Assesment 2

Ujjwal Gautam

2024-05-31

QN 9

Question a

```r
city_distances <- matrix(c(
  0, 587, 1212, 701, 1936, 604, 748, 2139, 2182, 543,
  587, 0, 920, 940, 1745, 1188, 713, 1858, 1737, 597,
  1212, 920, 0, 879, 831, 1726, 1611, 1949, 2204, 1494,
  701, 940, 879, 0, 1374, 968, 1420, 1645, 1891, 1220,
  1936, 1745, 831, 1374, 0, 2339, 2451, 347, 2734, 2300,
  604, 1188, 1726, 968, 2339, 0, 1092, 2594, 2408, 923,
  748, 713, 1611, 1420, 2451, 1092, 0, 2571, 678, 205,
  2139, 1858, 1949, 1645, 347, 2594, 2571, 0, 678, 2442,
  2182, 1737, 2204, 1891, 2734, 2408, 678, 678, 0, 2329,
  543, 597, 1494, 1220, 2300, 923, 205, 2442, 2329, 0
), nrow = 10, byrow = TRUE)

city_names <- c("Atlanta", "Chicago", "Denver", "Houston", "Los Angeles", "Miami", "New York", "San Fra
rownames(city_distances) <- city_names
colnames(city_distances) <- city_names

city_dissimilarity <- as.dist(city_distances)
```

- Create a distance matrix from the given data in the problem
- Assigning names to row and columns
- Convert to a dissimilarity object

Question b

```r
mds_model <- cmdscale(city_dissimilarity, eig = TRUE, k = 2)
```

- Fitting the classical MDS model using city.dissimilarity object

Question c

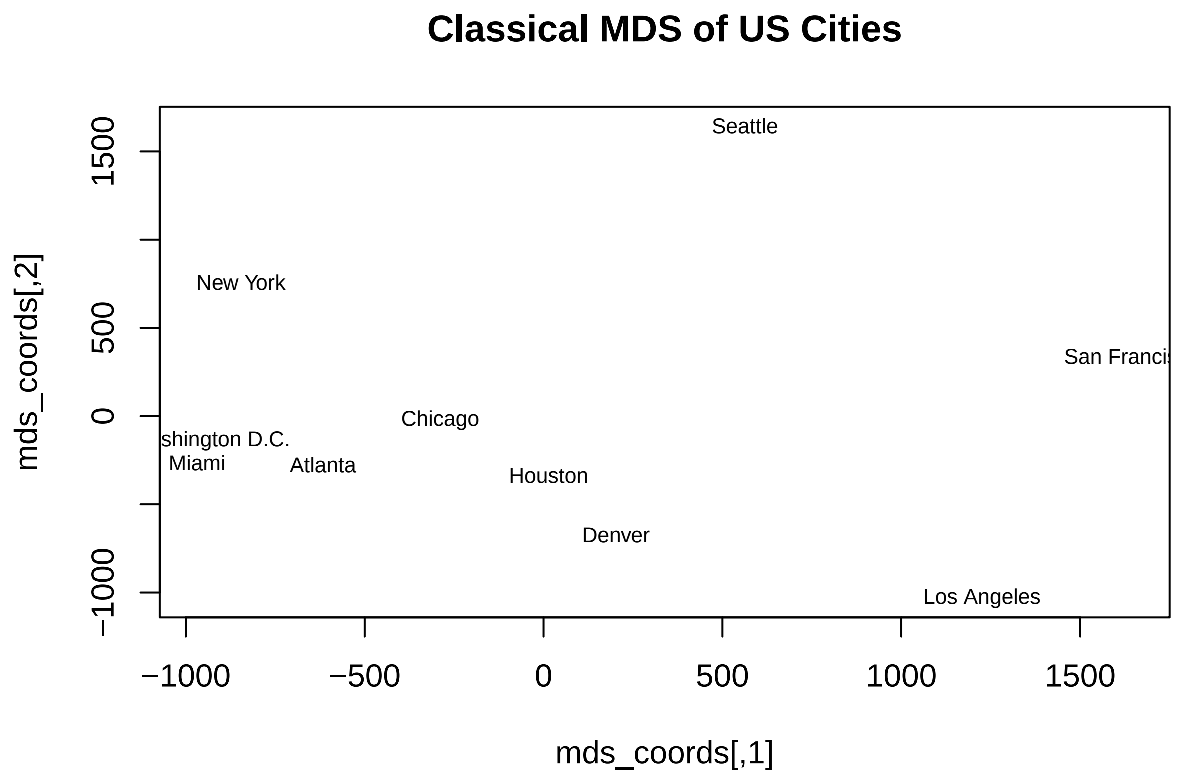
```r
mds_coords <- mds_model$points
print(mds_coords)
```

```
##                        [,1]        [,2]
## Atlanta         -616.46326  -277.03319
## Chicago         -288.61063   -22.16151
## Denver           202.61148  -672.61019
## Houston           14.25242  -335.54496
## Los Angeles     1225.78174 -1033.78934
## Miami           -968.45797  -264.31832
## New York        -845.50822   757.66327
## San Francisco   1645.58380   339.92746
## Seattle          563.12009  1646.43854
## Washington D.C. -932.30945  -138.57175
```

