Programming For Data Science Lab Assignment 4

L33-L34

Tejas Rokade

20BDS0033

Q1. Read gender_classification dataset from R and perform following model fitting techniques (Any other datasets also applicable)

a. Logistic Regression

b.Decision Tree

c. Naïve Bayes

d.SVM

e.Random forest

Q2. Compare each of the above models using the following parameters

a.Accuracy

b.Precision

c. Recall

d.Sensitivity

e. Specificity

Ans 1 & 2.

Code:

```
#FUNCTION TO FIND accuracy, precision, recall, Sensitivity, Specificity
classification_report_for_model <- function(conf_matr){
print(paste0("Accuracy: ", (

(conf_matr[1,1]+conf_matr[2,2])/(conf_matr[1,1]+conf_matr[1,2]+conf_matr[2,1]+conf_m
atr[2,2]) )))
  print("")
```

```
  print(paste0("Precision: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[1,2]))))
  print("")
  print(paste0("Recall: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[2,1])) ))
  print("")
  print(paste0("Senstivity: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[2,1])) ))
  print("")
  print(paste0("Specificity: ", (conf_matr[1,1]/(conf_matr[1,1]+conf_matr[1,2]))))

}
Gender_class=read.csv("C:\\Users\\Parshva Maniar\\Downloads\\archive\\Transformed Data Set - Sheet1.csv")
library(superml)
label=LabelEncoder$new()
Gender_class$Favorite.Color=label$fit_transform(Gender_class$Favorite.Color)
label=LabelEncoder$new()
Gender_class$Favorite.Music.Genre=label$fit_transform(Gender_class$Favorite.Music.Genre)
label=LabelEncoder$new()
Gender_class$Favorite.Beverage=label$fit_transform(Gender_class$Favorite.Beverage)
label=LabelEncoder$new()
Gender_class$Favorite.Soft.Drink     =label$fit_transform(Gender_class$Favorite.Soft.Drink
)
label=LabelEncoder$new()
Gender_class$Gender=label$fit_transform(Gender_class$Gender)
Gender_class$Gender = factor(Gender_class$Gender, levels = c(0, 1))
head(Gender_class)
library(caTools)
#splitting data into train and test
```

```
split=sample.split(Gender_class$Gender,SplitRatio =0.7)
train=subset(Gender_class,split==TRUE)
test=subset(Gender_class,split==FALSE)
library(e1071)
train[-5]=scale(train[-5])
test[-5]=scale(test[-5])
#LOGISTIC
model_log_reg =glm(Gender ~ ., data = train,family = "binomial")
model_log_reg
#Prediction
pred_lr = predict(model_log_reg, newdata = test[-5],type="response")
pred_lr <- ifelse(pred_lr>0.5,1,0)
pred_lr
# Making the Confusion Matrix
conf_matr = table(test[,5], pred_lr)
print(conf_matr)
classification_report_for_model(conf_matr)
#Desicion tree
library(party)
Desc_tree = ctree(Gender~., data = train)
pred_desc_tree = predict(Desc_tree, newdata = test[-5])
pred_desc_tree
# Making the Confusion Matrix
conf_matr = table(test[,5], pred_desc_tree)
print(conf_matr)
classification_report_for_model(conf_matr)
#Naive Bayes
naive_bayes =naiveBayes(Gender ~ ., data = train)
```

```
naive_bayes
#Prediction
pred_nb = predict(naive_bayes, newdata = test[-5])
# Making the Confusion Matrix
conf_matr = table(test[,5], pred_nb)
print(conf_matr)
classification_report_for_model(conf_matr)
#SVM
svm_model = svm(formula = Gender ~ ., data = train, type = 'C-classification', kernel
            = 'linear')
svm_model
#Prediction
pred_svm = predict(svm_model, newdata = test[-5])
#Confusion Matrix
conf_matr = table(test[,5], pred_svm)
print(conf_matr)
classification_report_for_model(conf_matr)
#Random Forest
library(randomForest)
randomforest_model = randomForest(x =train[-5], y =train$Gender, ntree = 500)
randomforest_model
#predict
pred_rft = predict(randomforest_model, newdata = test[-5])
#Confusion Matrix
conf_matr = table(test[,5], pred_rft)
conf_matr
classification_report_for_model(conf_matr)
```

Output:

Console Terminal Background Jobs

R 4.2.1 · ~/

```
> #FUNCTION TO FIND accuracy, precision, recall, Sensitivity, Specificity
> classification_report_for_model <- function(conf_matr){
+   print(paste0("Accuracy: ", (
+     (conf_matr[1,1]+conf_matr[2,2])/(conf_matr[1,1]+conf_matr[1,2]+conf_matr[2,1]+conf_matr[2,2]) )))
+   print("")
+   print(paste0("Precision: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[1,2]))))
+   print("")
+   print(paste0("Recall: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[2,1])) ))
+   print("")
+   print(paste0("Senstivity: ", (conf_matr[2,2]/(conf_matr[2,2]+conf_matr[2,1])) ))
+   print("")
+   print(paste0("Specificity: ", (conf_matr[1,1]/(conf_matr[1,1]+conf_matr[1,2]))))
+
+ }
> Gender_class=read.csv("C:\\Users\\Parshva Maniar\\Downloads\\archive\\Transformed Data Set - Sheet1.csv")
> library(superml)
> label=LabelEncoder$new()
> Gender_class$Favorite.Color=label$fit_transform(Gender_class$Favorite.Color)
> label=LabelEncoder$new()
> Gender_class$Favorite.Music.Genre=label$fit_transform(Gender_class$Favorite.Music.Genre)
> label=LabelEncoder$new()
> Gender_class$Favorite.Beverage=label$fit_transform(Gender_class$Favorite.Beverage)
> label=LabelEncoder$new()
> Gender_class$Favorite.Soft.Drink =label$fit_transform(Gender_class$Favorite.Soft.Drink
+ )
> label=LabelEncoder$new()
> Gender_class$Gender=label$fit_transform(Gender_class$Gender)
> Gender_class$Gender = factor(Gender_class$Gender, levels = c(0, 1))
> head(Gender_class)
  Favorite.Color Favorite.Music.Genre Favorite.Beverage Favorite.Soft.Drink Gender
1              0                    0                 0                   0      0
2              1                    1                 0                   1      0
3              2                    0                 1                   1      0
4              2                    2                 2                   2      0
5              0                    0                 0                   1      0
6              2                    3                 3                   2      0
```

Console Terminal Background Jobs

R 4.2.1 · ~/

```
> library(caTools)
> #splitting data into train and test
> split=sample.split(Gender_class$Gender,SplitRatio =0.7)
> train=subset(Gender_class,split==TRUE)
> test=subset(Gender_class,split==FALSE)
> library(e1071)
> train[-5]=scale(train[-5])
> test[-5]=scale(test[-5])
> #LOGISTIC
> model_log_reg =glm(Gender ~ ., data = train,family = "binomial")
> model_log_reg

Call:  glm(formula = Gender ~ ., family = "binomial", data = train)

Coefficients:
         (Intercept)        Favorite.Color  Favorite.Music.Genre     Favorite.Beverage   Favorite.Soft.Drink
             0.01176              -0.24580              -0.10460               0.06401               0.62048

Degrees of Freedom: 45 Total (i.e. Null);  41 Residual
Null Deviance:      63.77
Residual Deviance: 59.83        AIC: 69.83
> #Prediction
> pred_lr = predict(model_log_reg, newdata = test[-5],type="response")
> pred_lr <- ifelse(pred_lr>0.5,1,0)
> pred_lr
 4  9 12 14 15 21 27 28 29 33 38 39 41 42 43 51 53 55 64 65
 0  0  0  0  1  1  1  1  0  0  1  1  0  0  0  0  1  1  1  0
> # Making the Confusion Matrix
> conf_matr = table(test[,5], pred_lr)
> print(conf_matr)
   pred_lr
    0 1
  0 6 4
  1 5 5
> classification_report_for_model(conf_matr)
[1] "Accuracy: 0.55"
[1] ""
[1] "Precision: 0.555555555555556"
[1] ""
[1] "Recall: 0.5"
[1] ""
[1] "Senstivity: 0.5"
[1] ""
[1] "Specificity: 0.6"
> #Desicion tree
> library(party)
> Desc_tree = ctree(Gender~., data = train)
> pred_desc_tree = predict(Desc_tree, newdata = test[-5])
> pred_desc_tree
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Levels: 0 1
```

