

Implementation-of-K-Means-Clustering-for-Customer-Segmentation

AIM:

To write a program to implement the K Means Clustering for Customer Segmentation.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Jupyter notebook

Algorithm

1. Import pandas and matplotlib.pyplot
2. Read the dataset and transform it
3. Import KMeans and fit the data in the model
4. Plot the Cluster graph

Program:

Program to implement the K Means Clustering for Customer Segmentation.

Developed by: Iyalarasu C

RegisterNumber: 212223040069



```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv(r"C:\Users\admin\Downloads\Mall_Customers.csv")
data.head()
data.info()
data.isnull().sum()

from sklearn.cluster import KMeans
wcss = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i, init = "k-means++")
    kmeans.fit(data.iloc[:,3:])
    wcss.append(kmeans.inertia_)

plt.plot(range(1,11),wcss)
plt.xlabel("No. of clusters")
plt.ylabel("wcss")
plt.title("Elbow method")

km = KMeans(n_clusters=5)
km.fit(data.iloc[:,3:])

y_pred = km.predict(data.iloc[:,3:])
y_pred

data["cluster"] = y_pred

df0 = data[data["cluster"]==0]
df1 = data[data["cluster"]==1]
df2 = data[data["cluster"]==2]
df3 = data[data["cluster"]==3]
df4 = data[data["cluster"]==4]
plt.scatter(df0["Annual Income (k$)"],df0["Spending Score (1-100)"],c="red",label="cluster 0")
plt.scatter(df1["Annual Income (k$)"],df1["Spending Score (1-100)"],c="black",label="cluster 1")
```

```
plt.scatter(df2["Annual Income (k$)",df2["Spending Score (1-100)"],c="blue",label="cluster 2")
plt.scatter(df3["Annual Income (k$)",df3["Spending Score (1-100)"],c="green",label="cluster 3")
plt.scatter(df4["Annual Income (k$)",df4["Spending Score (1-100)"],c="magenta",label="cluster 4")

plt.legend()
plt.title("customer segmentation")
```

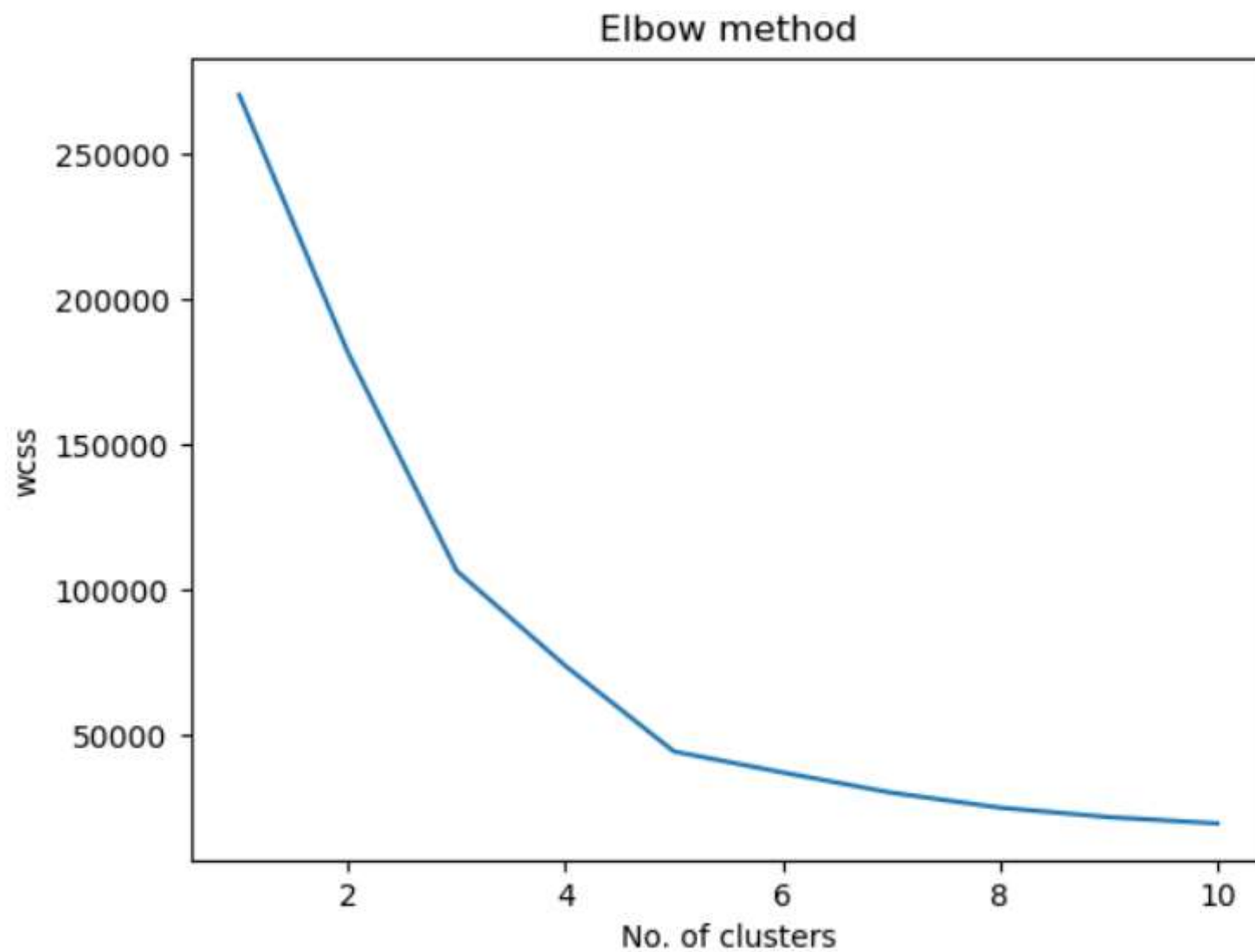
Output:

