Ex.No.6 **k Means Clustering Algorithm**

Date: 5-09-23

Aim

To implement k Means Clustering Algorithm under unsupervised machine learning by grouping the data by similarity through R programming.

Procedure

1. To do programming in R, first install "RStudio" and "R" in the system. RStudio is an integrated development environment [IDE] for R and python.
2. Select the File in taskbar →open New file →R script or use shortcut "ctrl+shift+N"
3. Install the 'ClusterR, cluster' package and load it in R.
4. Import the built-in dataset 'iris'
5. Apply the k Means Algorithm on the iris dataset.
6. Write the program in the script and save it using the extension R.
7. Run the program by clicking Run option or use the shortcut "ctrl+enter".
8. See the output in the console tab.

Concepts Involved

- Applying the k Means Clustering algorithm on the data set.

k-Means Clustering Algorithm

k Means Clustering in R programming is an unsupervised Non-linear algorithm that cluster data based on similarity. Segmentation of data takes place to assign each training example to a segment called as cluster. It is used in a variety of fields like Banking, healthcare, retail, Media etc.,

Algorithm:

1. Choose the number K clusters.

2. Select at random K points, the centroids (Not necessarily from the given data).

3. Assign each data point to closest centroid that forms K clusters.

4. Compute and place the new centroid of each centroid.

5. Reassign each data point to new cluster.

6. After final reassignment, name the cluster as Final cluster.

Dataset

Here, Iris dataset consisting of 50 samples from each of 3 species Iris (Iris setosa, Iris virginica, Iris versicolor) is used. Four features were measured from each sample i.e., length and width of the sepals and petals and based on a combination of these four features.

Script

```
# Installing Packages

install.packages("ClusterR")

install.packages("cluster")

# Loading package

library(ClusterR)

library(cluster)

# Removing initial label of

# Species from original dataset

iris_1 <- iris[, -5]

# Fitting K-Means clustering Model

# to training dataset

set.seed(240) # Setting seed

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

kmeans.re
```

Output

```
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
 [46] 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 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [91] 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
[136] 3 3 3 2 3 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"
[7] "size"         "iter"         "ifault"
```

Script

Cluster identification for

each observation

kmeans.re$cluster

Output

```
[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
[46] 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 2 2 2 2
2 2 3 2 2 2 2 2 2 2 2 2 2 2 2
[91] 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
[136] 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2
```

Script

Confusion Matrix

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

cm

Output

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

Script

Model Evaluation and visualization

plot(iris_1[c("Sepal.Length", "Sepal.Width")])

plot(iris_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster)

plot(iris_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster,

```
    main = "K-means with 3 clusters")
## Plotiing cluster centers
kmeans.re$centers
```

Output

```
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
```

Script

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

Output

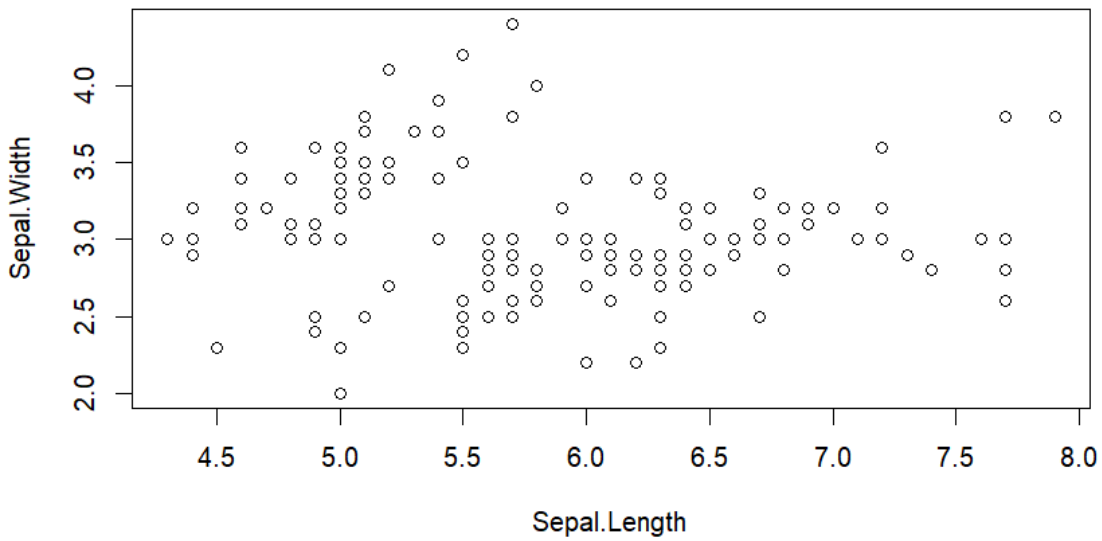
```
Sepal.Length Sepal.Width
1     5.006000    3.428000
2     5.901613    2.748387
3     6.850000    3.073684
```