Console Terminal Background Jobs

R 4.2.1 · ~/

```
> conf_matr = table(test[,5], pred_desc_tree)
> print(conf_matr)
   pred_desc_tree
     0  1
  0 10  0
  1 10  0
> classification_report_for_model(conf_matr)
[1] "Accuracy: 0.5"
[1] ""
[1] "Precision: NaN"
[1] ""
[1] "Recall: 0"
[1] ""
[1] "Senstivity: 0"
[1] ""
[1] "Specificity: 1"
> #Naive Bayes
> naive_bayes =naiveBayes(Gender ~ ., data = train)
> naive_bayes

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  0   1 
0.5 0.5 

Conditional probabilities:
   Favorite.Color
Y           [,1]     [,2]
  0  0.06922995 1.007715
  1 -0.06922995 1.009923

   Favorite.Music.Genre
Y           [,1]     [,2]
  0  0.02049915 0.950848
  1 -0.02049915 1.067925

   Favorite.Beverage
Y           [,1]      [,2]
  0 -0.07356222 1.0306934
  1  0.07356222 0.9858047

   Favorite.Soft.Drink
Y          [,1]      [,2]
  0 -0.2530263 0.7778618
  1  0.2530263 1.1430314
```

```
> #Prediction
> pred_nb = predict(naive_bayes, newdata = test[-5])
> # Making the Confusion Matrix
> conf_matr = table(test[,5], pred_nb)
> print(conf_matr)
   pred_nb
    0 1
  0 7 3
  1 5 5
> classification_report_for_model(conf_matr)
[1] "Accuracy: 0.6"
[1] ""
[1] "Precision: 0.625"
[1] ""
[1] "Recall: 0.5"
[1] ""
[1] "Senstivity: 0.5"
[1] ""
[1] "Specificity: 0.7"
> #SVM
> svm_model = svm(formula = Gender ~ ., data = train, type = 'C-classification', kernel
+                 = 'linear')
> svm_model

Call:
svm(formula = Gender ~ ., data = train, type = "C-classification", kernel = "linear")


Parameters:
   SVM-Type:  C-classification 
 SVM-Kernel:  linear 
       cost:  1 

Number of Support Vectors:  42

> #Prediction
> pred_svm = predict(svm_model, newdata = test[-5])
> #Confusion Matrix
> conf_matr = table(test[,5], pred_svm)
> print(conf_matr)
   pred_svm
    0 1
  0 7 3
  1 6 4
```

```
> classification_report_for_model(conf_matr)
[1] "Accuracy: 0.55"
[1] ""
[1] "Precision: 0.571428571428571"
[1] ""
[1] "Recall: 0.4"
[1] ""
[1] "Senstivity: 0.4"
[1] ""
[1] "Specificity: 0.7"
> #Random Forest
> library(randomForest)
randomForest 4.7-1.1
Type rfNews() to see new features/changes/bug fixes.
> randomforest_model = randomForest(x =train[-5], y =train$Gender, ntree = 500)
> randomforest_model

Call:
 randomForest(x = train[-5], y = train$Gender, ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 41.3%
Confusion matrix:
   0  1 class.error
0 16  7   0.3043478
1 12 11   0.5217391
> #predict
> pred_rft = predict(randomforest_model, newdata = test[-5])
> #Confusion Matrix
> conf_matr = table(test[,5], pred_rft)
> conf_matr
   pred_rft
    0 1
  0 6 4
  1 6 4
> classification_report_for_model(conf_matr)
[1] "Accuracy: 0.5"
[1] ""
[1] "Precision: 0.5"
[1] ""
[1] "Recall: 0.4"
[1] ""
[1] "Senstivity: 0.4"
[1] ""
[1] "Specificity: 0.6"
> |
```

Q3. Perform K Means Clustering on IRIS dataset

Ans 3.

CODE:

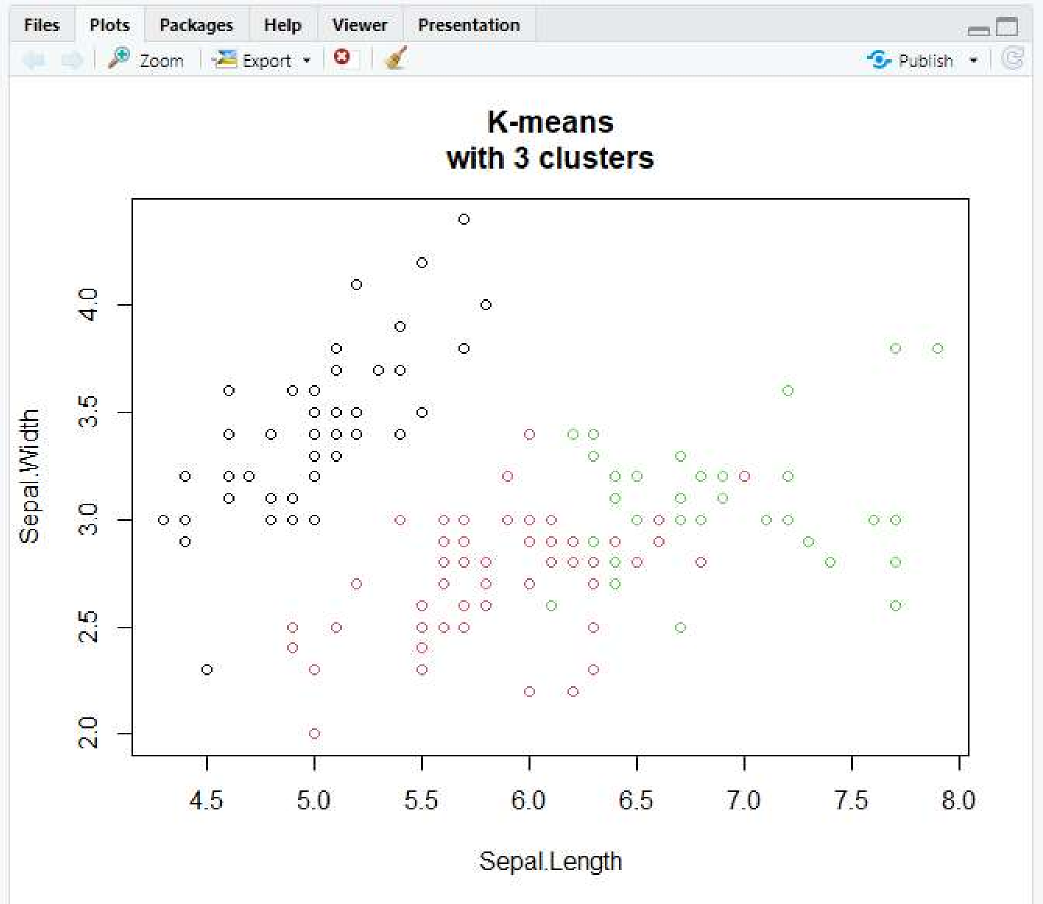
```
# Removing initial label of

x <- iris[, -5]

head(x)

# Fitting the clustering Model to the dataset

set.seed(240) # Setting seed

kmeans_clust <- kmeans(x, centers = 3, nstart = 20)

kmeans_clust

kmeans_clust$cluster

# Confusion Matrix

cm <- table(iris$Species, kmeans_clust$cluster)

cm

# Visualization
```

