Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(410908) Data Science Laboratory

****

Class: SY-MCA Shift / Div: A Roll Number: 52040

Name: Tanima Mandal Assignment No: 3 Date of Implementation: 04.10.2023

****

Q1.We have four things: grape, green bean, nuts and orange with two characteristics: sweetness (8, 3, 3, 7) and Crunchiness (5, 7, 6, 3). Among them two are fruits, one is protein and one is vegetable. Suppose we wanted to classify tomatoes into one of the classes. Is tomato a fruit, vegetable or protein? Tomato has the following characteristics; sweetness=6, crunchiness = 4. Let's add Carrots with characteristics sweetness = 4 and crunchiness = 9 keep k=1. Try for k=4 also.

Program:

```
things <- data.frame(ingredient =  c("grape", "green bean" , "nuts" , "orange"),

                     sweetness = c(8,3,3,7),

                     crunchiness = c(5,7,6,3),

                     class = c("fruit", "vegetable", "protein", "fruit"))

things



unknown <- data.frame(ingredient =  "tomato",

                      sweetness = 6,

                      crunchiness = 4,

                      class="unknown")

unknown



#install.packages("dplyr")
```

```
#install.packages("descr")

#install.packages("ggplot2")


library(dplyr)

library(descr)

library(ggplot2)

ggplot(bind_rows(things, unknown)) +

  geom_point(aes(x=sweetness, y=crunchiness, color=class),size=10) +

  geom_label(aes(x=sweetness, y=crunchiness, label=ingredient), hjust = 0, nudge_x = 0.25)+

  xlim(2,9) + ylim(3,8)


library(class) #contains knn function

pred <- knn(select(things, sweetness, crunchiness),

          select(unknown,sweetness, crunchiness), things$class, k=1)

pred


unknown <- data.frame(ingredient =  c("tomato", "carrot"),

                      sweetness = c(6,4),

                      crunchiness = c(4,9),

                      class=c("unknown", "unknown"))

unknown


pred <- knn(select(things, sweetness, crunchiness),

          select(unknown,sweetness, crunchiness), things$class, k=1)

pred
```

```
pred <- knn(select(things, sweetness, crunchiness),
            select(unknown,sweetness, crunchiness), things$class, k=4)
pred
```

Output:

```
> things <- data.frame(ingredient =  c("grape", "green bean" , "nuts" , "orange"),
+                      sweetness = c(8,3,3,7),
+                      crunchiness = c(5,7,6,3),
+                      class = c("fruit", "vegetable", "protein", "fruit"))
> things
  ingredient sweetness crunchiness     class
1      grape         8           5     fruit
2 green bean         3           7 vegetable
3       nuts         3           6   protein
4     orange         7           3     fruit
>
> unknown <- data.frame(ingredient =  "tomato",
+                       sweetness = 6,
+                       crunchiness = 4,
+                       class="unknown")
> unknown
  ingredient sweetness crunchiness   class
1     tomato         6           4 unknown
>
> #install.packages("dplyr")
> #install.packages("descr")
> #install.packages("ggplot2")
>
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

> library(descr)
> library(ggplot2)
> ggplot(bind_rows(things, unknown)) +
+   geom_point(aes(x=sweetness, y=crunchiness, color=class),size=10) +
+   geom_label(aes(x=sweetness, y=crunchiness, label=ingredient), hjust = 0, nudge_x = 0.25)+
+   xlim(2,9) + ylim(3,8)
```