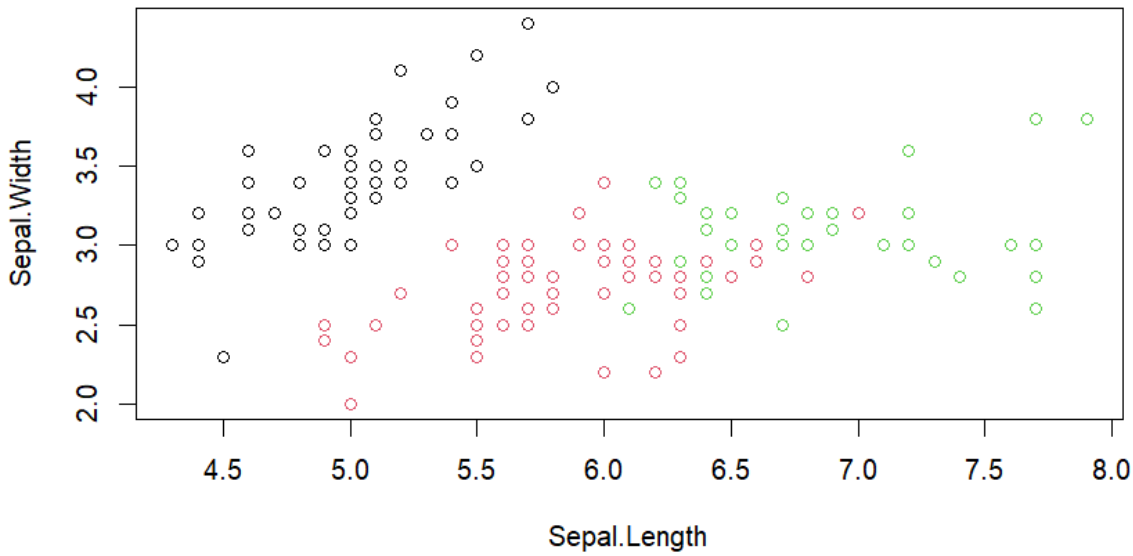
Script

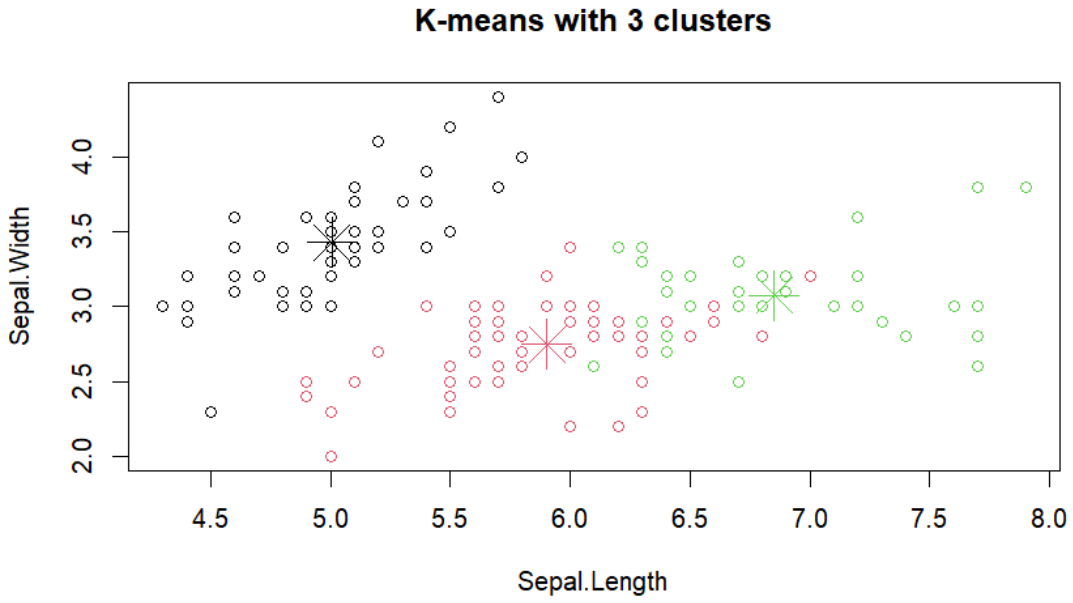
```
# cex is font size, pch is symbol
points(kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")],
      col = 1:3, pch = 8, cex = 3)
## Visualizing clusters
y_kmeans <- kmeans.re$cluster
clusplot(iris_1[, c("Sepal.Length", "Sepal.Width")],
        y_kmeans,
        lines = 0,
