

Exp.No: 9

Implement clustering techniques – Hierarchical and K-Means

AIM:

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

PROCEDURE:

- Load the dataset from sources such as CSV files or databases using appropriate libraries.
- Perform data cleaning and preprocessing, including handling missing values and scaling features for consistency.
- Decide on the number of clusters (K) for K-Means or establish the stopping criterion for Hierarchical Clustering.
- Select the suitable clustering algorithm: utilize K-Means for partitioning or Hierarchical Clustering for creating nested groups.
- Implement the K-Means algorithm using `fit_predict` to categorize data points into clusters.
- Use Agglomerative Clustering for Hierarchical Clustering to create a hierarchy of clusters.
- Visualize the resulting clusters using scatter plots for K-Means and dendrograms for Hierarchical Clustering.
- Assess clustering performance through metrics such as silhouette score or inertia for K-Means.
- Tune the clustering process by modifying the number of clusters or adjusting linkage criteria.
- Analyze the results to gain insights into the data structure and the relationships among clusters.

PROGRAM:

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
```

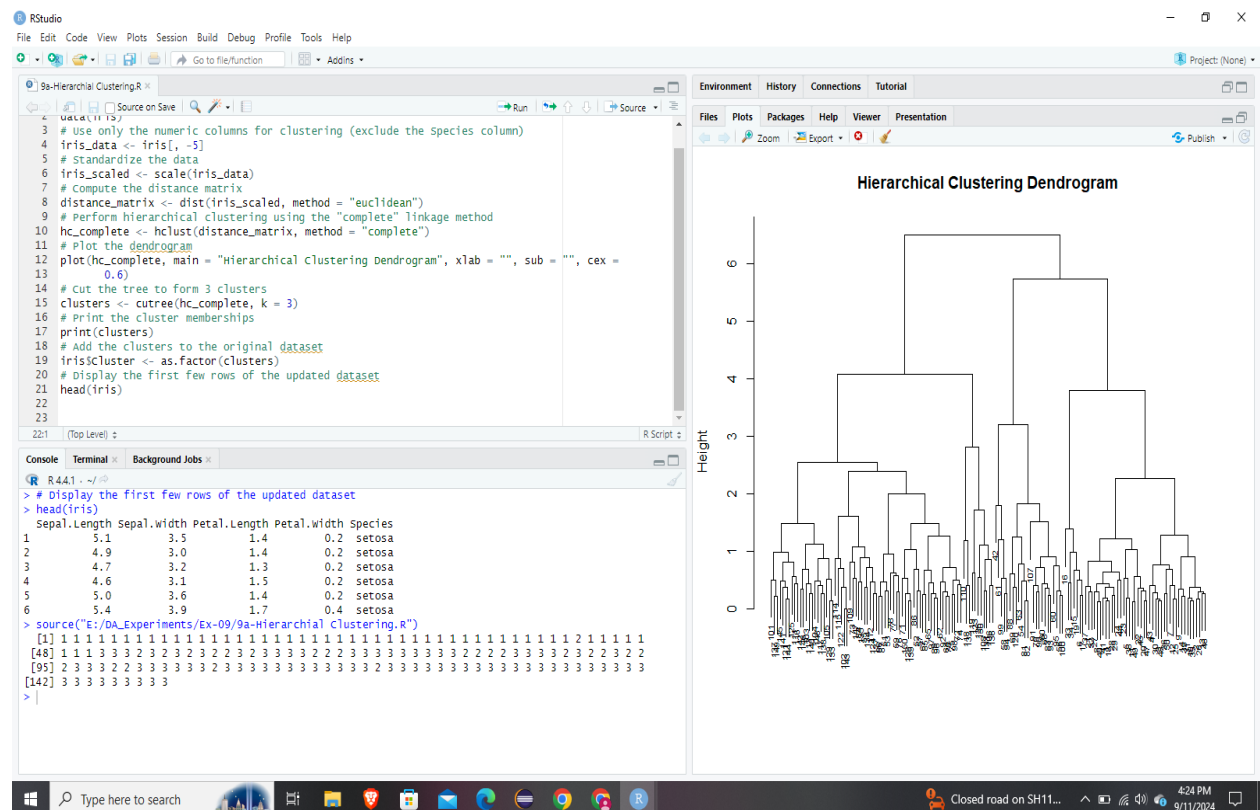
```

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")

```

OUTPUT:

Hierarchical Clustering:



The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for K-Means clustering. The code loads the Iris dataset, excludes the species column, standardizes the data, sets the number of clusters to 3, and performs the clustering using the `kmeans` function. It also prints the results and cluster centers.
- Environment:** Shows the objects created in the global environment, including `cluster_centers`, `hc_complete`, `iris`, `iris_data`, `iris_scaled`, `kmeans_result`, and `p`.
- Console:** Displays the output of the R script, including the clustering vector, the within-cluster sum of squares, and the available components of the `kmeans` object.
- Plots:** A scatter plot titled "K-Means Clustering of Iris Dataset" showing the Sepal Length (x-axis) versus Sepal Width (y-axis) for the Iris dataset, colored by cluster (1, 2, 3).