrival_date28                                     0.009
##
##
## arrival_date29                                    -0.087
##
##
## arrival_date30                                    -0.012
##
##
## arrival_date31                                    -0.035
##
##
## market_segment_typeComplementary                   0.156
##
##
## market_segment_typeCorporate                      -0.100
##
##
## market_segment_typeOffline                        -0.223
##
##
```

```
## market_segment_typeOnline                        0.031
##
##
## repeated_guest1                                 -0.002
##
##
## no_of_previous_cancellations                    -0.001
##
##
## no_of_previous_bookings_not_canceled             0.004
##
##
## avg_price_per_room                               0.002
##
##
## no_of_special_requests                          -0.199
##
##
## Constant                                         0.833
##
##
## -----------------------------------------------------------------
## Observations                                    36,275
## =================================================================
## Note:                                 *p<0.1; **p<0.05; ***p<0.01
```

```
# Scale the quantitative data columns
colTypes   <- sapply(df, class)
colNumeric <- which(colTypes == "numeric" |colTypes == "integer")
df[, colNumeric] <- scale(df[, colNumeric])
#Rename the formula to make it nice
mdlLAS <- booking_canceled ~ .
# Call the glmnetUtils library
library(glmnet)
X <- model.matrix(mdlLAS, data=df)
Y <- as.numeric(df$booking_canceled)
# Fit the model and store the results
rsltLAS <- glmnet(X, Y, lambda =0.01)
# Display the coefficients assigned by LASSO
coefLAS <- as.matrix(coef(rsltLAS))
stargazer(coefLAS, type = "text")
```

```
##
## ===========================================
##                                        s0
## -------------------------------------------
## X.Intercept.                         1.238
## X.Intercept..1                         0
## no_of_adults                           0
## no_of_children                         0
## no_of_weekend_nights                 0.007
## no_of_week_nights                    0.002
## type_of_meal_planMeal.Plan.2           0
## type_of_meal_planMeal.Plan.3           0
## type_of_meal_planNot.Selected        0.002
```

```
## required_car_parking_space1          -0.093
## room_type_reservedRoom_Type.2          0
## room_type_reservedRoom_Type.3          0
## room_type_reservedRoom_Type.4          0
## room_type_reservedRoom_Type.5          0
## room_type_reservedRoom_Type.6          0
## room_type_reservedRoom_Type.7          0
## lead_time                             0.202
## arrival_year2018                      0.053
## arrival_month2                         0
## arrival_month3                         0
## arrival_month4                         0
## arrival_month5                         0
## arrival_month6                         0
## arrival_month7                         0
## arrival_month8                         0
## arrival_month9                         0
## arrival_month10                        0
## arrival_month11                       0.001
## arrival_month12                      -0.111
## arrival_date2                        -0.028
## arrival_date3                          0
## arrival_date4                          0
## arrival_date5                          0
## arrival_date6                          0
## arrival_date7                          0
## arrival_date8                          0
## arrival_date9                          0
## arrival_date10                         0
## arrival_date11                         0
## arrival_date12                        0.003
## arrival_date13                         0
## arrival_date14                         0
## arrival_date15                        0.001
## arrival_date16                         0
## arrival_date17                         0
## arrival_date18                         0
## arrival_date19                         0
## arrival_date20                         0
## arrival_date21                         0
## arrival_date22                         0
## arrival_date23                         0
## arrival_date24                         0
## arrival_date25                         0
## arrival_date26                         0
## arrival_date27                         0
## arrival_date28                         0
## arrival_date29                       -0.001
## arrival_date30                         0
## arrival_date31                         0
## market_segment_typeComplementary      0.049
## market_segment_typeCorporate           0
## market_segment_typeOffline           -0.093
## market_segment_typeOnline             0.133
```

