Name: Vikram Surya R
Reg No: 19BCE1490

Essentials of Data Analytics
Tasks for Week-5: K- NN algorithm

Aim:

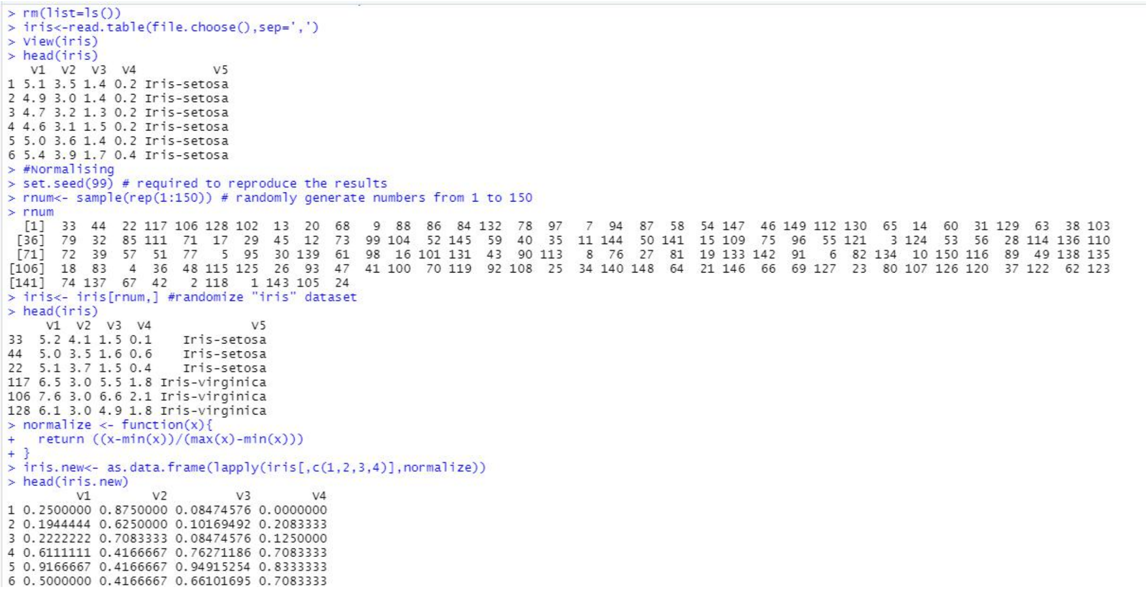
Understand the following operations/functions on to perform K- NN algorithm and perform similar operations on ‘iris’ dataset based on given instructions.

ALGORITHM :

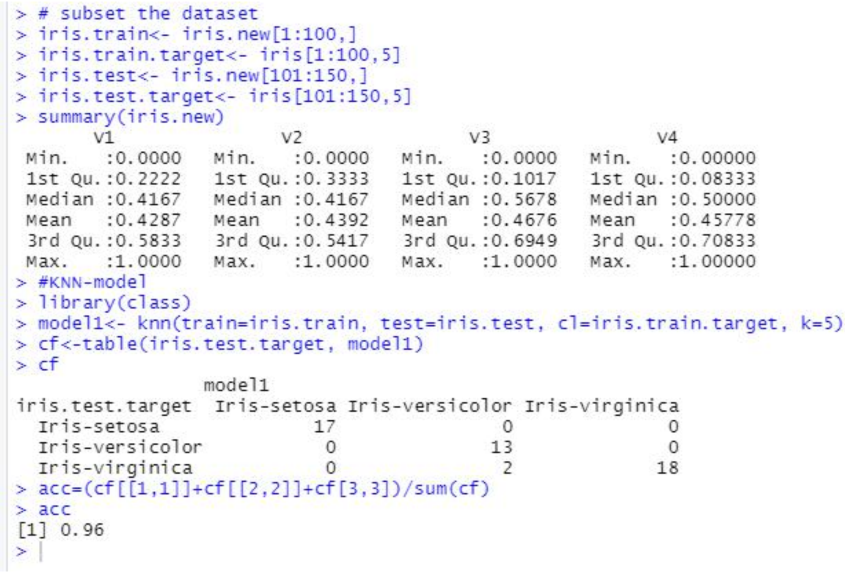
- Read the iris dataset using read.table()
- Normalised the dataset save it as iris.new
- Then subset the dataset.
- For comparing the original dataset and normalized dataset take 2 to 5 columns of both data set and apply summary() function to find summary.
- Import class using library() function.
- Do the K-NN algorithm and store all the predicted values in model1 variable.
- Form the confusion matrix form using predicted data from model1 and iris.test.target.
- Find the accuracy of the data.

