Chapter 11 - Exercise 3: Shopping Data

Cho dữ liệu shopping_data.csv, thực hiện việc phân nhóm dữ liệu theo KMeans Clustering theo 2 thuộc tính là Annual Income (k$)và Spending Score (1-100)

1. Đọc dữ liệu, chuẩn hóa dữ liệu (nếu cần)
2. Trực quan hóa dữ liệu
3. Áp dụng Elbow tìm k
4. Áp dụng thuật toán K-Means để giải bài toán phân cụm theo K
5. Trực quan hóa kết quả, nhận xét

In [1]:

```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:

```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter11_Kmeans/'
```

In [3]:

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn import metrics
from scipy.spatial.distance import cdist
```

In [4]:

```
df = pd.read_csv("shopping_data.csv")
df.head()
```

Out[4]:

	CustomerID	Genre	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

In [5]:

```
df_new = df.iloc[:, 3:5]
df_new.head()
```

Out[5]:

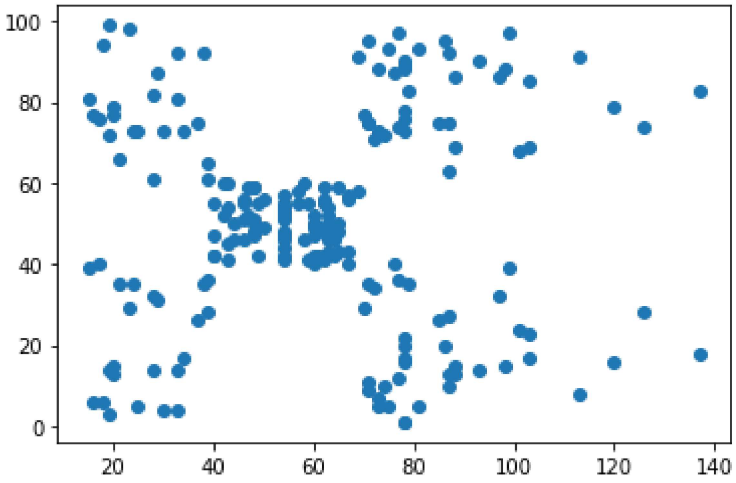
	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

In [6]:

```
plt.scatter(df_new['Annual Income (k$)'], df_new['Spending Score (1-100)'])
```

Out[6]:

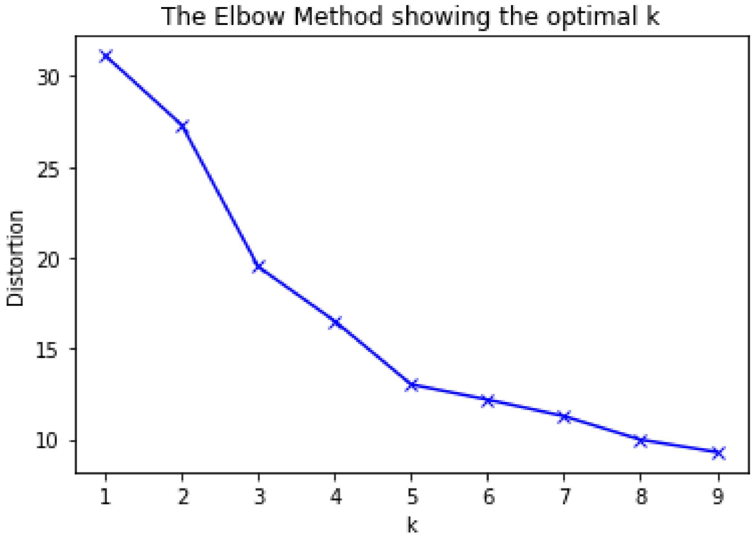
```
<matplotlib.collections.PathCollection at 0x1974a82b630>
```



In [7]:

```
# k means determine k
distortions = [] # WSSE
K = range(1,10) #
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df_new) # cluster center (x1, x2) ; (x1, x2, x3)
    distortions.append(sum(np.min(cdist(df_new, kmeanModel.cluster_centers_, 'euclidea
n'), axis=1)) / df.shape[0])

# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [8]:

```
# => Select k = 5
kmeans = KMeans(n_clusters=5)
kmeans.fit(df_new)

centroids = kmeans.cluster_centers_
labels = kmeans.labels_ # 0,1,2,3,4

print(centroids)
print(labels)
```

```
[[88.2        17.11428571]
 [55.2962963  49.51851852]
 [86.53846154 82.12820513]
 [25.72727273 79.36363636]
 [26.30434783 20.91304348]]
[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 1 4 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 2 0 2 1 2 0 2 0 2 1 2 0 2 0 2 0 2 0 2 1 2 0 2 0 2
 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0
 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]
```

In [9]:

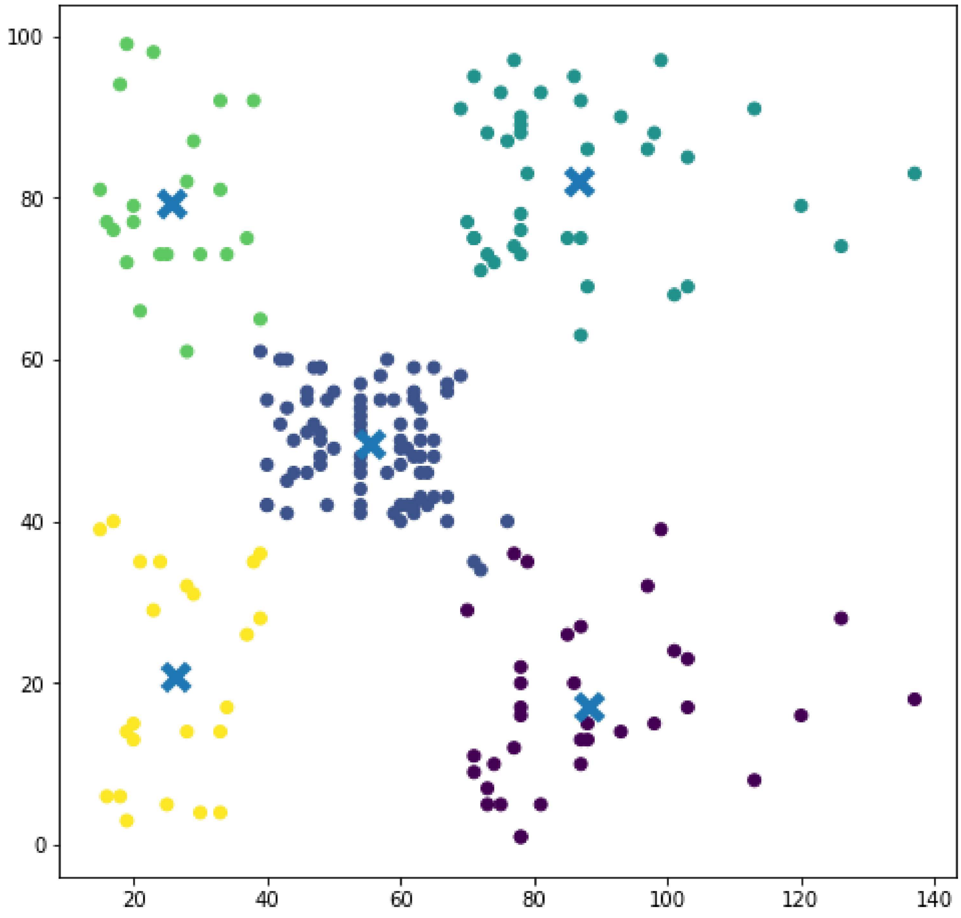
```
df_new['Group'] = pd.Series(labels)
df_new.head()
```

Out[9]:

	Annual Income (k$)	Spending Score (1-100)	Group
0	15	39	4
1	15	81	3
2	16	6	4
3	16	77	3
4	17	40	4

In [10]:

```
plt.figure(figsize=(8,8))
plt.scatter(centroids[:, 0],centroids[:, 1],
            marker = "x", s=150, linewidths = 5, zorder = 10)
plt.scatter(df_new['Annual Income (k$)'],
            df_new['Spending Score (1-100)'], c = df_new.Group)
# dat x, dat x
plt.show()
```



Giải thích cụ thể từng cụm.

Nếu bây giờ phân cụm theo:

- Annual Imcome + Spending Score + Age => ? cụm => Giải thích
- Annual Imcome + Spending Score + Gender => ? cụm => Giải thích
- Annual Imcome + Spending Score + Age + Gender => ? cụm => Giải thích