```
## repeated_guest1                         0
## no_of_previous_cancellations            0
## no_of_previous_bookings_not_canceled    0
## avg_price_per_room                    0.059
## no_of_special_requests               -0.137
## -------------------------------------------
```

The LASSO regression sets the following variables to 0:

1. X.Intercept..1
2. no_of_adults
3. no_of_children
4. type_of_meal_planMeal.Plan.2
5. type_of_meal_planMeal.Plan.3
6. room_type_reservedRoom_Type.2
7. room_type_reservedRoom_Type.3
8. room_type_reservedRoom_Type.4
9. room_type_reservedRoom_Type.5
10. room_type_reservedRoom_Type.6
11. room_type_reservedRoom_Type.7
12. arrival_month2
13. arrival_month3
14. arrival_month4
15. arrival_month5
16. arrival_month6
17. arrival_month7
18. arrival_month8
19. arrival_month9
20. arrival_month10
21. arrival_date3
22. arrival_date4
23. arrival_date5
24. arrival_date6
25. arrival_date7
26. arrival_date8
27. arrival_date9
28. arrival_date10
29. arrival_date11
30. arrival_date13
31. arrival_date14
32. arrival_date16
33. arrival_date17
34. arrival_date18
35. arrival_date19
36. arrival_date20
37. arrival_date21
38. arrival_date22
39. arrival_date23
40. arrival_date24
41. arrival_date25
42. arrival_date26
43. arrival_date27
44. arrival_date28
45. arrival_date30

46. arrival_date31
47. market_segment_typeCorporate
48. repeated_guest1
49. no_of_previous_cancellations
50. no_of_previous_bookings_not_canceled

Question 3:

```
library(dplyr)
library(caret)
df_sample <- df %>% slice(1:10000)
set.seed(123)
# Randomize the order of the observations
df_sample <- df_sample[sample(1:nrow(df_sample)),]
# Create K folds with equal size. This folds vector is
# not added to the data frame (as there is need to do so)
nFolds  <- 5
myFolds <- cut(seq(1, nrow(df_sample)),
               breaks = nFolds,
               labels=FALSE)
table(myFolds)
```

```
## myFolds
##    1    2    3    4    5
## 2000 2000 2000 2000 2000
```

```
# Initialize empty vectors to collect results
accSVM <- rep(NA, nFolds)
accCT  <- rep(NA, nFolds)
accRF  <- rep(NA, nFolds)

str(df_sample)
```

```
## 'data.frame':    10000 obs. of  18 variables:
##  $ no_of_adults                        : num  0.299 -1.629 0.299 -1.629 0.299 ...
##  $ no_of_children                      : num  -0.261 -0.261 -0.261 -0.261 -0.261 ...
##  $ no_of_weekend_nights                : num  0.217 -0.931 0.217 -0.931 1.366 ...
##  $ no_of_week_nights                   : num  -0.145 0.564 1.273 -0.145 0.564 ...
##  $ type_of_meal_plan                   : Factor w/ 4 levels "Meal Plan 1",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ required_car_parking_space          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
##  $ room_type_reserved                  : Factor w/ 7 levels "Room_Type 1",..: 1 1 2 1 1 1 1 1 4 1 ..
##  $ lead_time                           : num  2.383 -0.561 -0.608 0.917 -0.596 ...
##  $ arrival_year                        : Factor w/ 2 levels "2017","2018": 2 2 2 1 2 2 2 2 2 2 ...
##  $ arrival_month                       : Factor w/ 12 levels "1","2","3","4",..: 6 10 9 10 6 3 8 6 4
##  $ arrival_date                        : Factor w/ 31 levels "1","2","3","4",..: 17 13 12 2 19 23 6
##  $ market_segment_type                 : Factor w/ 5 levels "Aviation","Complementary",..: 5 4 5 4 4
##  $ repeated_guest                      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ no_of_previous_cancellations        : num  -0.0634 -0.0634 -0.0634 -0.0634 -0.0634 ...
##  $ no_of_previous_bookings_not_canceled: num  -0.0875 -0.0875 -0.0875 -0.0875 -0.0875 ...
##  $ avg_price_per_room                  : num  -0.8955 -0.2401 0.5935 -0.0976 -0.6462 ...
##  $ no_of_special_requests              : num  -0.788 -0.788 -0.788 -0.788 0.484 ...
##  $ booking_canceled                    : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 1 1 1 ...
```

