Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

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 tomato into one of the classes. Is tomato a fruit, vegetable or protein? Tomato has the following characteristics: sweetness = 6, crunchiness = 4. Let's add Carrot with characteristics sweetness = 4 and crunchiness = 9 keep k=1. Try for k=4 also.

1) K=1

Program :

```
existing_items <- data.frame(
  Sweetness = c(8, 3, 3, 7),
  Crunchiness = c(5, 7, 6, 3)
)

labels <- c(0, 1, 2, 0)
library(class)
k <- 1

item_to_classify1 <- data.frame(Sweetness = 6,Crunchiness = 4)

item_to_classify2 <- data.frame(Sweetness = 4,Crunchiness = 9)

predicted_class1 <- knn(existing_items, item_to_classify1, labels, k)
class_labels <- c("Fruit", "Vegetable", "Protein")
predicted_label1 <- class_labels[predicted_class1]

predicted_class2 <- knn(existing_items, item_to_classify2, labels, k)
class_labels <- c("Fruit", "Vegetable", "Protein")
predicted_label2 <- class_labels[predicted_class2]

cat("The item (tomato) is classified as:", predicted_label1, "\n")
cat("The item (carrot) is classified as:", predicted_label2, "\n")
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA   Shift / Div : S3/B   Roll Number : 52147

Name : Nisha Harish Parekh   Assignment No : 3   Date of Implementation : 16/10/2023

****

Output :

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> existing_items <- data.frame(
+   Sweetness = c(8, 3, 3, 7),
+   Crunchiness = c(5, 7, 6, 3)
+ )
>
> labels <- c(0, 1, 2, 0)
> library(class)
> k <- 1
>
> item_to_classify1 <- data.frame(Sweetness = 6,Crunchiness = 4)
>
> item_to_classify2 <- data.frame(Sweetness = 4,Crunchiness = 9)
>
> predicted_class1 <- knn(existing_items, item_to_classify1, labels, k)
> class_labels <- c("Fruit", "Vegetable", "Protein")
> predicted_label1 <- class_labels[predicted_class1]
>
> predicted_class2 <- knn(existing_items, item_to_classify2, labels, k)
> class_labels <- c("Fruit", "Vegetable", "Protein")
> predicted_label2 <- class_labels[predicted_class2]
>
> cat("The item (tomato) is classified as:", predicted_label1, "\n")
The item (tomato) is classified as: Fruit
> cat("The item (carrot) is classified as:", predicted_label2, "\n")
The item (carrot) is classified as: Vegetable
> 
```

****
Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023
****

2) K=4

Program :

```
existing_items <- data.frame(
  Sweetness = c(8, 3, 3, 7),
  Crunchiness = c(5, 7, 6, 3)
)

labels <- c(0, 1, 2, 0)
library(class)
k <- 4

item_to_classify1 <- data.frame(Sweetness = 6,Crunchiness = 4)

item_to_classify2 <- data.frame(Sweetness = 4,Crunchiness = 9)

class_labels <- c("Fruit", "Vegetable", "Protein")

predicted_class1 <- knn(existing_items, item_to_classify1, labels, k)
predicted_label1 <- class_labels[predicted_class1]

predicted_class2 <- knn(existing_items, item_to_classify2, labels, k)
predicted_label2 <- class_labels[predicted_class2]

cat("The item (tomato) is classified as:", predicted_label1, "\n")
cat("The item (carrot) is classified as:", predicted_label2, "\n")
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Output :

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> existing_items <- data.frame(
+   Sweetness = c(8, 3, 3, 7),
+   Crunchiness = c(5, 7, 6, 3)
+ )
>
> labels <- c(0, 1, 2, 0)
> library(class)
> k <- 4
>
> item_to_classify1 <- data.frame(Sweetness = 6,Crunchiness = 4)
>
> item_to_classify2 <- data.frame(Sweetness = 4,Crunchiness = 9)
>
> class_labels <- c("Fruit", "Vegetable", "Protein")
>
> predicted_class1 <- knn(existing_items, item_to_classify1, labels, k)
> predicted_label1 <- class_labels[predicted_class1]
>
> predicted_class2 <- knn(existing_items, item_to_classify2, labels, k)
> predicted_label2 <- class_labels[predicted_class2]
>
> cat("The item (tomato) is classified as:", predicted_label1, "\n")
The item (tomato) is classified as: Fruit
> cat("The item (carrot) is classified as:", predicted_label2, "\n")
The item (carrot) is classified as: Fruit
>
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

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

Program :