- Summarizing the model
- the above points gives the coordinate for each city in two dimension
- the points are in such a way that it preserves the distance between cities

Question d

```r
plot(mds_coords, type = "n")
text(mds_coords, labels = city_names, cex = 0.7)
title("Classical MDS of US Cities")
```



- Bi-plot of the model
- the plot shows the cities with given coordinate
- the coordinates are replaced with cities name , the city names are then added to the plot using the text() function, with the coordinates from the MDS analysis.
- First of all, a distance matrix city_distances is created from the provided data. Each cell represents the distance between two cities.
- Classical MDS is performed on the dissimilarity matrix using the cmdscale() function.
- The resulting coordinates of the cities in the MDS space are extracted from the model.
- These coordinates are then printed to the console.
- A bi-plot of the MDS model is then created using the plot() function.
- The type = "n" argument ensures that only the plot framework is created, with no points plotted initially.
- The city names are then added to the plot using the text() function, with the coordinates from the MDS analysis.

QN 10

```r
library(ClusterR)
library(cluster)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
table(iris$Species)
```

```
## 
##     setosa versicolor  virginica 
##         50         50         50
```

```r
iris_1<-iris[,-5]
sum(is.na(iris_1))
```

```
## [1] 0
```

```r
set.seed(34)
```

- observing the data set

a

```r
kmeans.c2<-kmeans(iris_1,centers = 2,nstart = 20)
kmeans.c2
```

```