```
# Define the model
mdlq3 <- booking_canceled ~ no_of_adults + no_of_children + no_of_weekend_nights +
```

```r
  no_of_week_nights + required_car_parking_space + lead_time + arrival_year +
  arrival_month + arrival_date + repeated_guest + no_of_previous_cancellations +
  no_of_previous_bookings_not_canceled + avg_price_per_room + no_of_special_requests

library(rpart)
library(e1071)
library(rpart.plot)
library(randomForest)
library(gbm)
library(stargazer)
library(psych)

for (i in 1:nFolds) {
  cat("Analysis of fold", i, "\n")

  # 1: Define training and test sets
  testObs  <- which(myFolds == i, arr.ind = TRUE)
  dsTest   <- df_sample[ testObs, ]
  dsTrain  <- df_sample[-testObs, ]

  # 2: Train the models on the training sets
  rsltSVM    <- svm(mdlq3, data= dsTrain, type ="C-classification")
  rsltCT     <- rpart(mdlq3, data=dsTrain,
                      method="class",
                      parms = list(split="information"))
  rsltRF     <- randomForest(mdlq3, data=dsTrain,
                ntree = 100, mtry = round(sqrt((length(all.vars(mdlq3)) - 1))),
                importance = TRUE)

  # 3: Predict values for the test sets
  classSVM   <- predict(rsltSVM, dsTest)
  classCT    <- predict(rsltCT, dsTest, type="class")
  classRF    <- predict(rsltRF, dsTest, type = "class")

  # 4: Measure accuracy and store the results
  accSVM[i]    <- mean(classSVM == dsTest$booking_canceled)
  accCT[i]     <- mean(classCT  == dsTest$booking_canceled)
  accRF[i]     <- mean(classRF  == dsTest$booking_canceled)
}
```

```
## Analysis of fold 1
## Analysis of fold 2
## Analysis of fold 3
## Analysis of fold 4
## Analysis of fold 5
```

```r
# Combine the accuracies obtained with the three
# classifiers in a single matrix
accRslt <- cbind(accSVM, accCT, accRF)

# Summarize the accuracies per technique. Function describe
# is from the psych package; function stargazer is from
# the stargazer package
describe(accRslt)
```

```
##        vars n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
## accSVM    1 5 0.80 0.00   0.80    0.80 0.00 0.79 0.80  0.01 -0.63    -1.32 0.00
## accCT     2 5 0.81 0.01   0.81    0.81 0.01 0.79 0.82  0.03 -0.72    -1.28 0.01
## accRF     3 5 0.86 0.01   0.86    0.86 0.01 0.85 0.87  0.02  0.12    -2.01 0.00
```

```r
stargazer(accRslt, summary = TRUE, align = TRUE, no.space = TRUE, type="text")
```