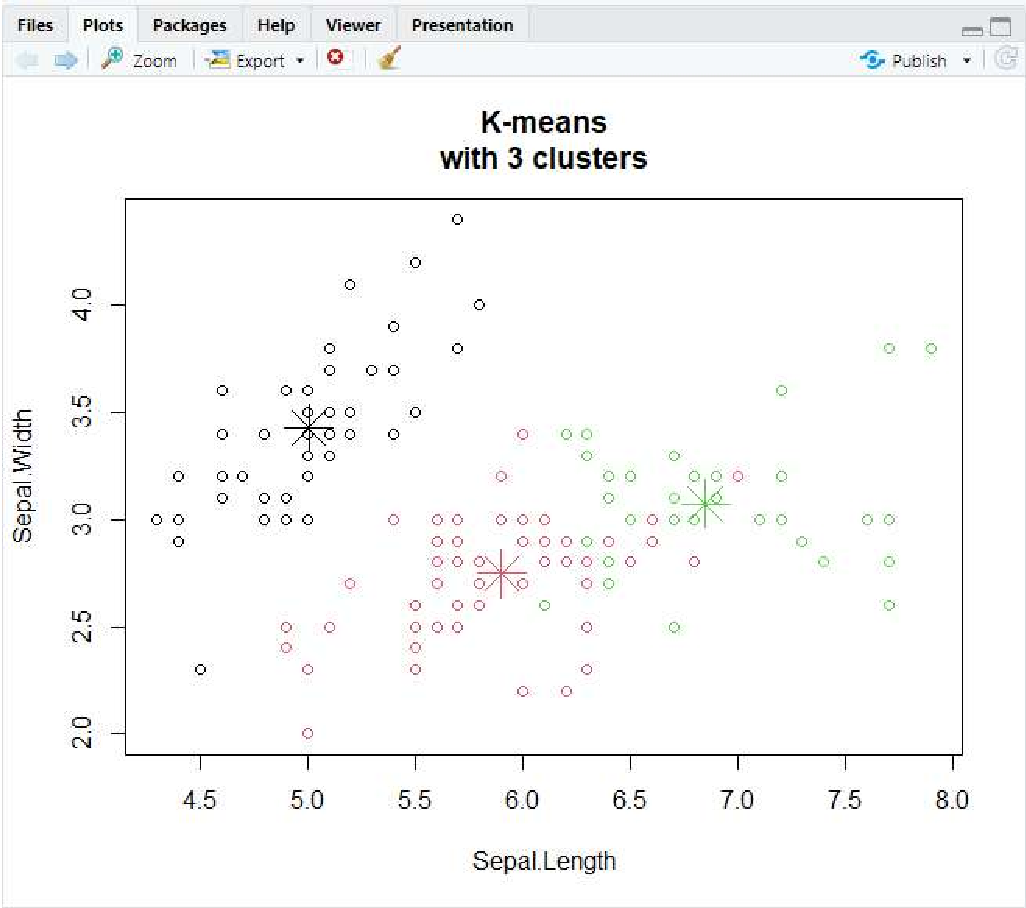
```
plot(x[c("Sepal.Length", "Sepal.Width")], col = kmeans_clust$cluster, main = "K-means
with 3 clusters")
# Plotting clusters with centres
kmeans_clust$centers
kmeans_clust$centers[, c("Sepal.Length", "Sepal.Width")]
points(kmeans_clust$centers[, c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8,
    cex = 3)
# Visualizing clusters
y <- kmeans_clust$cluster
library(cluster)
clusplot(x[, c("Sepal.Length", "Sepal.Width")],y,lines = 0,shade = TRUE, color = TRUE,
     labels = 2, plotchar = FALSE, span = TRUE, main = paste("K-means Cluster iris"),
xlab
     = 'Sepal.Length', ylab = 'Sepal.Width')
```