## K-means clustering with 2 clusters of sizes 97, 53
## 
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.301031    2.886598     4.958763    1.695876
## 2     5.005660    3.369811     1.560377    0.290566
## 
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 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
## [149] 1 1
## 
## Within cluster sum of squares by cluster:
## [1] 123.79588  28.55208
##  (between_SS / total_SS =  77.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- k mean clustering with two clusters
- it forms two clusters of sizes 53, 97

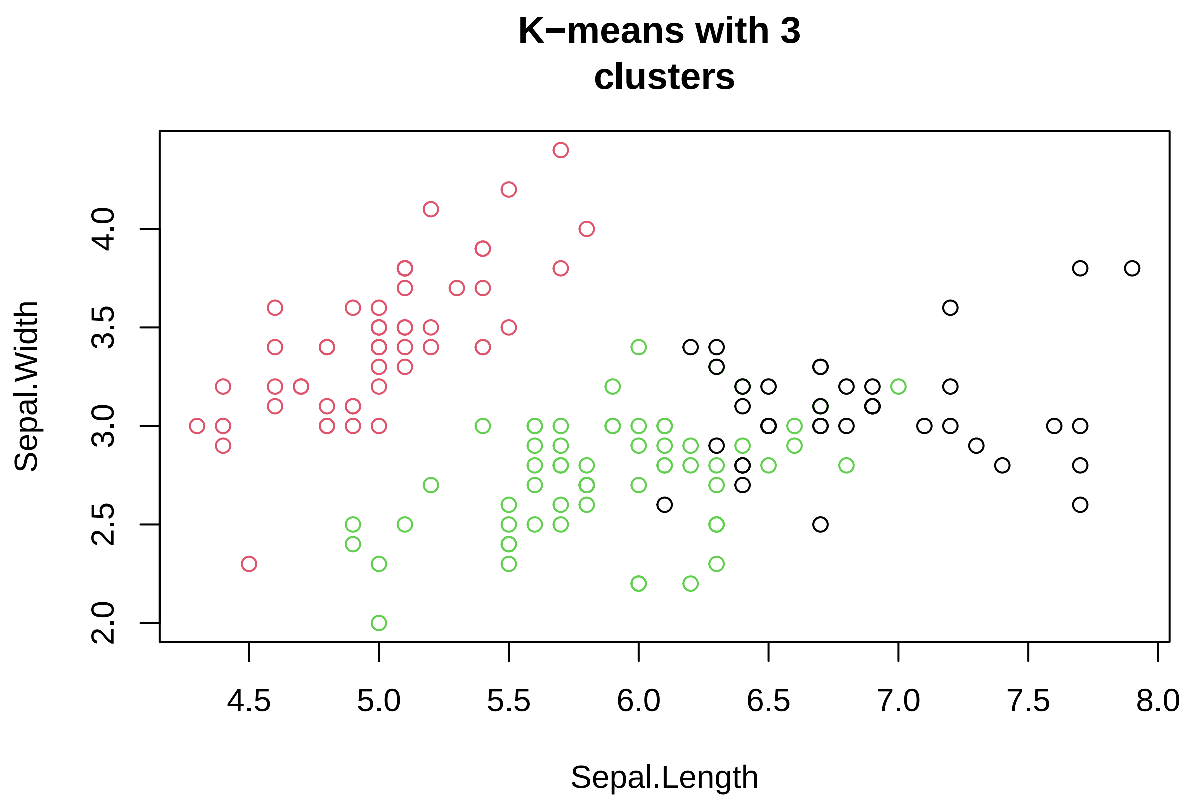
```
kmeans.c3<-kmeans(iris_1,centers = 3,nstart = 20)
kmeans.c3
```

```
## K-means clustering with 3 clusters of sizes 38, 50, 62
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.850000    3.073684     5.742105    2.071053
## 2     5.006000    3.428000     1.462000    0.246000
## 3     5.901613    2.748387     4.393548    1.433871
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1
## [112] 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 1 3 1
## [149] 1 3
##
## Within cluster sum of squares by cluster:
## [1] 23.87947 15.15100 39.82097
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- k mean clustering with three clusters
- it forms two clusters of sizes 50, 62, 38

b

```
plot(iris_1[c("Sepal.Length",
              "Sepal.Width")],
     col = kmeans.c3$cluster,
     main = "K-means with 3
clusters")
```



- the plot for three clusters is formed
- each cluster is colored differently ie. black, red and green
- in plot some of the green and red are seen mixed which means points are not well clustered
- for better visualization sepal length and sepal width are considered for plotting

