**TASK 8**

**Implement K-Means\_Clustering using python**

**ALGORITHM:**

**Step 1:** Read the Given data Sample to X

**Step 2:** Train Dataset with K=5

**Step 3:** Find optimal number of clusters(k) in a dataset using Elbow method

**Step 4:** Train Dataset with K=3 (optimal K-Value)

**Step 4:** Compare results

**Step 6:** End

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.datasets import make\_blobs

# Step 1: Generate synthetic dataset

X, y\_true = make\_blobs(n\_samples=300, centers=4, cluster\_std=0.7, random\_state=42)

# Step 2: Plot the original dataset (unlabeled)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.scatter(X[:, 0], X[:, 1], s=50, color='gray')

plt.title("Original Data (before clustering)")

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

# Step 3: Apply KMeans clustering

kmeans = KMeans(n\_clusters=4, random\_state=42)

y\_kmeans = kmeans.fit\_predict(X)

# Step 4: Plot clustered data

plt.subplot(1, 2, 2)

plt.scatter(X[:, 0], X[:, 1], c=y\_kmeans, cmap='viridis', s=50)

centers = kmeans.cluster\_centers\_

plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, marker='X', label="Centroids")

plt.title("Clustered Data (after K-Means)")

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

plt.legend()

plt.tight\_layout()

plt.show()