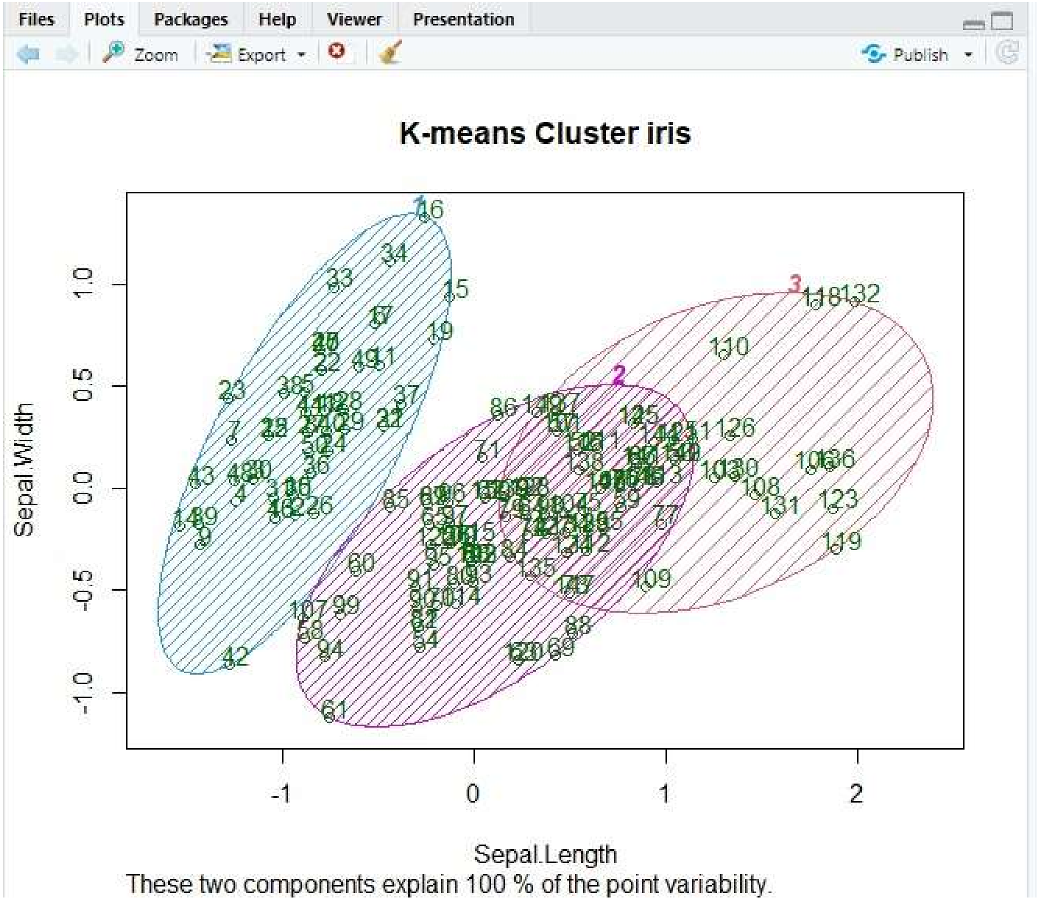
Output:

```
Console  Terminal ×  Background Jobs ×
R 4.2.1 · ~/
> # Removing initial label of
> x <- iris[, -5]
> head(x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
> # Fitting the clustering Model to the dataset
> set.seed(240) # Setting seed
> kmeans_clust <- kmeans(x, centers = 3, nstart = 20)
> kmeans_clust
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [71] 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3
[141] 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 15.15100 39.82097 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
> kmeans_clust$cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [71] 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3
[141] 3 3 2 3 3 3 2 3 3 2
> kmeans_clust$cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [71] 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3
[141] 3 3 2 3 3 3 2 3 3 2
> # Confusion Matrix
> cm <- table(iris$Species, kmeans_clust$cluster)
> cm

              1  2  3
  setosa     50  0  0
  versicolor  0 48  2
  virginica   0 14 36
```

```
> # Visualization
> plot(x[c("Sepal.Length", "Sepal.Width")], col = kmeans_clust$cluster, main = "K-means
+ with 3 clusters")
> # Plotting clusters with centres
> kmeans_clust$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053
> kmeans_clust$centers[, c("Sepal.Length", "Sepal.Width")]
  Sepal.Length Sepal.Width
1     5.006000    3.428000
2     5.901613    2.748387
3     6.850000    3.073684
> points(kmeans_clust$centers[, c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8,
+        cex = 3)
> # Visualizing clusters
> y <- kmeans_clust$cluster
> library(cluster)
> clusplot(x[, c("Sepal.Length", "Sepal.Width")],y,lines = 0,shade = TRUE, color = TRUE,
+          labels = 2, plotchar = FALSE, span = TRUE, main = paste("K-means Cluster iris"), xlab
+          = 'Sepal.Length', ylab = 'Sepal.Width')
> |
```


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
K-means
with 3 clusters
Sepal.Width
Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
2.0
2.5
3.0
3.5
4.0


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
K-means
with 3 clusters
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0



Q4. Perform Hierarchical clustering in mtcars dataset

Ans 4

Code:

```
mtcars_temp <- dist(mtcars, method = 'euclidean')

mtcars_temp

set.seed(240)

hierarchial_cluster_model <- hclust(mtcars_temp, method = "average")

hierarchial_cluster_model

# Plotting dendrogram

plot(hierarchial_cluster_model)

# Cutting tree by height
```

abline(h = 110, col = "green")

Cutting by no. of clusters

final_fit <- cutree(hierarchial_cluster_model, k = 3)

final_fit

table(final_fit)

rect.hclust(hierarchial_cluster_model, k = 3, border = "yellow")

