5-1-2022:Ds\_Ml lab: outcome3(c03)

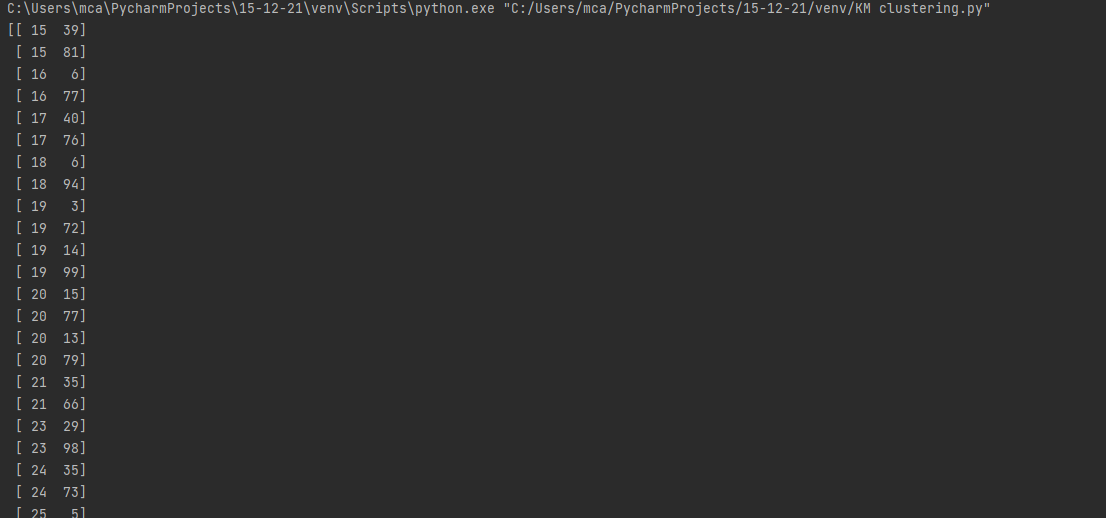
AIM:

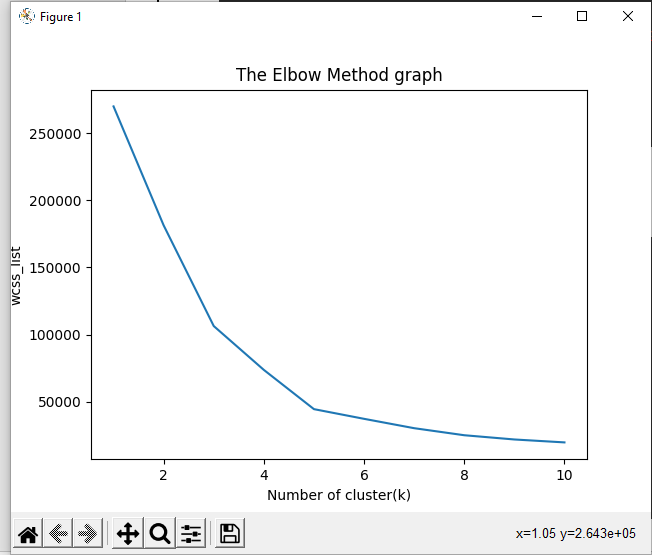
Program to implement k-means clustering technique using any standard dataset available in the public in the public domain.

Program:

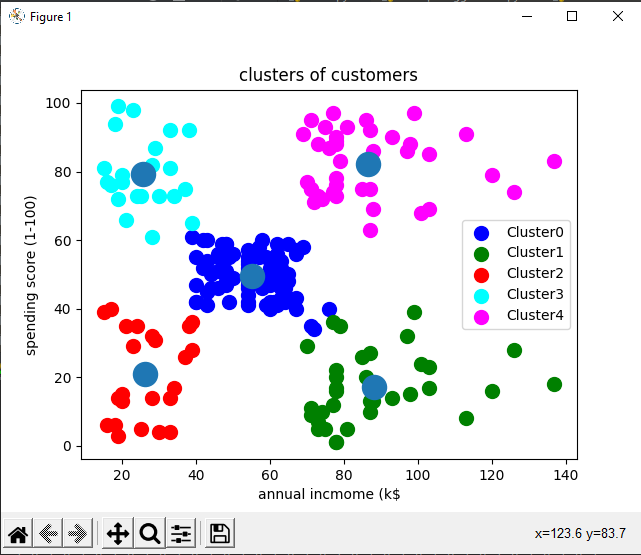
import matplotlib.pyplot as mtp  
import pandas as pd  
#importing the datasets  
datasets = pd.read\_csv('Mall\_Customers.csv')  
x = datasets.iloc[:,[3, 4]].values  
print(x)  
# finding optimal nu of clusters using the elbow method  
from sklearn.cluster import KMeans  
wcss\_list = [] #initilizing the list for the values for wcss  
  
#using for loop for itreation from 1 to 10  
  
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 Elbow Method graph')  
mtp.xlabel('Number of cluster(k)')  
mtp.ylabel('wcss\_list')  
mtp.show()

output:





import numpy as np  
import matplotlib.pyplot as mtp  
import pandas as pd  
dataset=pd.read\_csv('Mall\_Customers.csv')  
x=dataset.iloc[:,[3,4]].values  
print(x)  
from sklearn.cluster import KMeans  
wcss\_list = []  
for i in range(1, 11):  
 kmeans = KMeans(n\_clusters=i, init='k-means++')  
 kmeans.fit(x)  
 wcss\_list.append(kmeans.inertia\_)  
mtp.plot(range(1,11), wcss\_list)  
mtp.title('The elbow method Graph')  
mtp.xlabel('Number of clusters (k)')  
mtp.ylabel('wcss\_list')  
mtp.show()  
kmeans = KMeans(n\_clusters=5,init='k-means++',random\_state=42)  
y\_predict=kmeans.fit\_predict(x)  
print(y\_predict)  
mtp.scatter(x[y\_predict == 0,0], x[y\_predict ==0,1], s=100, c='blue', label='Cluster0')  
mtp.scatter(x[y\_predict == 1,0], x[y\_predict ==1,1], s=100, c='green', label= 'Cluster1')  
mtp.scatter(x[y\_predict == 2,0], x[y\_predict ==2,1], s=100, c='red', label= 'Cluster2')  
mtp.scatter(x[y\_predict == 3,0], x[y\_predict ==3,1], s=100, c='cyan', label= 'Cluster3')  
mtp.scatter(x[y\_predict == 4,0], x[y\_predict ==4,1], s=100, c='magenta', label= 'Cluster4')  
mtp.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1], s = 300,)  
mtp.title('clusters of customers')  
mtp.xlabel('annual incmome (k$')  
mtp.ylabel('spending score (1-100)')  
mtp.legend()  
mtp.show()



2) Program to implement k-means clustering technique using any standard dataset available in the public in the public domain.

mport matplotlib.pyplot as mtp

import pandas as pd

dataset=pd.read\_csv('country.csv')

x=dataset.iloc[:,[1,2]].values

print(x)

from sklearn.cluster import KMeans

wcss\_list = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, init='k-means++')

kmeans.fit(x)

wcss\_list.append(kmeans.inertia\_)

mtp.plot(range(1,11), wcss\_list)

mtp.title('The elbow method Graph')

mtp.xlabel('Number of clusters (k)')

mtp.ylabel('wcss\_list')

mtp.show()

kmeans = KMeans(n\_clusters=3,init='k-means++',random\_state=42)

y\_predict=kmeans.fit\_predict(x)

print(y\_predict)

mtp.scatter(x[y\_predict == 0,0], x[y\_predict ==0,1], s=100, c='blue', label='Cluster0')

mtp.scatter(x[y\_predict == 1,0], x[y\_predict ==1,1], s=100, c='green', label= 'Cluster1')

mtp.scatter(x[y\_predict == 2,0], x[y\_predict ==2,1], s=100, c='red', label= 'Cluster2')

mtp.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1], s = 300,)

mtp.title('clusters of customers')

mtp.xlabel('latitude')

mtp.ylabel('longitude')

mtp.legend()

mtp.show()