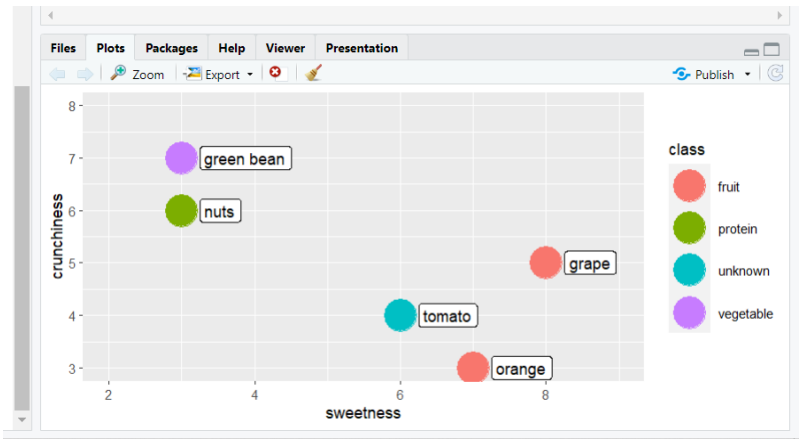
```
> ggplot(bind_rows(things, unknown)) +
+   geom_point(aes(x=sweetness, y=crunchiness, color=class),size=10) +
+   geom_label(aes(x=sweetness, y=crunchiness, label=ingredient), hjust = 0, nudge_x = 0.25)+
+   xlim(2,9) + ylim(3,8)
>
> library(class) #contains knn function
> pred <- knn(select(things, sweetness, crunchiness),
+             select(unknown,sweetness, crunchiness), things$class, k=1)
> pred
[1] fruit
Levels: fruit protein vegetable
>
> unknown <- data.frame(ingredient =  c("tomato", "carrot"),
+                       sweetness = c(6,4),
+                       crunchiness = c(4,9),
+                       class=c("unknown", "unknown"))
> unknown
  ingredient sweetness crunchiness   class
1     tomato         6           4 unknown
2     carrot         4           9 unknown
>
> pred <- knn(select(things, sweetness, crunchiness),
+             select(unknown,sweetness, crunchiness), things$class, k=1)
> pred
[1] fruit     vegetable
Levels: fruit protein vegetable
>
> pred <- knn(select(things, sweetness, crunchiness),
+             select(unknown,sweetness, crunchiness), things$class, k=4)
> pred
[1] fruit fruit
Levels: fruit protein vegetable
```

For k=1

```
>
> pred <- knn(select(things, sweetness, crunchiness),
+             select(unknown,sweetness, crunchiness), things$class, k=1)
> pred
[1] fruit     vegetable
Levels: fruit protein vegetable
>
```

For k=4

```
>
> pred <- knn(select(things, sweetness, crunchiness),
+             select(unknown,sweetness, crunchiness), things$class, k=4)
> pred
[1] fruit fruit
Levels: fruit protein vegetable
>
```



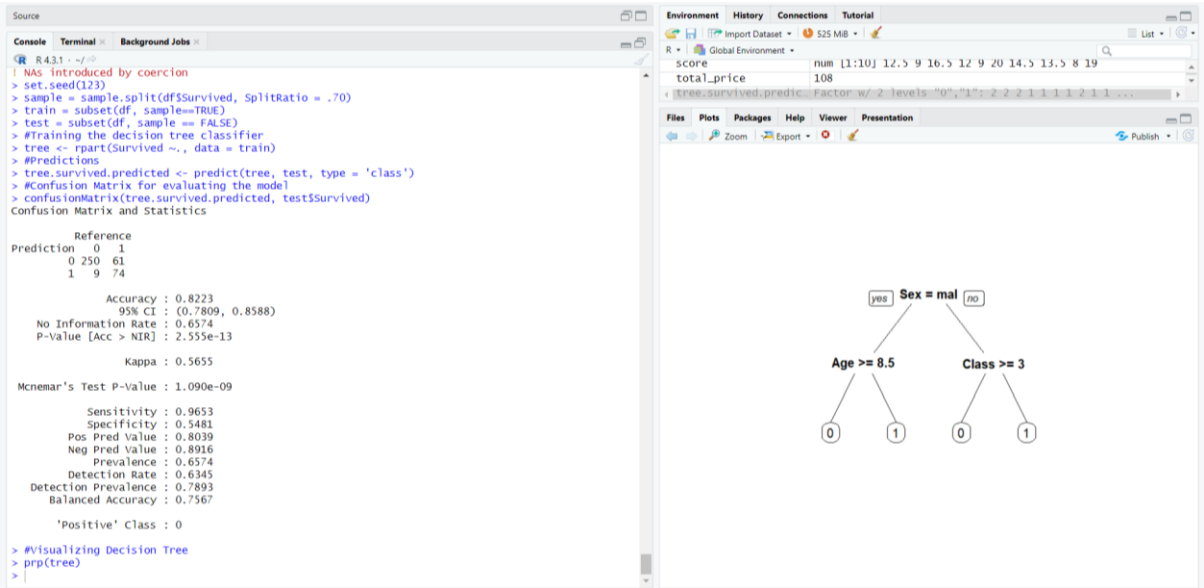
Q2.Using Titanic.CSV file predict which people are more likely to survive after the collision with the iceberg using Decision Trees.