        shade = TRUE,
        color = TRUE,
        labels = 2,
        plotchar = FALSE,
        span = TRUE,
        main = paste("Cluster iris"),
```

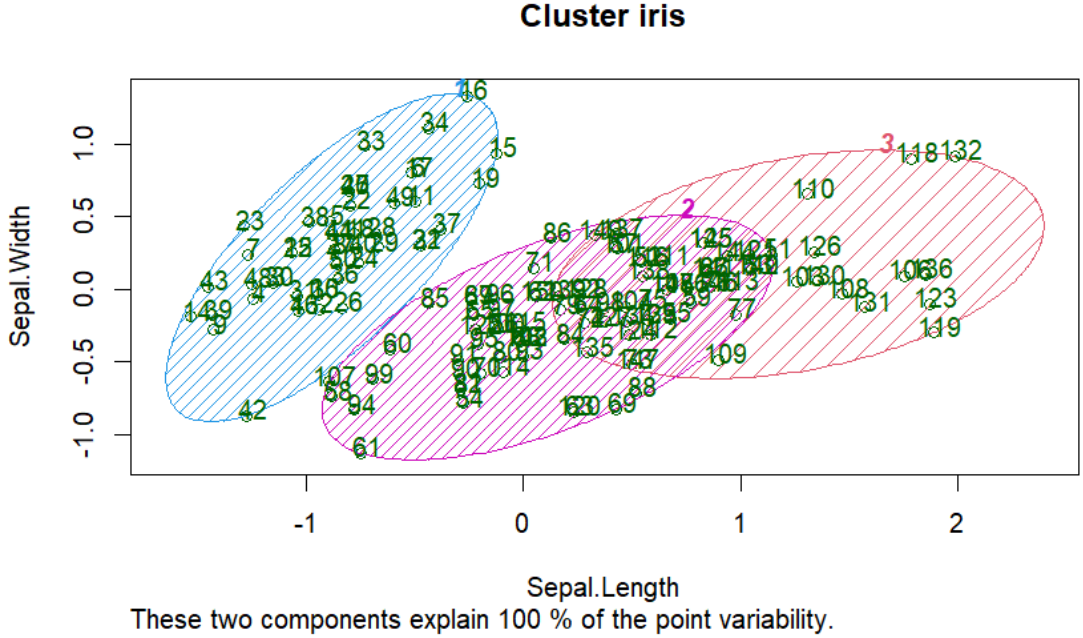
```
    xlab = 'Sepal.Length',
    ylab = 'Sepal.Width')
```

Output









Result

Thus the k Means Clustering Algorithm has been successfully implemented on Iris data set using R programming.