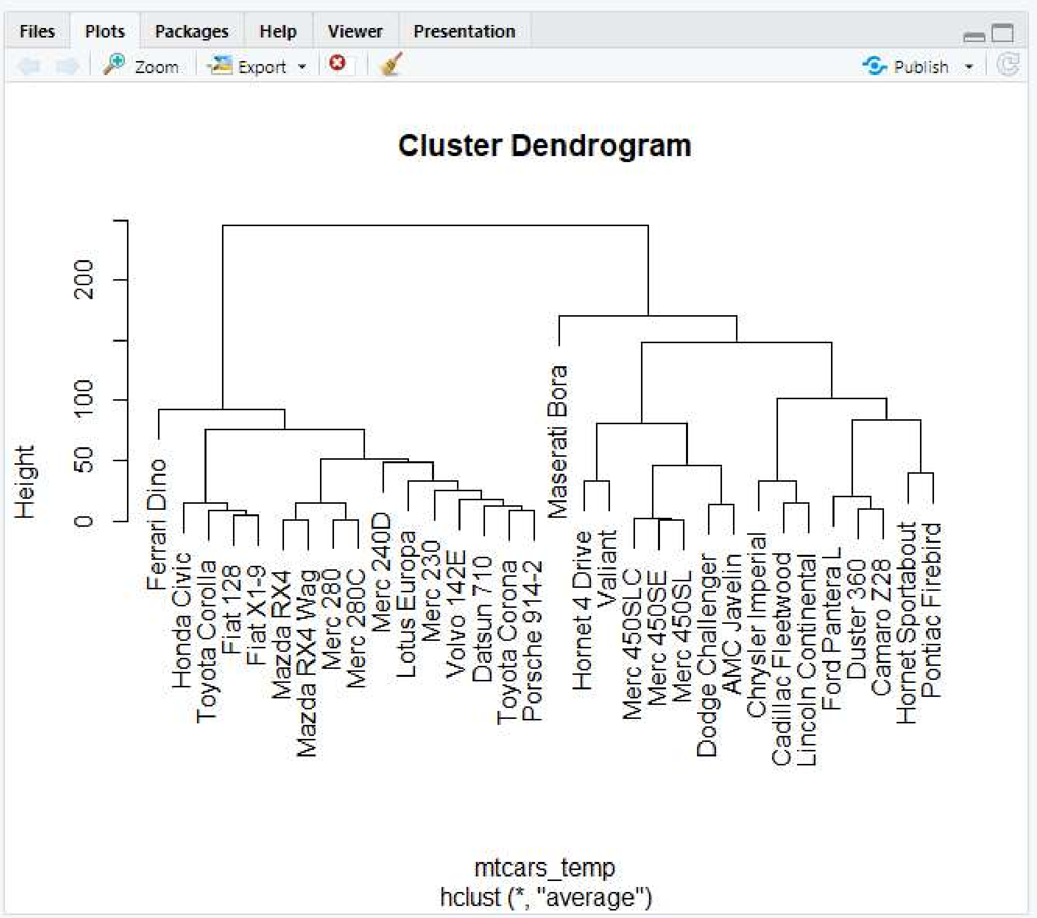
Output:

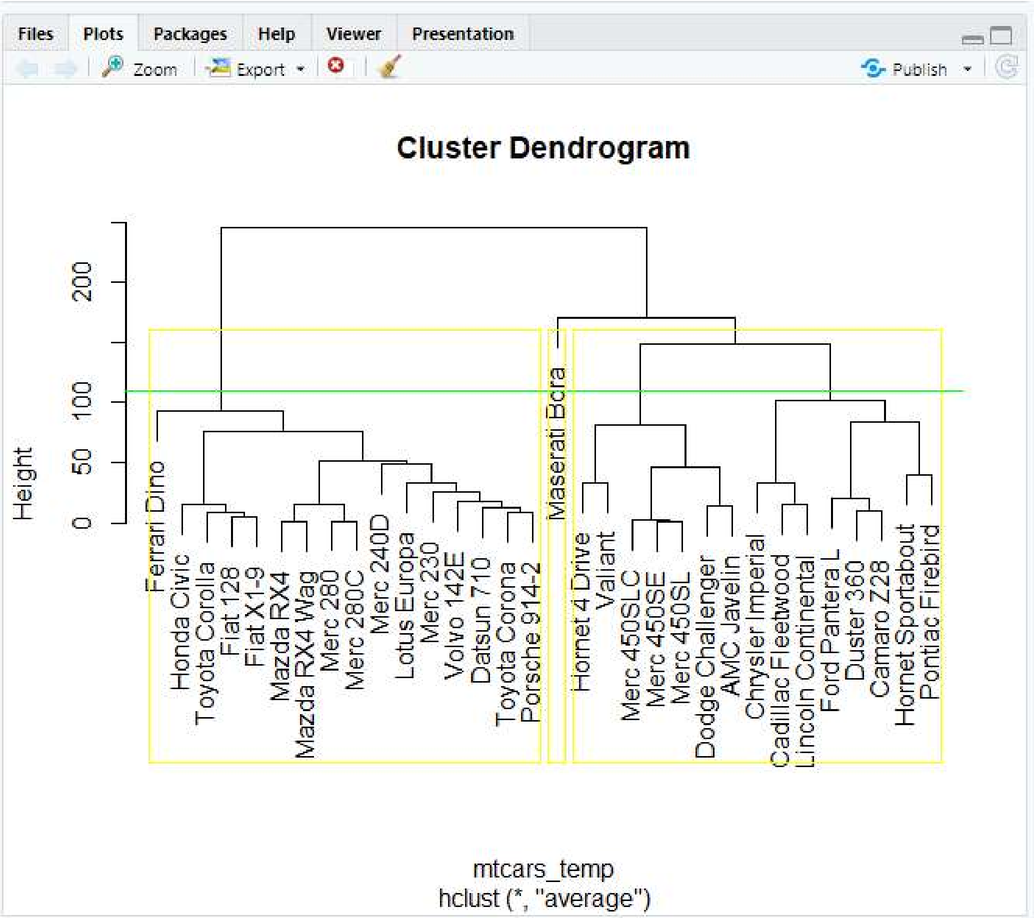
```
Console   Terminal ×   Background Jobs ×
R 4.2.1 · ~/
> mtcars_temp <- dist(mtcars, method = 'euclidean')
> mtcars_temp
                      Mazda RX4 Mazda RX4 Wag  Datsun 710 Hornet 4 Drive Hornet Sportabout     Valiant  Duster 360   Merc 240D    Merc 230
Mazda RX4 Wag         0.6153251
Datsun 710           54.9086059    54.8915169
Hornet 4 Drive       98.1125212    98.0958939 150.9935191
Hornet Sportabout   210.3374396   210.3358546 265.0831615    121.0297564
Valiant              65.4717710    65.4392224 117.7547018     33.5508692       152.1241352
Duster 360          241.4076490   241.4088680 294.4790230    169.4299647        70.1767262 194.6094525
Merc 240D            50.1532711    50.1146059  49.6584796    121.2739722       241.5069657  89.5911056 281.2962502
Merc 230             25.4683117    25.3284509  33.1803843    118.2433145       233.4924012  85.0079649 265.8823313  33.6873047
Merc 280             15.3641921    15.2956865  66.9363534     91.4224033       199.3344960  60.2909811 227.8998521  64.7754228  39.2994160
Merc 280C            15.6724727    15.5837744  67.0261397     91.4612914       199.3406564  60.2655656 227.8813169  64.8898713  39.3868519
Merc 450SE          135.4307018   135.4254826 189.1954941     72.4964325        84.3888482  90.6970264 106.4084264 175.1620073 159.8179555
Merc 450SL          135.4014424   135.3960351 189.1631745     72.4313532        84.3683999  90.6769728 106.4320572 175.1189767 159.7760899
Merc 450SLC         135.4794674   135.4723157 189.2345426     72.5718466        84.4332423  90.7092989 106.4010305 175.2118218 159.8495837
Cadillac Fleetwood  326.3395903   326.3355070 381.0926242    234.4403876       116.2804201 266.6280942 119.0239068 355.6627498 349.2832611
Lincoln Continental 318.0469808   318.0429333 372.8012090    227.9726091       108.0624299 259.6304391 104.5112999 348.9901277 341.3154316
Chrysler Imperial   304.7203408   304.7169175 359.3014906    218.1548299        97.2049146 248.7713290  81.4297699 338.1959373 328.4335161
Fiat 128             93.2679950    93.2530993  40.9933763    184.9689734       302.0377212 152.1153263 333.9792070  68.6105903  69.3127910
Honda Civic         102.8307567   102.8238713  52.7704607    191.5518700       310.0324645 158.9615769 344.0518316  72.0014488  78.5387212
Toyota Corolla      100.6040368   100.5887588  47.6535017    192.6714187       309.5581776 159.8302995 341.0218232  76.2806458  76.7731674
Toyota Corona        42.3075233    42.2659224  12.9654743    138.5304725       252.3331988 105.2876428 282.0508820  44.0850975  21.0962017
Dodge Challenger    163.1150750   163.1134210 217.7795805     72.4403915        48.9838851 103.4310693 103.9023864 192.8617917 185.8331870
AMC Javelin         149.6047203   149.6014522 204.3188913     61.3601899        61.4274240  91.0444349 110.3084921 180.5479760 172.5312555
Camaro Z28          233.2228758   233.2248748 286.0049209    163.6632641        70.9665308 187.8463771  10.0761203 273.8367985 257.7469734
Pontiac Firebird    248.6780270   248.6762035 303.3583889    156.2240346        40.0052475 188.5272116  80.8057339 277.4606884 271.3871978
Fiat X1-9            92.5048389    92.4940020  39.8815148    184.4471198       301.5669483 151.4379425 333.4843231  67.9163981  68.5564864
Porsche 914-2        44.4033659    44.4073589  13.1357109    139.1579524       254.1452553 106.0585767 285.1986201  39.4469276  22.1180967
Lotus Europa         65.7328377    65.7362635  25.0948550    163.2367437       272.3582423 130.8248192 296.4572287  72.8971106  50.1094030
Ford Pantera L      245.4247064   245.4293785 297.2940489    180.1140339        89.5934049 203.0177926  21.2655990 287.5238795 269.9772035
Ferrari Dino         66.7661029    66.7764167  90.2415509    130.5523007       215.0673853 106.5694802 226.2036333 113.3023005  80.6550953
Maserati Bora       265.6454248   265.6491465 309.7718171    229.3419352       170.7094473 242.4393015 107.7224977 313.8633093 288.8755628
Volvo 142E           39.1894029    39.1626037  20.6939436    137.0363299       248.0063378 104.1863681 275.1353516  53.6823481  24.6913548
                       Merc 280   Merc 280C  Merc 450SE  Merc 450SL Merc 450SLC Cadillac Fleetwood Lincoln Continental Chrysler Imperial
Mazda RX4 Wag
Datsun 710
Hornet 4 Drive
Hornet Sportabout
Valiant
Duster 360
Merc 240D
Merc 230
Merc 280
Merc 280C             1.5231546
Merc 450SE          122.3642489 122.3461050
Merc 450SL          122.3443771 122.3355492   0.9826495
Merc 450SLC         122.3934970 122.3586862   1.3726252   2.1383405
Cadillac Fleetwood  315.3904859 315.3557081 197.8842803 197.9154476 197.8526242
Lincoln Continental 306.6760719 306.6406187 187.5997191 187.6330806 187.5671081         15.6224446
Chrysler Imperial   292.7146896 292.6989332 171.6600758 171.6743028 171.6557637         40.8399636          25.3714237
Fiat 128            106.5053149 106.6829794 228.3247948 228.2592340 228.4051825        417.7687579         410.0206984       397.2276375
Honda Civic         116.7280991 116.8711475 238.0141824 237.9588183 238.0828999        425.3271621         417.9679574       405.8152201
Toyota Corolla      113.6290721 113.8118009 235.5183809 235.4481971 235.6024098        425.3446517         417.5429986       404.6335386
Toyota Corona        54.3641713  54.4258314 176.6020527 176.5727477 176.6305359        368.3195488         360.0267515       346.5724649
Dodge Challenger    152.8929263 152.8722437  51.8008639  51.8242520  51.8012606        163.6314881         156.2805020       145.9194779
```

