

**Ex. No. 9**

**CLUSTERING ALGORITHM**

**Date:**

**Aim:**

To implement k-Means clustering algorithm to classify the Iris Dataset.

**Algorithm:**

- Step 1. Import the necessary modules from scikit-learn, including KMeans for clustering, load\_iris to load the Iris dataset.
- Step 2. Load the Iris dataset.
- Step 3. Extract the data and target values.
- Step 4. Create a KMeans object with 3 clusters.
- Step 5. Fit the KMeans object to the data.
- Step 6. Get the predicted cluster labels.
- Step 7. Plot the data points and centroids.

**Program:**

```
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt

# Load the Iris dataset
iris = load_iris()

# Extract the data and target values
X = iris.data
y = iris.target

# Create a KMeans object with 3 clusters
kmeans = KMeans(n_clusters=3, n_init=10)

# Fit the KMeans object to the data
kmeans.fit(X)

# Get the predicted cluster labels
labels = kmeans.predict(X)

# Plot the data points and centroids
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[0], kmeans.cluster_centers_[1], marker='*', s=200, c='red')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.show()
```

**Viva questions:**

1. What is the purpose of this code?
2. What is KMeans clustering and how does it work?
3. What is the difference between the "data" and "target" values in the Iris dataset?
4. What is the purpose of the n\_init parameter in the KMeans object?
5. How does the scatter plot generated by this code represent the clusters in the Iris dataset?

**Result:**

Thus, the program for implementing the K-means Algorithm for clustering the Iris dataset was executed successfully, and the output was verified.