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   CustomerID                           200 non-null   int64
1   Gender                               200 non-null   object
2   Age                                   200 non-null   int64
3   Annual Income (k$)                   200 non-null   int64
4   Spending Score (1-100)                200 non-null   int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

CustomerID	0
Gender	0
Age	0
Annual Income (k\$)	0
Spending Score (1-100)	0
dtype: int64	

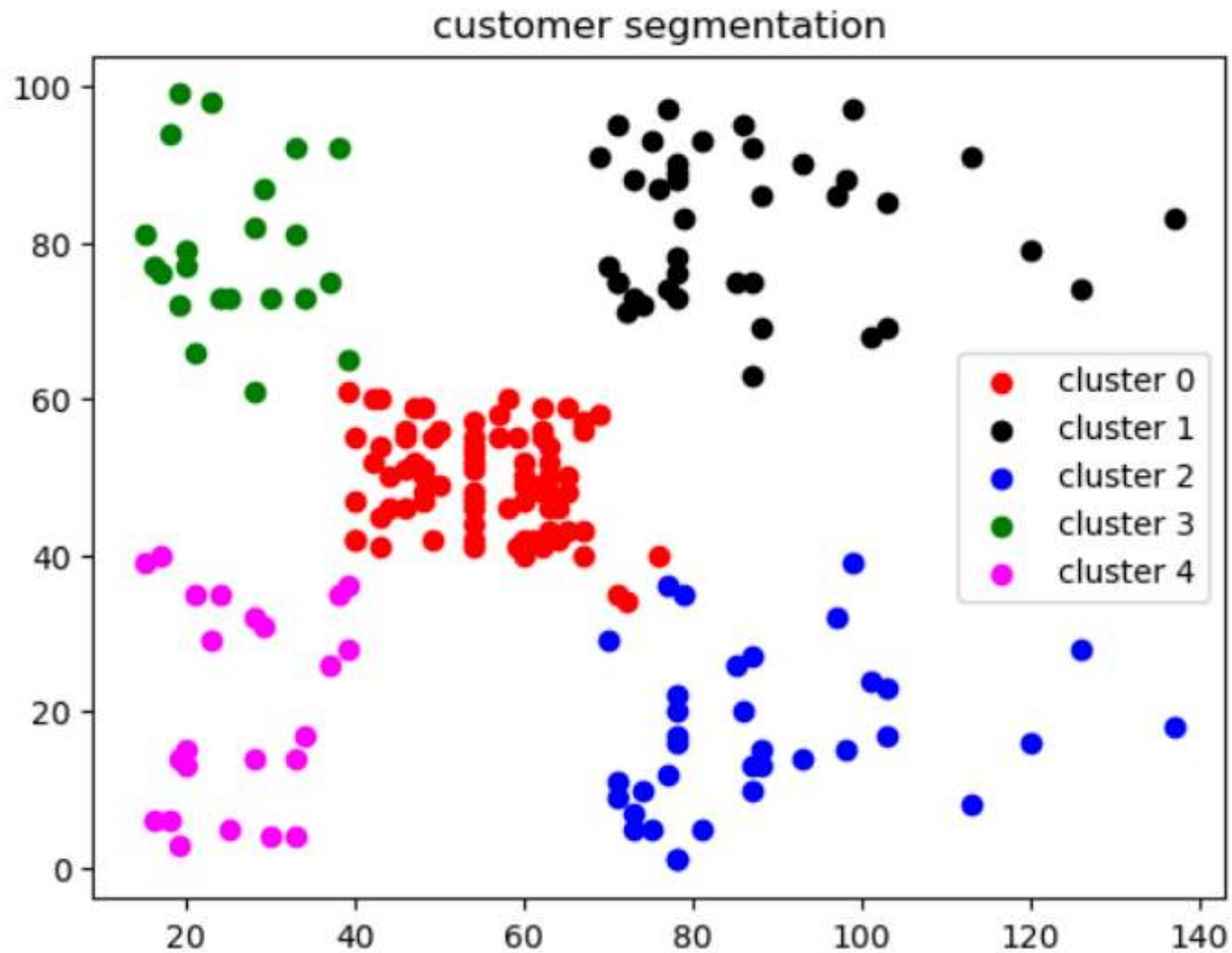
```
Text(0.5, 1.0, 'Elbow method')
```



▼ KMeans
KMeans(n_clusters=5)

```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
      4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 0,
      4, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1, 2, 1, 2, 1,
      0, 1, 2, 1, 2, 1, 2, 1, 2, 1, 0, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
      2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
      2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
      2, 1])
```

```
Text(0.5, 1.0, 'customer segmentation')
```



Result:

Thus the program to implement the K Means Clustering for Customer Segmentation is written and verified using python programming.