```
library(rpart)
library(rpart.plot)
library(caret)
titanic <- read.csv("G:\\titanic.csv")
titanic$Age[is.na(titanic$Age)] <- mean(titanic$Age, na.rm = TRUE)
titanic$Sex <- as.factor(titanic$Sex)
features <- c("Age", "Sex", "Pclass", "Fare")
titanic <- titanic[, c("Survived", features)]
set.seed(123)
trainIndex <- createDataPartition(titanic$Survived, p = 0.8,list = FALSE,times = 1)
trainData <- titanic[trainIndex,]
testData <- titanic[-trainIndex,]
titanicTree <- rpart(Survived ~ ., data = trainData, method = "class")
rpart.plot(titanicTree)
predictions <- predict(titanicTree, testData, type = "class")
#confusionMatrix(predictions, testData$Survived)
summary(titanicTree)
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Output :

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
>
> summary(titanicTree)
Call:
rpart(formula = Survived ~ ., data = trainData, method = "class")
  n= 1048

          CP nsplit rel error    xerror       xstd
1 0.10181818      0 1.0000000 1.0000000 0.05178962
2 0.02181818      2 0.7963636 0.7963636 0.04786146
3 0.01272727      3 0.7745455 0.8363636 0.04872224
4 0.01000000      5 0.7490909 0.8254545 0.04849212

Variable importance
   Sex Pclass    Age   Fare
    52     18     16     13

Node number 1: 1048 observations,    complexity param=0.1018182
  predicted class=0  expected loss=0.2624046  P(node) =1
    class counts:   773   275
   probabilities: 0.738 0.262
  left son=2 (673 obs) right son=3 (375 obs)
  Primary splits:
      Sex    splits as  LR,           improve=62.282140, (0 missing)
      Fare   < 10.9208  to the left,  improve=21.490130, (0 missing)
      Pclass < 2.5      to the right, improve=20.544210, (0 missing)
      Age    < 5.5      to the right, improve= 8.807734, (0 missing)
  Surrogate splits:
      Fare < 75.24585 to the left,  agree=0.665, adj=0.064, (0 split)
      Age  < 5.5      to the right, agree=0.645, adj=0.008, (0 split)

Node number 2: 673 observations,    complexity param=0.01272727
  predicted class=0  expected loss=0.1337296  P(node) =0.6421756
    class counts:   583    90
   probabilities: 0.866 0.134
  left son=4 (650 obs) right son=5 (23 obs)
  Primary splits:
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
  Primary splits:
      Age    < 4.5      to the right, improve=8.867407, (0 missing)
      Pclass < 1.5      to the right, improve=7.602696, (0 missing)
      Fare   < 26.26875 to the left,  improve=7.387627, (0 missing)

Node number 3: 375 observations,    complexity param=0.1018182
  predicted class=0  expected loss=0.4933333  P(node) =0.3578244
    class counts:   190   185
   probabilities: 0.507 0.493
  left son=6 (177 obs) right son=7 (198 obs)
  Primary splits:
      Pclass < 2.5      to the right, improve=18.397160, (0 missing)
      Fare   < 10.48125 to the left,  improve= 5.906829, (0 missing)
      Age    < 31.5     to the left,  improve= 1.962379, (0 missing)
  Surrogate splits:
      Fare < 20.7875  to the left,  agree=0.813, adj=0.605, (0 split)
      Age  < 28.5     to the left,  agree=0.661, adj=0.282, (0 split)