INFERENCE :

Since the accuracy of the model is 0.96 it is accepted.



```
> rm(list=ls())
> iris<-read.table(file.choose(),sep=',')
> View(iris)
> head(iris)
   V1  V2  V3  V4          V5
1 5.1 3.5 1.4 0.2 Iris-setosa
2 4.9 3.0 1.4 0.2 Iris-setosa
3 4.7 3.2 1.3 0.2 Iris-setosa
4 4.6 3.1 1.5 0.2 Iris-setosa
5 5.0 3.6 1.4 0.2 Iris-setosa
6 5.4 3.9 1.7 0.4 Iris-setosa
> #Normalising
> set.seed(99) # required to reproduce the results
> rnum<- sample(rep(1:150)) # randomly generate numbers from 1 to 150
> rnum
  [1]  33  44  22 117 106 128 102  13  20  68   9  88  86  84 132  78  97   7  94  87  58  54 147  46 149 112 130  65  14  60  31 129  63  38 103
 [36]  79  32  85 111  71  17  29  45  12  73  99 104  52 145  59  40  35  11 144  50 141  15 109  75  96  55 121   3 124  53  56  28 114 136 110
 [71]  72  39  57  51  77   5  95  30 139  61  98  16 101 131  43  90 113   8  76  27  81  19 133 142  91   6  82 134  10 150 116  89  49 138 135
[106]  18  83   4  36  48 115 125  26  93  47  41 100  70 119  92 108  25  34 140 148  64  21 146  66  69 127  23  80 107 126 120  37 122  62 123
[141]  74 137  67  42   2 118   1 143 105  24
> iris<- iris[rnum,] #randomize "iris" dataset
> head(iris)
     V1  V2  V3  V4              V5
33  5.2 4.1 1.5 0.1     Iris-setosa
44  5.0 3.5 1.6 0.6     Iris-setosa
22  5.1 3.7 1.5 0.4     Iris-setosa
117 6.5 3.0 5.5 1.8 Iris-virginica
106 7.6 3.0 6.6 2.1 Iris-virginica
128 6.1 3.0 4.9 1.8 Iris-virginica
> normalize <- function(x){
+   return ((x-min(x))/(max(x)-min(x)))
+ }
> iris.new<- as.data.frame(lapply(iris[,c(1,2,3,4)],normalize))
> head(iris.new)
         V1        V2         V3        V4
1 0.2500000 0.8750000 0.08474576 0.0000000
2 0.1944444 0.6250000 0.10169492 0.2083333
3 0.2222222 0.7083333 0.08474576 0.1250000
4 0.6111111 0.4166667 0.76271186 0.7083333
5 0.9166667 0.4166667 0.94915254 0.8333333
6 0.5000000 0.4166667 0.66101695 0.7083333
```



```
> # subset the dataset
> iris.train<- iris.new[1:100,]
> iris.train.target<- iris[1:100,5]
> iris.test<- iris.new[101:150,]
> iris.test.target<- iris[101:150,5]
> summary(iris.new)
       V1               V2               V3               V4         
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
 1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333  
 Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000  
 Mean   :0.4287   Mean   :0.4392   Mean   :0.4676   Mean   :0.45778  
 3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000  
> #KNN-model
> library(class)
> model1<- knn(train=iris.train, test=iris.test, cl=iris.train.target, k=5)
> cf<-table(iris.test.target, model1)
> cf
                 model1
iris.test.target  Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              17               0              0
  Iris-versicolor           0              13              0
  Iris-virginica            0               2             18
> acc=(cf[[1,1]]+cf[[2,2]]+cf[3,3])/sum(cf)
> acc
[1] 0.96
> 
```

CODE -

```
rm(list=ls())
iris<-read.table(file.choose(),sep=',')
View(iris)
head(iris)

#Normalising
set.seed(99) # required to reproduce the results
rnum<- sample(rep(1:150)) # randomly generate numbers from 1 to 150
rnum
iris<- iris[rnum,] #randomize "iris" dataset
head(iris)
normalize <- function(x){
   return ((x-min(x))/(max(x)-min(x)))
}
iris.new<- as.data.frame(lapply(iris[,c(1,2,3,4)],normalize))
head(iris.new)

# subset the dataset
iris.train<- iris.new[1:100,]
iris.train.target<- iris[1:100,5]
iris.test<- iris.new[101:150,]
iris.test.target<- iris[101:150,5]
summary(iris.new)

#KNN-model
library(class)
model1<- knn(train=iris.train, test=iris.test, cl=iris.train.target, k=5)
cf<-table(iris.test.target, model1)
cf
acc=(cf[[1,1]]+cf[[2,2]]+cf[3,3])/sum(cf)
acc
```