

EXP NO: 9

IMPLEMENT CLUSTERING TECHNIQUES – HIERARCHICAL AND

K- MEANS

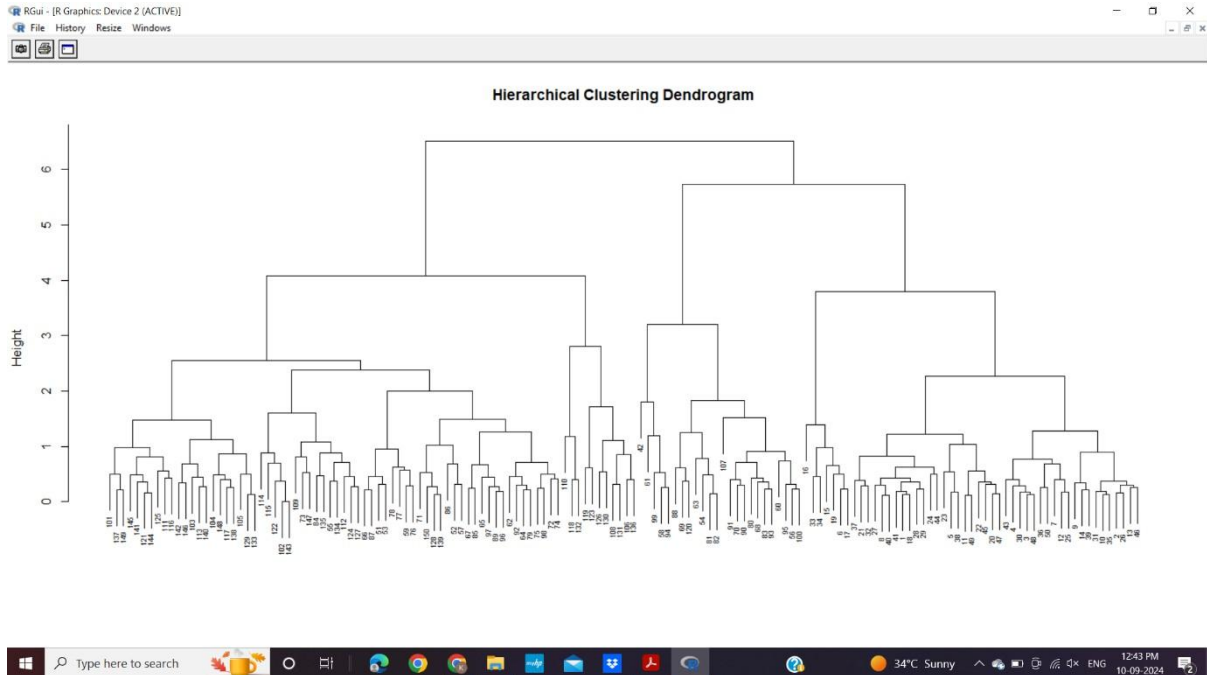
a) HIERARCHIAL CLUSTERING

```
# 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)
```

OUTPUT:

```
RGui
File Edit Packages Windows Help

# Console
> print(confusion_matrix)
      Actual
Predicted setosa versicolor virginica
setosa      14         0         0
versicolor  0        18         1
virginica   0         0        12
> # Calculate accuracy
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> cat("Accuracy:", accuracy * 100, "%\n")
Accuracy: 97.77778 %
> # 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)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Cluster
1         5.1         3.5         1.4         0.2 setosa      1
2         4.9         3.0         1.4         0.2 setosa      1
3         4.7         3.2         1.3         0.2 setosa      1
4         4.6         3.1         1.5         0.2 setosa      1
5         5.0         3.6         1.4         0.2 setosa      1
6         5.4         3.9         1.7         0.4 setosa      1
```



b) K-MEANS CLUSTERING

```
# 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:

