**Data Warehousing and Data Mining.**

**Name : Antuley Aman Siraj.**

**Roll No. : 23CO25.**

**Class : TE-CO**

**Batch : 01**

**Experiment - 07**

**Aim:** Implementation of Clustering algorithm (K-means).

**Theory:** Clustering is an unsupervised machine learning technique used to group similar data points together based on their features. Among various clustering algorithms, K-means is one of the most widely used due to its simplicity and efficiency.

**K-means Algorithm Steps:**

1. Choose the number of clusters (k).

2. Initialize k centroids randomly.

3. Assign each data point to the nearest centroid (cluster).

4. Recalculate the centroids as the mean of the points in each cluster.

5. Repeat steps 3 and 4 until the centroids do not change significantly (convergence).

K-means works best when the clusters are spherical and evenly sized. It is sensitive to the initial placement of centroids and the value of k, which must be predefined.

Applications of K-means include:

● Customer segmentation

● Market basket analysis

● Image compression

● Pattern recognition

**Program:**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from sklearn.cluster import KMeans**

**# Reading the database**

**data = pd.read\_csv('/content/cluster\_data.csv')**

**print("\nFirst 5 rows of data:")**

**print(data.head())**

**# Apply K-means clustering**

**k = 3 # You can change the number of clusters**

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

**data['Cluster'] = kmeans.fit\_predict(data[['Feature 1', 'Feature 2']])**

**# Get cluster centroids**

**centroids = kmeans.cluster\_centers\_**

**# Plot clustered points**

**plt.figure(figsize=(8,6))**

**plt.title('K-means Clustering on Features')**

**# Points colored by cluster**

**plt.scatter(data['Feature 1'], data['Feature 2'],**

**c=data['Cluster'], cmap='rainbow', alpha=0.7, edgecolors='k')**

**# Plot centroids**

**plt.scatter(centroids[:, 0], centroids[:, 1],**

**c='black', s=200, marker='X', label='Centroids')**

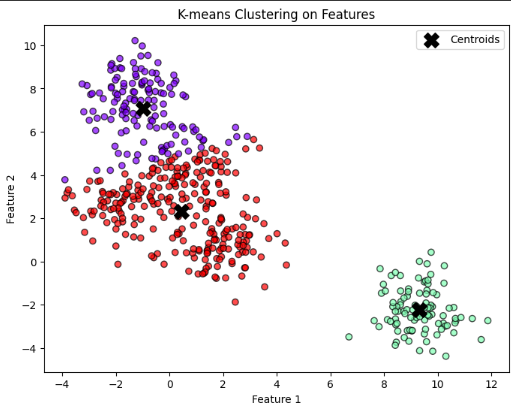
**plt.xlabel("Feature 1")**

**plt.ylabel("Feature 2")**

**plt.legend()**

**plt.show()**

**Output :**

****

**Conclusion:**

In this experiment, the K-means clustering algorithm was successfully implemented using Python. The algorithm grouped data points based on similarity into 3 distinct clusters. By visualizing the clusters and their centroids using a scatter plot, we were able to clearly observe the separation and structure of the data. This demonstrates how K-means can be effectively used in unsupervised learning to discover hidden patterns or groupings within datasets. The experiment also highlights the importance of selecting the right number of clusters (k) and understanding the nature of the data for meaningful results.