K-means Algorithm

Aim

To partition a dataset into **K distinct, non-overlapping clusters**, where each data point belongs to the cluster with the **nearest mean (centroid)**.
It helps in discovering structure or patterns in unlabeled data.

Program (Python using `sklearn`)

```
from sklearn.cluster import KMeans
import numpy as np

# Sample data points: [X, Y] coordinates
X = np.array([[1, 2], [1, 4], [1, 0],
              [10, 2], [10, 4], [10, 0]])

# Create KMeans model with 2 clusters
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)

# Check cluster centers
print("Cluster Centers:\n", kmeans.cluster_centers_)

# Predict which cluster a new point belongs to
prediction = kmeans.predict([[0, 0]])
print("Predicted cluster for [0, 0]:", prediction[0])
```

Output

```
Cluster Centers:
 [[10.  2.]
  [ 1.  2.]]
Predicted cluster for [0, 0]: 1
```

Result

- The dataset is **divided into 2 clusters** based on feature similarity.
- Each cluster has its own **centroid**.
- The new point `[0, 0]` is closest to the second centroid (around `[1, 2]`), so it's assigned to **cluster 1**.
- This shows how K-Means can automatically **group unlabeled data** and classify new points based on learned patterns.