c

```
kmeans.c3$centers
```

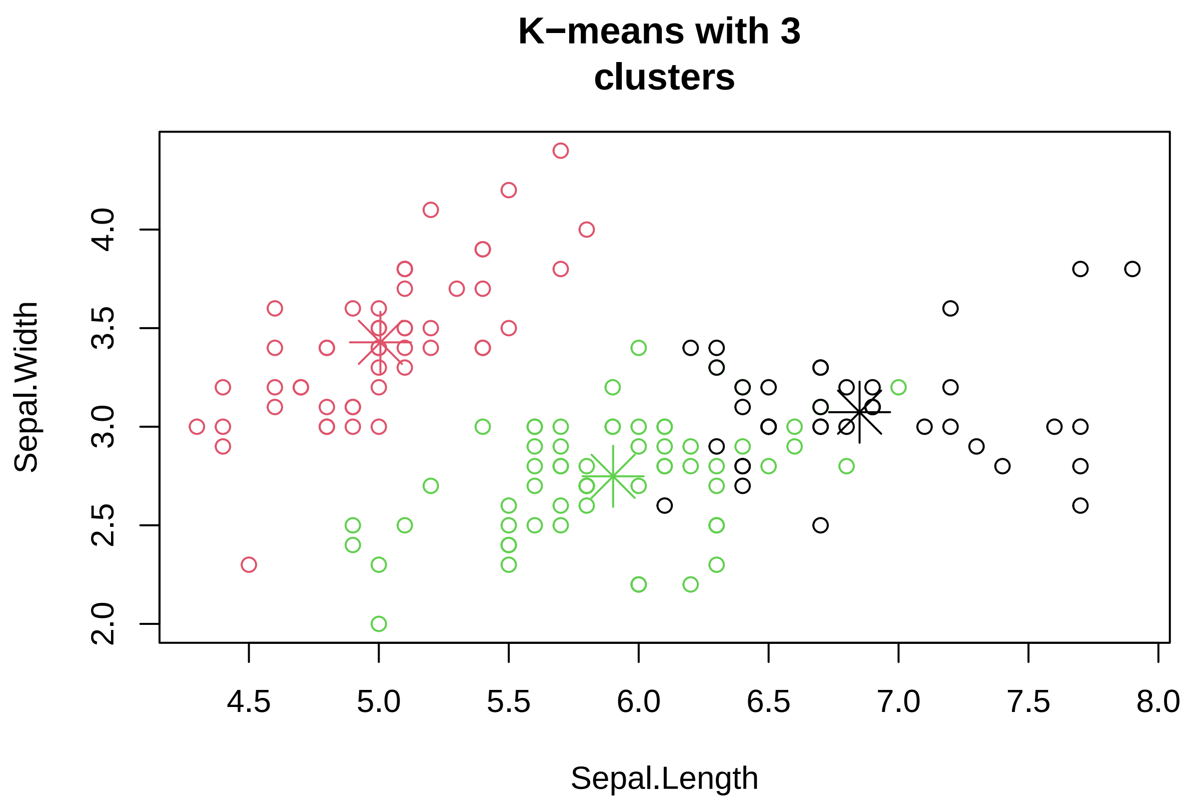
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.850000    3.073684     5.742105    2.071053
## 2     5.006000    3.428000     1.462000    0.246000
## 3     5.901613    2.748387     4.393548    1.433871
```

```
kmeans.c3$centers[,
                  c("Sepal.Length", "Sepal.Width")]
```

```
##   Sepal.Length Sepal.Width
## 1     6.850000    3.073684
## 2     5.006000    3.428000
## 3     5.901613    2.748387
```

```
plot(iris_1[c("Sepal.Length",
              "Sepal.Width")],
     col = kmeans.c3$cluster,
     main = "K-means with 3
clusters")

points(kmeans.c3$centers[,
                          c("Sepal.Length", "Sepal.Width")],col = 1:3, pch= 8, cex= 3)
```



- the center of clusters with respect to sepal length and sepal width is found
- the center is plotted with plot () function
- the center gives mean of each cluster

d

```
 cm<-table(iris$Species,kmeans.c3$cluster)
cm
```

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

```r
(accuracy<-
    sum(diag(cm))/sum(cm))
```

```
## [1] 0.09333333
```

- three clusters were compared using confusion matrix
- the cm gives that 50 of setosa , 48 of versicolor and 36 of virginica were clustered correctly
- 14 of virginica and 2 of versicolr are predicted incorrectly
- using confusion matrix accuracy was found to be 89.33 %

QN 8