Node number 4: 650 observations
  predicted class=0  expected loss=0.1184615  P(node) =0.620229
    class counts:   573    77
   probabilities: 0.882 0.118

Node number 5: 23 observations,    complexity param=0.01272727
  predicted class=1  expected loss=0.4347826  P(node) =0.02194656
    class counts:    10    13
   probabilities: 0.435 0.565
  left son=10 (14 obs) right son=11 (9 obs)
  Primary splits:
      Pclass < 2.5      to the right, improve=3.0979990, (0 missing)
      Fare   < 20.825   to the right, improve=1.7150620, (0 missing)
      Age    < 1.5      to the right, improve=0.6428094, (0 missing)
  Surrogate splits:
      Age  < 0.96     to the right, agree=0.696, adj=0.222, (0 split)
      Fare < 64.37915 to the left,  agree=0.696, adj=0.222, (0 split)
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

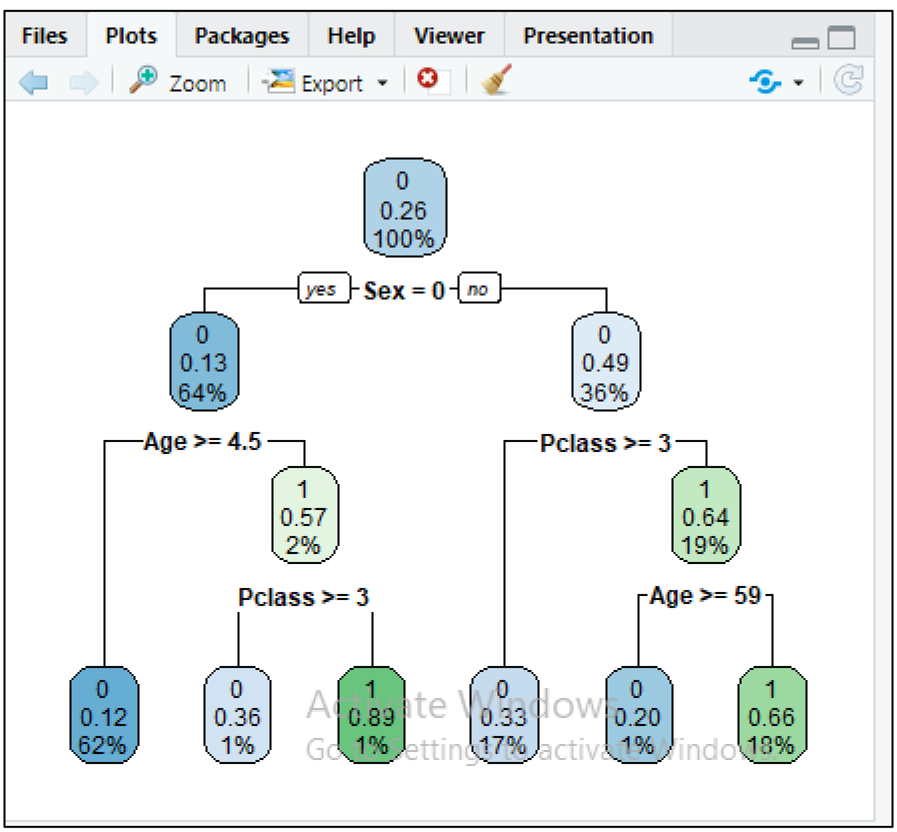
****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

```
      Fare < 64.37915 to the left,  agree=0.696, adj=0.222, (0 split)

Node number 6: 177 observations
  predicted class=0  expected loss=0.3276836  P(node) =0.1688931
    class counts:   119    58
   probabilities: 0.672 0.328

Node number 7: 198 observations,    complexity param=0.02181818
  predicted class=1  expected loss=0.3585859  P(node) =0.1889313
    class counts:    71   127
   probabilities: 0.359 0.641
  left son=14 (10 obs) right son=15 (188 obs)
  Primary splits:
      Age    < 58.5    to the right, improve=4.10421200, (0 missing)
      Fare   < 20.25   to the right, improve=2.97844200, (0 missing)
      Pclass < 1.5     to the left,  improve=0.05875154, (0 missing)

Node number 10: 14 observations
  predicted class=0  expected loss=0.3571429  P(node) =0.01335878
    class counts:     9     5
   probabilities: 0.643 0.357

Node number 11: 9 observations
  predicted class=1  expected loss=0.1111111  P(node) =0.008587786
    class counts:     1     8
   probabilities: 0.111 0.889

Node number 14: 10 observations
  predicted class=0  expected loss=0.2  P(node) =0.009541985
    class counts:     8     2
   probabilities: 0.800 0.200

Node number 15: 188 observations
  predicted class=1  expected loss=0.3351064  P(node) =0.1793893
    class counts:    63   125
   probabilities: 0.335 0.665
```