Program:

```
library(caret)

library(FSelector)

library(rpart)

library(rpart.plot)

library(dplyr)

library(xlsx)

library(data.tree)

library(caTools)



df <- read.xlsx("C:\\Users\\Home\\Desktop\\Data_Science\\Titanic_Assign_3_2.xlsx",sheetIndex = 1
)

df



#Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jdk1.8.0_202')

summary(df)

#Titanic <- Titanic[,c('Class','Age','Sex','Survived')]

df <- select(df, Survived, Class, Sex, Age)

df <- na.omit(df)

df

df <- mutate(df, Survived = factor(Survived), Class = as.numeric(Class), Age =

        as.numeric(Age))

set.seed(123)
```

```
sample = sample.split(df$Survived, SplitRatio = .70)

train = subset(df, sample==TRUE)

test = subset(df, sample == FALSE)

#Training the decision tree classifier

tree <- rpart(Survived ~., data = train)

#Predictions

tree.survived.predicted <- predict(tree, test, type = 'class')

#Confusion Matrix for evaluating the model

confusionMatrix(tree.survived.predicted, test$Survived)

#Visualizing Decision Tree

prp(tree)
```

Output:



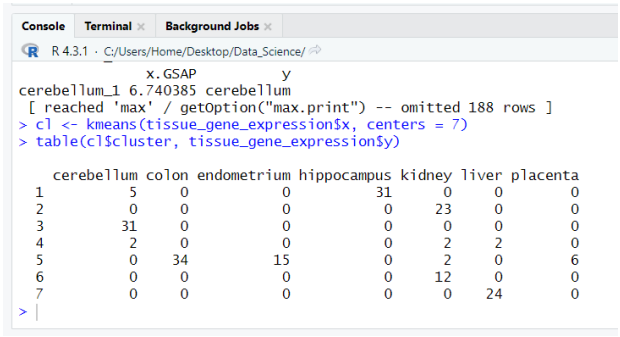
Q3.Load the tissue gene expression dataset. Run a k-means clustering on the data with K=7. Make a table comparing the identified clusters to the actual tissue types. Run the algorithm several times to see how the answer changes.

Program:

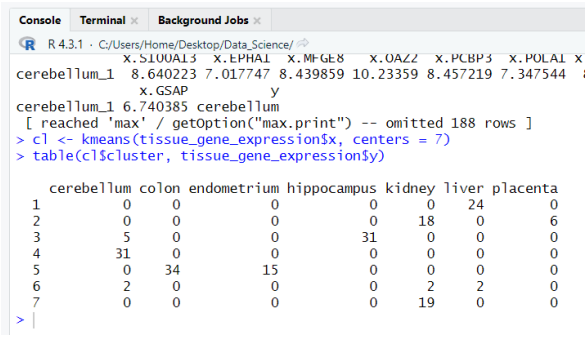
```
#install.packages("dslabs")

library(dslabs)

data("tissue_gene_expression")

df <- data.frame(tissue_gene_expression)

df

cl <- kmeans(tissue_gene_expression$x, centers = 7)

table(cl$cluster, tissue_gene_expression$y)
```

Output:

1st run



2nd run



Q4.Plot the distribution of distances between data points and their fifth nearest neighbors using the KNNdistplot function from the dbscan package. Examine the plot and find a tentative threshold at which distances start increasing quickly. On the same plot, draw a horizontal line at the level of the threshold (use Iris dataset)

Program:

```
#install.packages("dbscan")

df <- iris[, -ncol(iris)]


df <- scale(df)

df <- as.data.frame(df)


library(dbscan)

kNNdistplot(df, k = 5)

abline(h = 0.8, col = "red")
```

Output: