Write a program of cluster analysis using simple k-means algorithm python programming language.

Cluster Analysis:

Cluster Analysis is a statistical method for processing data. It works by organizing items into groups, or clusters on the basis of how closely associated they are.

K-means algorithm:

K-means algorithm is a simple two steps clustering process. The first step is cluster assignment and the second one is the move centroid step. However, this unsupervised algorithm can easily create, implement and handle massive datasets.

Steps involved in k-means Algorithm:

Step 1: Select the number k to decide the number of clusters.

Step 2: Select random k points or centroids.

Step 3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step 4: Calculate the variance and place a new centroid of each cluster.

Step 5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

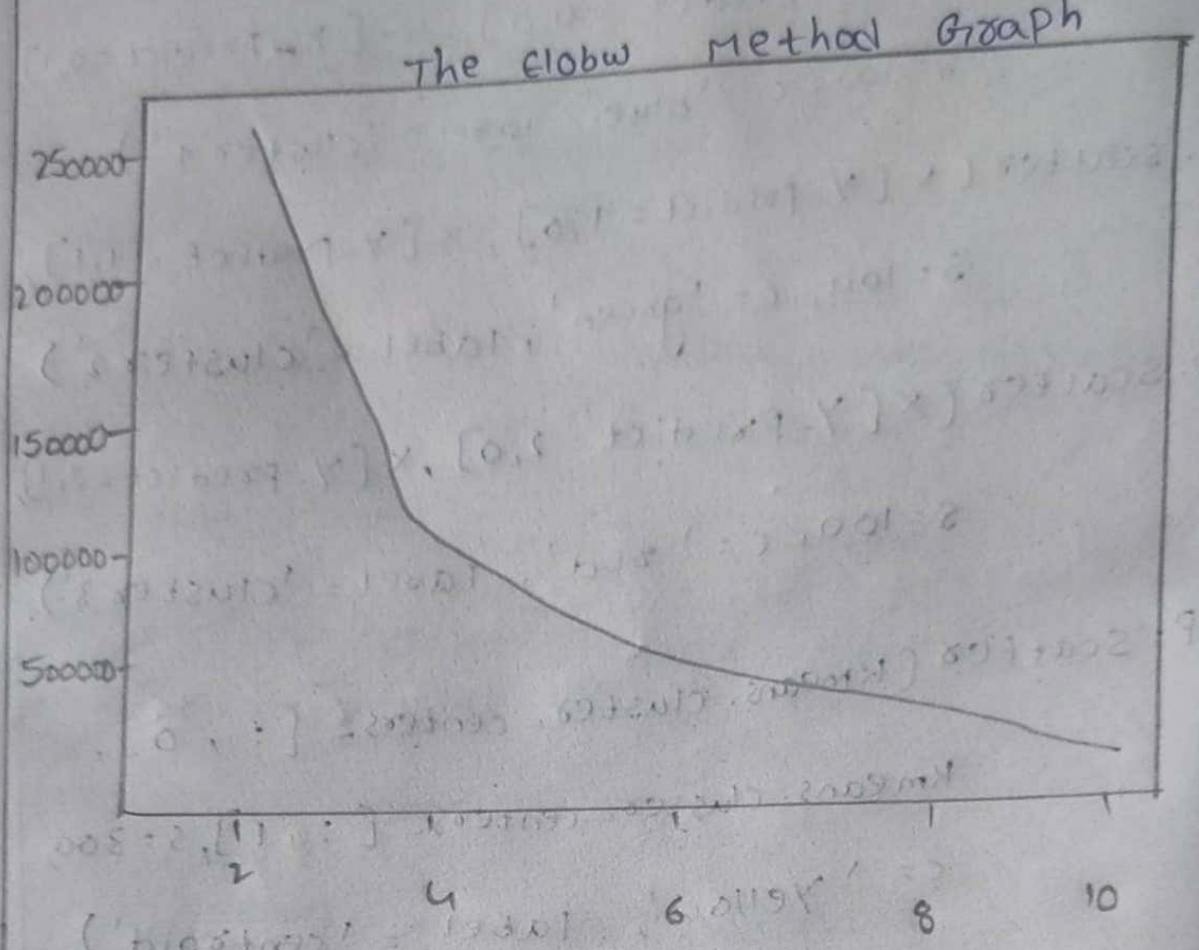
Step 6: If any reassignment occurs, then go to step-4, else go to FINISH.

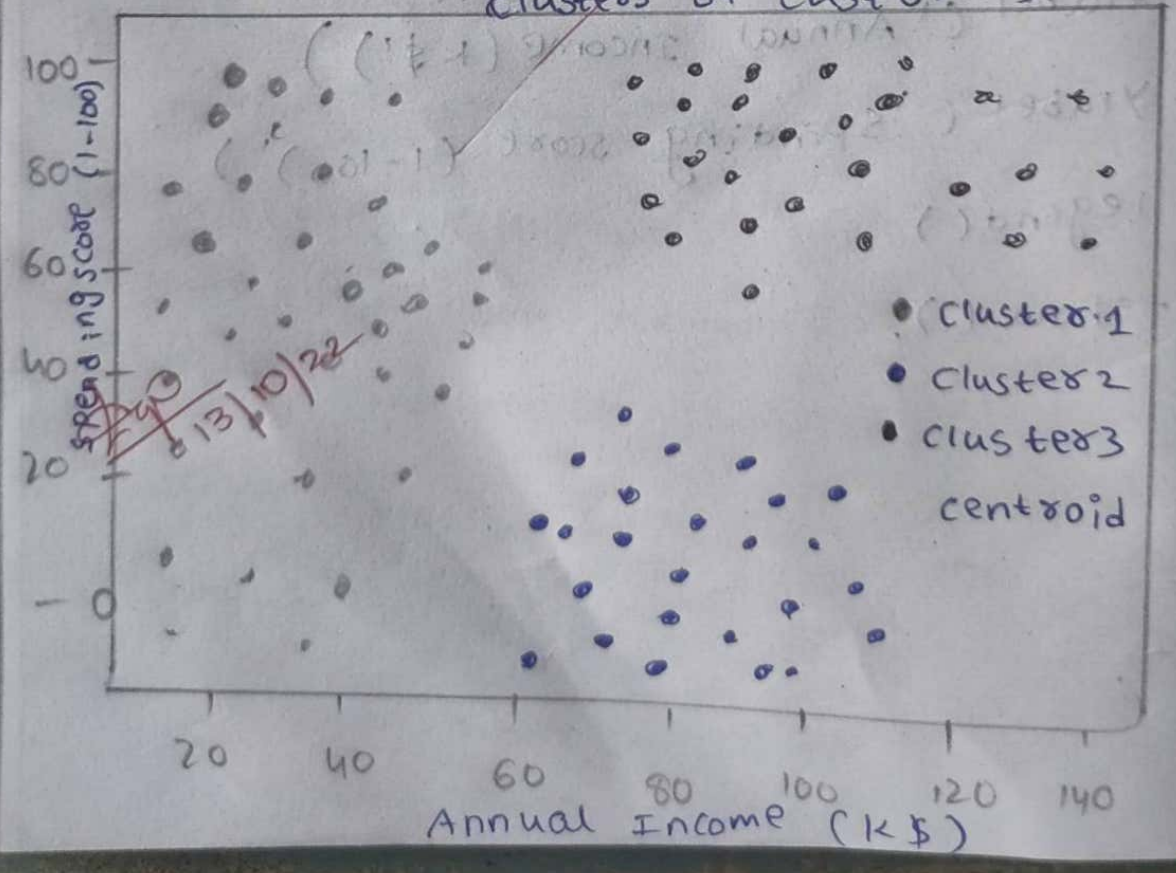
Step 7: The model is ready.

K-means Algorithm using python programming:

```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
dataset = pd.read_csv('/content/sample_data/Mall_
                       customers.csv')
X = dataset.iloc[:, [3,4]].values
from sklearn.cluster import KMeans
wcss_list = []
for i in range(1,11):
    Kmeans = KMeans(n_clusters = i, init =
                    'k-means++', random_state=42)
    Kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11), wcss_list)
mtp.title('The Elobw Method Graph')
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()
kmeans = KMeans mode(n_clusters=5, init='k-means++',
         random_state=42)
Y_predict = kmeans.fit_predict(x)
mtp.scatter(x[Y_predict=0,0], x[Y_predict=0,1],
            s=100, c='blue', label='cluster 1')
mtp.scatter(x[Y_predict=1,0], x[Y_predict=1,1],
            s=100, c='green', label='cluster 2')
mtp.scatter(x[Y_predict=2,0], x[Y_predict=2,1],
            s=100, c='red', label='cluster 3')
mtp.scatter(kmeans.cluster_centers_[:,0],
            kmeans.cluster_centers_[:,1], s=300,
            c='yellow', label='centroid')
mtp.title('clusters of customers')
mtp.xlabel('Annual Income(k$)')
mtp.ylabel('Spending score (1-100)')
mtp.legend()
mtp.show().
```

Output


The Elobw Method Graph
250000
200000
150000
100000
50000
2
4
6
8
10
Number of clusters(K)


Clusters of customers
100
80
60
40
20
0
Spending Score (1-100)
cluster 1
cluster 2
cluster 3
centroid
20
40
60
80
100
120
140
Annual Income (k$)