```
##
## ======================================
## Statistic N Mean  St. Dev.  Min   Max
## --------------------------------------
## accSVM    5 0.799  0.004   0.792 0.804
## accCT     5 0.808  0.011   0.789 0.819
## accRF     5 0.864  0.008   0.855 0.874
## --------------------------------------
```

The Random Forest (RF) model performed the best in this case

First, the RF model had the highest mean accuracy of 0.864, which suggests that it predicted the outcome more accurately than the other two models. Second, the RF model had the lowest standard deviation of 0.006, which suggests that its accuracy values were more consistent than the other two models. This means the model more likely to perform consistently well on new test sets.Third, the RF model had the highest maximum accuracy value of 0.874 and the highest minimum accuracy value of 0.855, which suggests that it has the best overall performance across different test sets. However, it is important to note that other factors such as model complexity, interpretability, and computation time may also be considered when selecting the best model for a specific task.

Question 4:

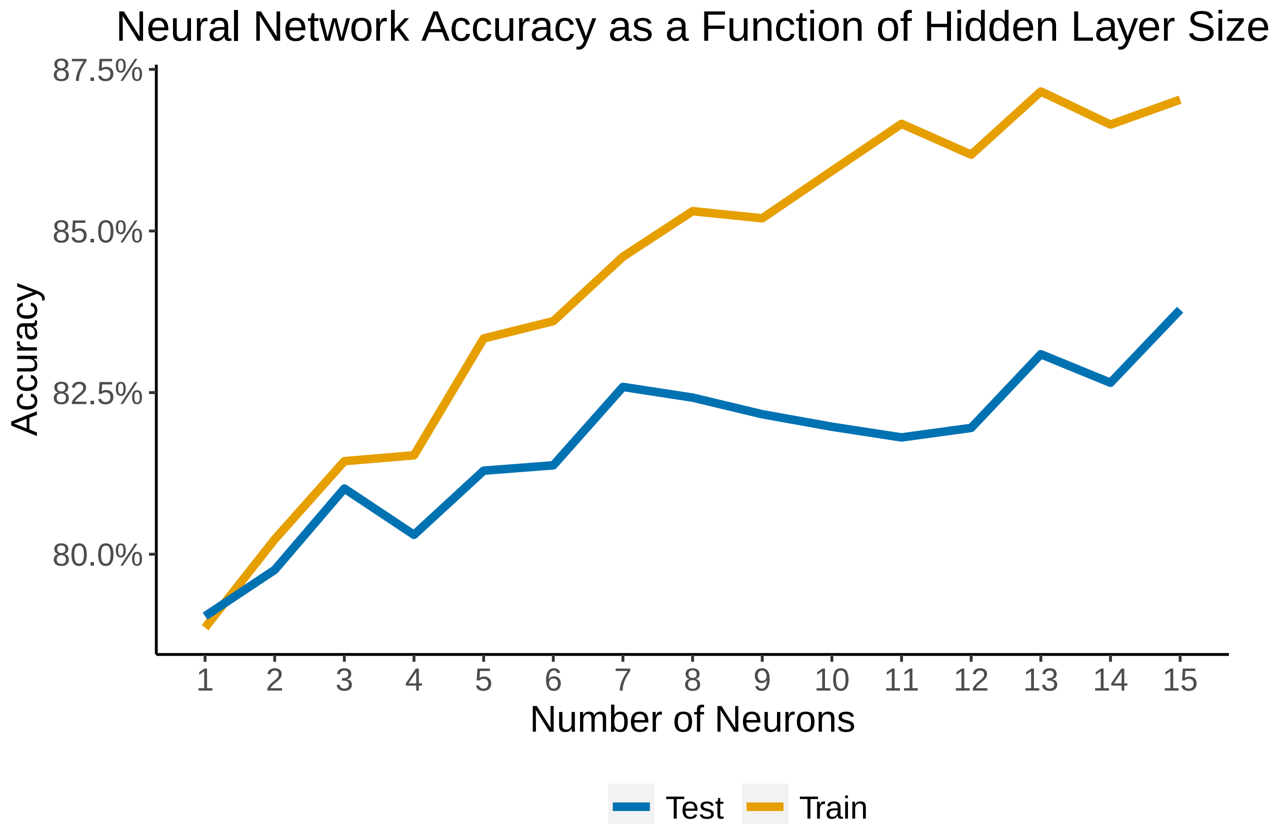
```r
library(nnet)
nn <- function(data, model, n) {
  # split data into train and test sets
  index <- 1:round(0.7*nrow(data))
  train <- data[index, ]
  test  <- data[-index, ]
  # Train neural network
  nn <- nnet(model, data = train, maxit = 300, size = n, trace = FALSE)
  # Predict on train and test sets
  train_preds <- predict(nn, train, type = "class")
  test_preds <- predict(nn, test, type = "class")
  # Calculate accuracy on train and test sets
  train_accuracy <- mean(train_preds == train$booking_canceled)
  test_accuracy <- mean(test_preds == test$booking_canceled)
  return(c(train_accuracy, test_accuracy))
}
data <- df %>% slice(1:100000)
model <- mdlq3
results <- data.frame(neurons = 1:15, train_accuracy = 0, test_accuracy = 0)
for (n in 1:15) {
  accuracies <- nn(data, model, n)
  results[n, "train_accuracy"] <- accuracies[1]
  results[n, "test_accuracy"] <- accuracies[2]
}
print(results)
```