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Q3) Naïve Bayes Classifier-- Predict whether to play or not to play on the 15th day using naive bayes classifier using R programming by a csv file.

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	No
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	

Program :

```
library(e1071)
data <- read.csv("C:\\Users\\DELL\\Downloads\\play_data.csv", header = TRUE)
data$Outlook <- as.factor(data$Outlook)
data$Temp <- as.factor(data$Temp)
data$Humidity <- as.factor(data$Humidity)
data$Wind <- as.factor(data$Wind)
data$Play <- as.factor(data$Play)
new_data <- data.frame(
  Day = 14,
  Outlook = "Sunny",
  Temp = "Cool",
  Humidity = "High",
  Wind = "Strong",
  Play = "?"
)
data <- data[-nrow(data), ]
model <- naiveBayes(Play ~ Temp + Outlook + Humidity + Wind, data = data)
predictions <- predict(model, newdata = new_data, type = "class")
print(predictions)
```

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Output :

```
> library(e1071)
> data <- read.csv("C:\\Users\\DELL\\Downloads\\play_data.csv", header = TRUE)
> data$Outlook <- as.factor(data$Outlook)
> data$Temp <- as.factor(data$Temp)
> data$Humidity <- as.factor(data$Humidity)
> data$Wind <- as.factor(data$Wind)
> data$Play <- as.factor(data$Play)
> new_data <- data.frame(
+   Day = 14,
+   Outlook = "Sunny",
+   Temp = "Cool",
+   Humidity = "High",
+   Wind = "Strong",
+   Play = "?"
+ )
> data <- data[-nrow(data), ]
> model <- naiveBayes(Play ~ Temp + Outlook + Humidity + Wind, data = data)
> predictions <- predict(model, newdata = new_data, type = "class")
> print(predictions)
[1] No
Levels:  No Yes
> 
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Q4) 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")
a=tissue_gene_expression
getwd()
print(a)
write.csv(a,file="R_Assignment3.c2.csv",row.names=FALSE)
library("factoextra")
library("cluster")
k1=as.numeric(unlist(a))
km=kmeans(k1,centers=7,nstart=6)
km
```

Output :

```
> km
K-means clustering with 7 clusters of sizes 7634, 2895, 14487, 18117, 17826, 15297, 18433

Cluster means:
       [,1]
1 10.223761
2 12.114900
3  8.981408
4  7.129415
5  8.052818
6  5.112177
7  6.161241

Clustering vector:
   [1] 1 1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 3 1 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3
  [40] 3 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 3 1 3 3 1 3 3 3 1 3 3 3 1
  [79] 3 3 3 1 3 1 3 3 3 1 1 1 3 1 1 1 3 1 1 1 3 3 3 3 1 3 3 3 3 3 1 1 1 3 3 1 3 1 1
 [118] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [157] 3 3 3 3 3 5 3 3 3 3 3 5 3 3 3 3 5 3 3 3 3 3 5 3 3 3 5 1 3 3 3 3 3 5 3 5 5 5 5
 [196] 5 3 3 3 5 5 5 3 3 5 5 5 3 5 3 3 3 5 5 5 3 3 1 1 5 5 5 5 5 3 3 3 5 5 5 5 4 5 5
 [235] 5 5 5 5 5 5 5 5 5 5 5 4 5 5 5 5 5 3 5 5 5 5 5 5 5 5 5 5 4 5 5 5 4 5 5 4 5 5 5
 [274] 5 5 5 1 1 1 1 1 1 1 1 1 1 1 3 1 1 3 1 1 3 1 1 1 1 1 2 1 1 1 1 1 3 1 5 5 3 5 5
 [313] 3 5 3 5 3 5 3 3 1 5 3 3 4 5 3 4 5 4 5 3 4 5 5 5 5 5 3 5 5 5 3 3 1 3 3 3 3 5 5
 [352] 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 5 4 5 3 5 6 7 7 7 7 7 7 6 4 7 7 7
 [391] 7 4 7 7 7 7 7 7 7 7 7 7 6 7 7 7 6 6 5 5 3 3 5 7 3 5 5 3 3 3 3 5 5 3 5 3 3 5 5
 [430] 5 5 3 3 3 5 4 3 3 5 5 3 5 3 4 5 3 5 5 3 3 3 1 3 3 1 3 5 3 1 1 3 1 3 3 3 5 5 7
 [469] 4 5 5 5 5 3 5 4 5 4 4 4 5 5 4 3 5 3 3 5 3 3 3 3 3 5 4 5 3 3 3 3 1 5 4 3 3 3 3
 [508] 5 5 3 1 3 3 3 1 5 3 1 1 3 3 1 1 3 5 3 3 5 1 1 1 3 3 7 7 5 5 5 3 3 3 3 3 5 3 4
 [547] 5 7 7 5 5 4 3 5 4 3 5 3 3 5 3 1 1 1 1 1 5 3 3 5 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [586] 3 3 3 3 3 3 3 3 3 3 1 1 5 5 5 5 5 5 5 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3
 [625] 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 3 1 3 3 3 3 3 3
 [664] 3 3 3 3 3 3 3 3 3 3 3 5 5 3 3 3 5 3 3 3 3 3 5 5 3 3 5 3 3 3 3 3 3 3 3 3 3 3 3
 [703] 5 3 3 5 5 3 3 3 3 5 3 3 3 3 3 5 3 5 3 1 1 3 3 1 3 3 3 3 3 3 3 3 1 3 1 1 3 1 1
 [742] 3 3 1 3 1 1 3 1 3 5 5 5 3 3 3 7 7 7 7 6 7 7 7 7 6 7 7 6 7 7 7 7 7 7 7 6 7 7 7
 [781] 7 7 7 6 6 6 6 6 7 7 6 6 7 7 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
```

```
 [820] 6 6 6 6 6 6 6 6 6 6 7 7 6 6 7 6 6 7 7 7 7 7 7 7 7 7 6 6 6 6 7 7 7 6 7 7 7 7 7
 [859] 7 7 7 6 7 7 7 7 7 7 7 6 7 6 6 7 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 6 6 7 6
 [898] 6 6 6 7 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 [937] 6 6 6 7 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 4 7 7 7 7 7
 [976] 4 7 4 7 4 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 4 7
 [ reached getOption("max.print") -- omitted 93689 entries ]

Within cluster sum of squares by cluster:
[1] 1395.152 1463.557 1346.104 1343.832 1265.100 2550.184 1528.993
 (between_SS / total_SS =  96.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
> 
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(410908) Data Science Laboratory

****

Class : SY-MCA Shift / Div : S3/B Roll Number : 52147

Name : Nisha Harish Parekh Assignment No : 3 Date of Implementation : 16/10/2023

****

Q5) 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 :

```
df=iris[,-ncol(iris)]
df<-scale(df)
df<-as.data.frame(df)
install.packages("dbscan")
library(dbscan)
kNNdistplot(df,k=5)
abline(h=0.8,col="red")
```

Output :