

Exp no: 09
Date:

Implementation of Clustering Techniques - K Means

Aim:
To implement a K-means Clustering
Techniques using python language.

Explanation:

- * import K means from sklearn - Cluster
- * Assign x & y
- * call the function kmeans()
- * Perform scatter operation and display output.

Algorithm:

- * Initialize
 - Choose the no. of clusters k ,
 - Randomly initialize k clusters
- * Assign Data points to clusters
 - for each dataset
 - calculate the distance between the data point and each centroid
 - Assign the data point to the cluster whose centroid is the closest
- * Recalculate Centroids
 - for each cluster, compute the new centroid by calculating the mean of all data points assigned to that cluster.

* Repeat

→ Repeat step 2 and 3 until the Cluster assignment do not change. This is called convergence

* Stopping Criteria:

→ algorithm stops when One of the following occurs

→ The Centroids do not change between iterations

Code:

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.array([1, 2], [1, 4], [1, 0], [4, 2],
              [4, 4], [4, 0], [10, 2], [10, 4], [10, 0])
```

```
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(x)
```

```
y_kmeans = kmeans.predict(x)
```

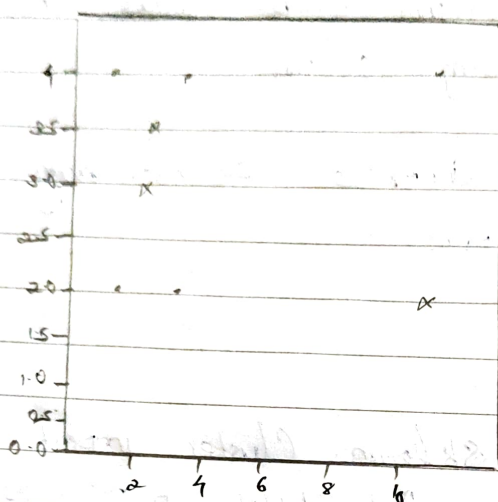
```
plt.scatter(x[:, 0], x[:, 1], c=y_kmeans,
            s=50, cmap='viridis')
```

```
centroids = kmeans.cluster_centers_
```

```
plt.scatter(centroids[:, 0], centroids[:, 1],
            c='red', s=200, alpha=0.75,
            marker='x')
```

```
plt.xlabel("x-axis")  
plt.ylabel("y-axis")  
plt.title("K-means Clustering")  
plt.show()
```

Output:



x-axis

Result:

This is the K-means Clustering program in executed successfully and the output is verified.

~~✗~~

✓