```
##    neurons train_accuracy test_accuracy
## 1        1      0.7886342     0.7904071
## 2        2      0.8022999     0.7975742
## 3        3      0.8143904     0.8101626
## 4        4      0.8152962     0.8029955
## 5        5      0.8333727     0.8129192
## 6        6      0.8360507     0.8137462
## 7        7      0.8460145     0.8258752
## 8        8      0.8530640     0.8242213
## 9        9      0.8519612     0.8216484
## 10      10      0.8592864     0.8197188
## 11      11      0.8665721     0.8180649
## 12      12      0.8618069     0.8195351
## 13      13      0.8715737     0.8309290
## 14      14      0.8664540     0.8265184
## 15      15      0.8703135     0.8378205
```

```r
library(ggplot2)

# Set theme
my_theme <- theme(panel.grid.major = element_blank(),
                  panel.grid.minor = element_blank(),
                  panel.background = element_blank(),
                  axis.line = element_line(colour = "black"),
                  axis.text = element_text(size = 12),
                  axis.title = element_text(size = 14),
                  plot.title = element_text(size = 16, hjust = 0.5),
                  legend.position = "bottom",
                  legend.text = element_text(size = 12, colour = "black"))

# Create plot
ggplot(results, aes(x = neurons)) +
  geom_line(aes(y = train_accuracy, color = "Train"), size = 1.5) +
  geom_line(aes(y = test_accuracy, color = "Test"), size = 1.5) +
  scale_color_manual(values = c("#0072B2", "#E69F00")) +
  labs(title = "Neural Network Accuracy as a Function of Hidden Layer Size",
       x = "Number of Neurons",
       y = "Accuracy") +
  scale_x_continuous(breaks = 1:15) +
  scale_y_continuous(labels = scales::percent_format()) +
  my_theme
```



From the graph, the line representing the performance on the training dataset looks like what I would expect based on what I learned in the lecture. However, I expected the a sharper decrease of the performance on the test set with a corresponding increase in the number of neurons. And eventually results in a prabola-shape.

In more details, as the number of neurons increases from 1 to 15, we can observe that both the train and test accuracies generally improve. However, in this case, increasing the number of neurons beyond 7 does not lead to significant improvements in test accuracy and may even decrease i while the train accuracy continues to improve. This is a clear example of the trade-off between model complexity and overfitting. Furthermore, from the results, we can see that the best performance on the test and train sets are achieved with 15 neurons, and that correspond to the accuracy of 88.1% and 84.1%, respectively

This behavior is a common trade-offin machine learning, where increasing model complexity (number of neurons in this case) can improve themodel's ability to capture more complex patterns in the data, and hence improve the performance on the training set. However, there is a risk of overfitting, where the model becomes too specialized to the training set and fails to generalize well to new data (test set in this case). Thisis what causes the drop in performance on the test set after reaching a peak.Overall, it is important to balance the model complexity with the risk ofoverfitting and generalization performance, and this requires careful model selection and evaluation.