

Ex No 9**Implement clustering techniques – Hierarchical and K-Means****AIM:**

To Implement clustering techniques – Hierarchical and K-Means using R.

PROCEDURE:

- Collect and load the dataset from sources like CSV files or databases.
- Clean and preprocess the data, including handling missing values and scaling features.
- Determine the number of clusters (K) for K-Means, or decide on the stopping criterion for Hierarchical Clustering.
- Choose the appropriate clustering algorithm: K-Means for partitioning, Hierarchical for nested clustering.
- Apply the K-Means algorithm using `fit_predict` to assign data points to clusters.
- Apply the Hierarchical Clustering algorithm using Agglomerative Clustering for hierarchical clusters.
- Visualize the clusters with scatter plots for K-Means, and dendrograms for Hierarchical Clustering.
- Evaluate clustering performance using metrics like silhouette score or inertia (for K-Means).
- Fine-tune the clustering by adjusting the number of clusters or linkage criteria.
- Interpret the results to understand the structure and relationships within the data.

CODE:**Hierarchical Clustering.R:**

```
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Compute the distance matrix
distance_matrix <- dist(iris_scaled, method = "euclidean")
# Perform hierarchical clustering using the "complete" linkage method
hc_complete <- hclust(distance_matrix, method = "complete")
# Plot the dendrogram
plot(hc_complete, main = "Hierarchical Clustering Dendrogram", xlab = "", sub = "", cex = 0.6)
```

```
# Cut the tree to form 3 clusters
clusters <- cutree(hc_complete, k = 3)
# Print the cluster memberships
print(clusters)
# Add the clusters to the original dataset
iris$Cluster <- as.factor(clusters)
# Display the first few rows of the updated dataset
head(iris)
```

K-Means Clustering.R:

```
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Set the number of clusters
set.seed(123) # For reproducibility
k <- 3 # Number of clusters
# Perform K-Means clustering
kmeans_result <- kmeans(iris_scaled, centers = k, nstart = 25)
# Print the K-Means result
print(kmeans_result)
# Print the cluster centers
print(kmeans_result$centers)
# Add the cluster assignments to the original dataset
iris$Cluster <- as.factor(kmeans_result$cluster)
# Display the first few rows of the updated dataset
head(iris)
# Plot the clusters
library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +
  geom_point(size = 3) +
  labs(title = "K-Means Clustering of Iris Dataset", x = "Sepal Length", y = "Sepal Width")
```

The top panel shows R code for hierarchical clustering:

```
1 data(iris)
2 # Use only the numeric columns for clustering (exclude the Species column)
3 iris_data <- iris[, -5]
4 # Standardize the data
5 iris_scaled <- scale(iris_data)
6 # Compute the distance matrix
7 distance_matrix <- dist(iris_scaled, method = "euclidean")
8 # Perform hierarchical clustering using the "complete" linkage method
9 hc_complete <- hclust(distance_matrix, method = "complete")
10 # Plot the dendrogram
11 plot(hc_complete, main = "Hierarchical Clustering Dendrogram", xlab = "", sub = "", cex = 0.6)
12 # Cut the tree to form 3 clusters
13 clusters <- cutree(hc_complete, k = 3)
14 # Print the cluster memberships
15 print(clusters)
16 # Add the clusters to the original dataset
17 iris$Cluster <- as.factor(clusters)
18 # Display the first few rows of the updated dataset
19 head(iris)
20
```

The bottom panel shows the output of the code:

```
> # Display the first few rows of the updated dataset
> 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         0.2 setosa
```

K-Means Clustering: