Ex. No.: 10 Write a program for K-means Clustering

AIM:

To write a Python program to calculate the K-means clustering by using the given series of files.

K-meansis an unsupervised learning method for clustering data points. The algorithm iteratively divides data points into K clusters by minimizing the variance in each cluster.

How does it work?

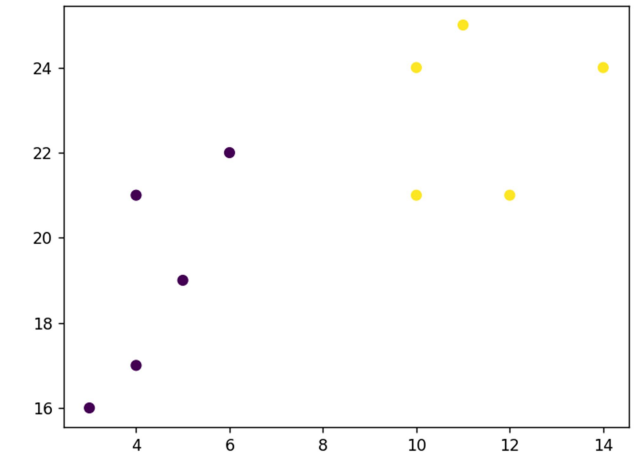
First, each data point is randomly assigned to one of the K clusters. Then, we compute the centroid (functionally the center) of each cluster, and reassign each data point to the cluster with the closest centroid. We repeat this process until the cluster assignments for each data point are no longer changing.

K-means clustering requires us to select K, the number of clusters we want to group the data into. The elbow method lets us graph the inertia (a distance-based metric) and visualize the point at which it starts decreasing linearly. This point is referred to as the "elbow" and is a good estimate for the best value for K based on our data.

(i) Program:

```
importmatplotlib.pyplot as plt

fromsklearn.cluster import KMeans

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

data = list(zip(x, y))

kmeans = KMeans(n_clusters=2)

kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels_)

plt.show()
```

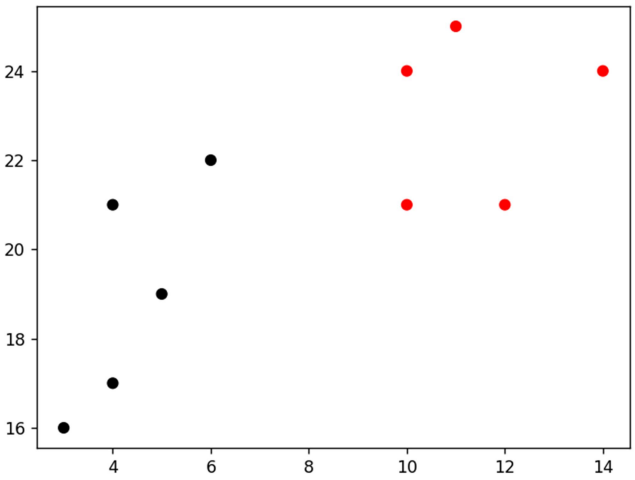
Output:



(ii) Program:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
data = list(zip(x, y))
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
plt.scatter(x, y, c=['red' if label==0 else 'black' for label in kmeans.labels_])
plt.show()
```

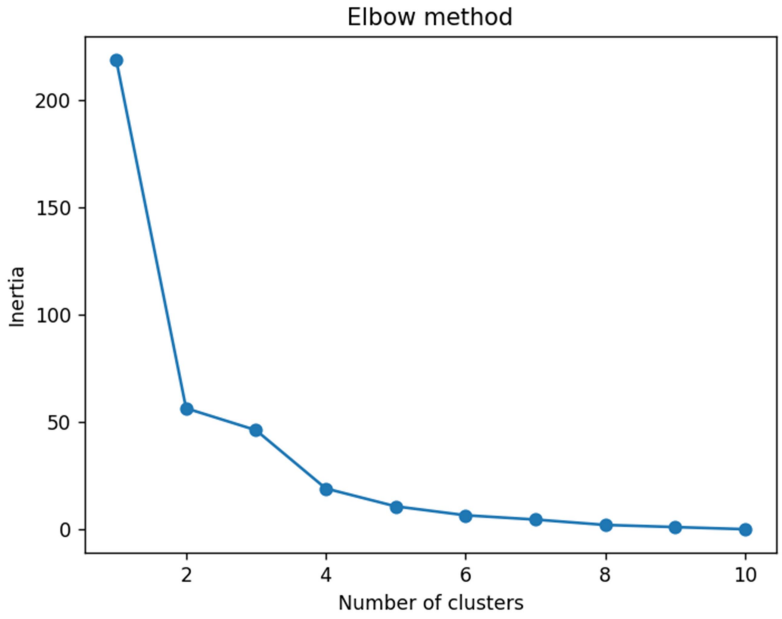
Output:



(iii) Program:

```
importmatplotlib.pyplot as plt
fromsklearn.cluster import KMeans
x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
data = list(zip(x, y))
inertias = []
fori in range(1,11):
kmeans = KMeans(n_clusters=i)
kmeans.fit(data)
inertias.append(kmeans.inertia_)
plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

Output:



The elbow method shows that 2 is a good value for K, so we retrain and visualize the result.