```r
library(car)
```

```
## Loading required package: carData
```

```r
head(Arrests)
```

```
##   released colour year age    sex employed citizen checks
## 1      Yes  White 2002  21   Male      Yes     Yes      3
## 2       No  Black 1999  17   Male      Yes     Yes      3
## 3      Yes  White 2000  24   Male      Yes     Yes      3
## 4       No  Black 2000  46   Male      Yes     Yes      1
## 5      Yes  Black 1999  27 Female      Yes     Yes      1
## 6      Yes  Black 1998  16 Female      Yes     Yes      0
```

```r
str(Arrests)
```

```
## 'data.frame':    5226 obs. of  8 variables:
##  $ released: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 2 2 2 ...
##  $ colour  : Factor w/ 2 levels "Black","White": 2 1 2 1 1 1 2 2 1 2 ...
##  $ year    : int  2002 1999 2000 2000 1999 1998 1999 1998 2000 2001 ...
##  $ age     : int  21 17 24 46 27 16 40 34 23 30 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 1 2 2 ...
##  $ employed: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 1 2 2 2 ...
##  $ citizen : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ checks  : int  3 3 3 1 1 0 0 1 4 3 ...
```

```r
dim(Arrests)
```

```
## [1] 5226    8
```

```r
data<-Arrests
data<-data[,-3]
data<-data[,-7]
head(data)
```

```
##   released colour age    sex employed citizen
## 1      Yes  White  21   Male      Yes     Yes
## 2       No  Black  17   Male      Yes     Yes
## 3      Yes  White  24   Male      Yes     Yes
## 4       No  Black  46   Male      Yes     Yes
## 5      Yes  Black  27 Female      Yes     Yes
## 6      Yes  Black  16 Female      Yes     Yes
```

- removing 3rd and 4 th column from data

a

```
ind <- sample(2, nrow(data),
              replace=T, prob = c(0.8, 0.2))
train <- data[ind==1,]
test <- data[ind==2,]
```

- splitting data for train and test in ratio 80 nad 20

b

```
library(e1071)
set.seed(34)
model_lr<-glm(released~.,data=train,family = binomial)
```

- fitting logistic binomial regression using glm() function

```
model.nb<-naiveBayes(released~.,data=train)
```

- fitting naive bayes using naiveBayes() function

c

```
predict<-predict(model_lr,newdata=test,type="response")
summary(predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4734  0.8035  0.8787  0.8294  0.8898  0.9154
```

```
head(predict)
```

```
##        10        15        16        18        28        32
## 0.8822581 0.6305144 0.7471007 0.8891582 0.8230127 0.8210177
```

- prediting test data on model
- predict is continuous but we need in 0 and 1

```r
predict_lr<-as.numeric((ifelse(predict>0.5,1,0)))
summary(predict_lr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  1.0000  1.0000  0.9915  1.0000  1.0000
```

```r
head(predict_lr)
```

```
## [1] 1 1 1 1 1 1
```

- prediction to binary variable

```r
pred_naive<-predict(model.nb,newdata=test)
head(pred_naive)
```

```
## [1] Yes Yes Yes Yes Yes Yes
## Levels: No Yes
```

- predicting 20% of test data using naive bayes model

d

```r
cm_lr<-table(test$released,predict_lr)
cm_lr
```

```
##      predict_lr
##         0   1
##   No    2 167
##   Yes   7 878
```

```r
(accuracy_lr<-sum(diag(cm_lr)/sum(cm_lr)))
```

```
## [1] 0.8349146
```

- finding confusion matrix of predicted values using logistic regression
- finding accuracy of prediction
- accuracy was found to be 83.49%

```r
cm_nv<-table(test$released,pred_naive)
cm_nv
```

```
##      pred_naive
##        No Yes
##   No   12 157
##   Yes  25 860
```

```
(accuracy_nv<-sum(diag(cm_nv)/sum(cm_nv)))
```

```
## [1] 0.8273245
```

- finding confusion matrix of predicted values using naive bayes
- finding accuracy of prediction
- accuracy was found to be 82.72%
- In conclusion logistic regression model is better for this model as it gives higher accuracy of 83.49%.