Console Terminal Background Jobs

R 4.2.1 · ~/

```
Merc 280C
Merc 450SE
Merc 450SL
Merc 450SLC
Cadillac Fleetwood
Lincoln Continental
Chrysler Imperial
Fiat 128
Honda Civic
Toyota Corolla
Toyota Corona
Dodge Challenger
AMC Javelin
Camaro Z28
Pontiac Firebird
Fiat X1-9
Porsche 914-2
Lotus Europa          33.7678653
Ford Pantera L       288.5852993  297.5376920
Ferrari Dino          87.9105966   80.4553451   224.4587490
Maserati Bora        303.9222549  303.2796468    86.9383253  223.5342175
Volvo 142E            18.7555858   27.8104457   277.4803312   70.4751034   289.1157363
> set.seed(240)
> hierarchial_cluster_model <- hclust(mtcars_temp, method = "average")
> hierarchial_cluster_model

Call:
hclust(d = mtcars_temp, method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 32

> # Plotting dendrogram
> plot(hierarchial_cluster_model)
> # Cutting tree by height
> abline(h = 110, col = "green")
> # Cutting by no. of clusters
> final_fit <- cutree(hierarchial_cluster_model, k = 3 )
> final_fit
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive   Hornet Sportabout             Valiant          Duster 360
                  1                   1                   1                   2                   2                   2                   2
         Merc 240D            Merc 230            Merc 280           Merc 280C          Merc 450SE          Merc 450SL         Merc 450SLC
                  1                   1                   1                   1                   2                   2                   2
 Cadillac Fleetwood Lincoln Continental   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla       Toyota Corona
                  2                   2                   2                   1                   1                   1                   1
   Dodge Challenger         AMC Javelin          Camaro Z28    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
                  2                   2                   2                   2                   1                   1                   1
     Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
                  2                   1                   3                   1
> table(final_fit)
final_fit
 1  2  3 
16 15  1 
> rect.hclust(hierarchial_cluster_model, k = 3, border = "yellow")
> 
```


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Cluster Dendrogram
Height
0
50
100
200
Ferrari Dino
Honda Civic
Toyota Corolla
Fiat 128
Fiat X1-9
Mazda RX4
Mazda RX4 Wag
Merc 280
Merc 280C
Merc 240D
Lotus Europa
Merc 230
Volvo 142E
Datsun 710
Toyota Corona
Porsche 914-2
Maserati Bora
Hornet 4 Drive
Valiant
Merc 450SLC
Merc 450SE
Merc 450SL
Dodge Challenger
AMC Javelin
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Ford Pantera L
Duster 360
Camaro Z28
Hornet Sportabout
Pontiac Firebird
mtcars_temp
hclust (*, "average")


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Cluster Dendrogram
Height
0
50
100
200
Ferrari Dino
Honda Civic
Toyota Corolla
Fiat 128
Fiat X1-9
Mazda RX4
Mazda RX4 Wag
Merc 280
Merc 280C
Merc 240D
Lotus Europa
Merc 230
Volvo 142E
Datsun 710
Toyota Corona
Porsche 914-2
Maserati Bora
Hornet 4 Drive
Valiant
Merc 450SLC
Merc 450SE
Merc 450SL
Dodge Challenger
AMC Javelin
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Ford Pantera L
Duster 360
Camaro Z28
Hornet Sportabout
Pontiac Firebird
mtcars_temp
hclust (*, "average")