**K-Means Clustering on Iris Dataset**

**Aim**

To implement the K-Means clustering algorithm using the Iris dataset and visualize the clusters in a 2D plot using the first two features of the data.

**Procedure**

1. Load the Iris dataset using sklearn.datasets.load\_iris().
2. Ask the user for the desired number of clusters (k).
3. Apply the K-Means clustering algorithm using KMeans from sklearn.cluster.
4. Fit the model and obtain the cluster labels and centers.
5. Display the cluster centers.
6. Plot the clusters on a 2D scatter plot using the first two features (sepal length and sepal width).
7. Mark the cluster centers on the plot with red "X" markers.

**Program**

from sklearn.datasets import load\_iris

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

data = load\_iris()

X = data.data

# Ask user for number of clusters

k = int(input("Enter number of clusters (e.g., 3): "))

# Apply K-Means

kmeans = KMeans(n\_clusters=k, random\_state=0)

kmeans.fit(X)

labels = kmeans.labels\_

# Print cluster centers

print("Cluster centers:\n", kmeans.cluster\_centers\_)

plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1],

color='red', marker='X', s=200, label='Centers')

plt.xlabel(data.feature\_names[0])

plt.ylabel(data.feature\_names[1])

plt.title("K-Means Clustering")

plt.legend()

plt.show()

**Input**

Enter number of clusters (e.g., 3): 3

**Output**

Cluster centers:

[[5.006 3.428 1.462 0.246]

[5.906 2.748 4.26 1.3 ]

[6.588 2.974 5.552 2.026]]

**Result**

The program successfully applies the K-Means clustering algorithm to the Iris dataset, clusters the flowers based on the first two features (sepal length and sepal width), and visualizes the clusters with a scatter plot. The cluster centers are marked with red "X" markers.