Course: Laboratory Practice-III (Machine Learning)

Name: Anuj Mahendra Mutha

Class: BE-4

Batch : R4

Roll No. : 41443

Assignment Number : Group B - 06

Title: Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method.

Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

In [ ]:

```
dataset = pd.read_csv('/content/sales_data_sample.csv',sep=",", encoding='
```

In [ ]:

```
dataset.head()
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDA
0	10107	30	95.70	2	2871.00	2/24/2003 0:00
1	10121	34	81.35	5	2765.90	5/7/2003 0:00
2	10134	41	94.74	2	3884.34	7/1/2003 0:00
3	10145	45	83.26	6	3746.70	8/25/2003 0:00
4	10159	49	100.00	14	5205.27	10/10/200 0:00

5 rows × 25 columns

```
In [ ]: dataset.tail()
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDER
2818	10350	20	100.00	15	2244.40	12/2/20 0:00
2819	10373	29	100.00	1	3978.51	1/31/20 0:00
2820	10386	43	100.00	4	5417.57	3/1/200 0:00
2821	10397	34	62.24	1	2116.16	3/28/20 0:00
2822	10414	47	65.52	9	3079.44	5/6/200 0:00

```
5 rows × 25 columns
```

In [ ]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   ORDERNUMBER       2823 non-null   int64  
 1   QUANTITYORDERED   2823 non-null   int64  
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64  
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object 
 6   STATUS            2823 non-null   object 
 7   QTR_ID            2823 non-null   int64  
 8   MONTH_ID          2823 non-null   int64  
 9   YEAR_ID           2823 non-null   int64  
 10  PRODUCTLINE       2823 non-null   object 
 11  MSRP              2823 non-null   int64  
 12  PRODUCTCODE       2823 non-null   object 
 13  CUSTOMERNAME      2823 non-null   object 
 14  PHONE             2823 non-null   object 
 15  ADDRESSLINE1      2823 non-null   object 
 16  ADDRESSLINE2      302 non-null    object 
 17  CITY              2823 non-null   object 
 18  STATE             1337 non-null   object 
 19  POSTALCODE        2747 non-null   object 
 20  COUNTRY           2823 non-null   object 
 21  TERRITORY         1749 non-null   object 
 22  CONTACTLASTNAME   2823 non-null   object 
 23  CONTACTFIRSTNAME  2823 non-null   object 
 24  DEALSIZE          2823 non-null   object 
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

In [ ]:

```
dataset.shape
```

```
(2823, 25)
```

In [ ]:

```
dataset.isnull().sum()
```

```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

In [ ]:

```
X = dataset.iloc[:, [1, 2]].values
```

In [ ]:

```
X
```

```
array([[ 30.  ,  95.7 ],
       [ 34.  ,  81.35],
       [ 41.  ,  94.74],
       ...,
       [ 43.  , 100.  ],
       [ 34.  ,  62.24],
       [ 47.  ,  65.52]])
```

In [ ]:

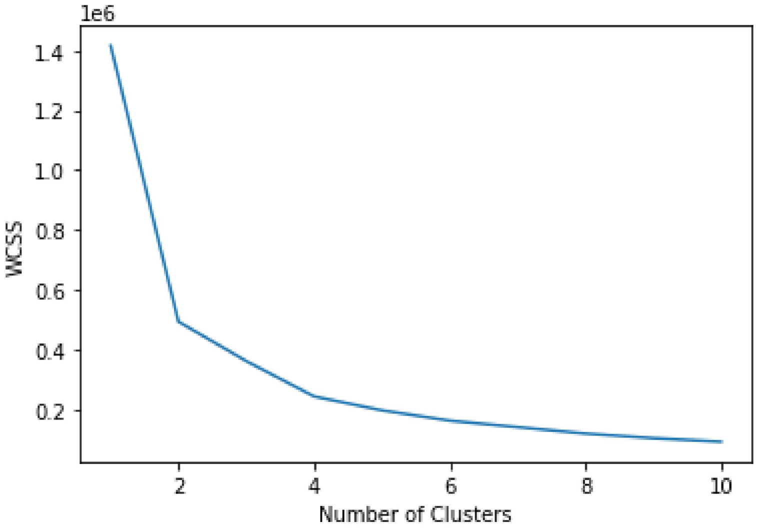
```
from sklearn.cluster import KMeans
```

In [ ]:

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

In [ ]:

```
plt.plot(range(1,11), wcss)
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.show()
```



In [ ]:

```
kmeans = KMeans(n_clusters = 5, init = "k-means++", random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

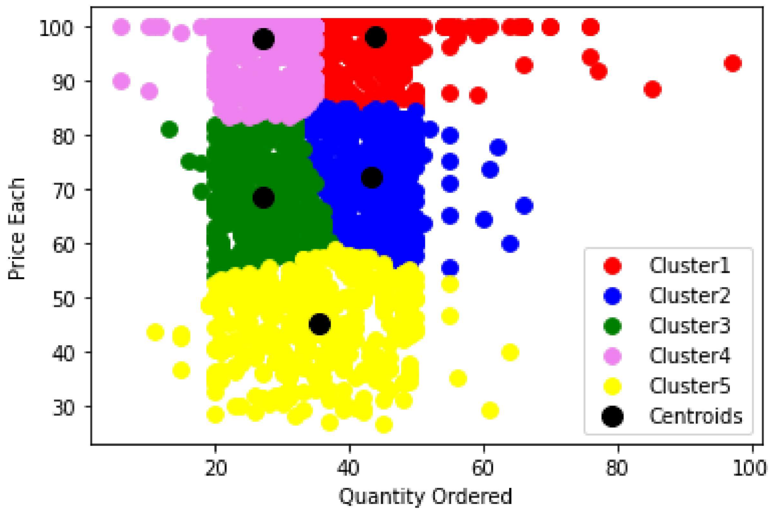
In [ ]:

```
y_kmeans
```

```
array([3, 1, 0, ..., 0, 2, 1], dtype=int32)
```

```
In [ ]: plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 60, c = 'red', l
        plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 60, c = 'blue',
        plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 60, c = 'green',
        plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 60, c = 'violet'
        plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 60, c = 'yellow'
        plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
        plt.xlabel('Quantity Ordered')
        plt.ylabel('Price Each')
        plt.legend()

        plt.show()
```


Price Each
Quantity Ordered
Cluster1
Cluster2
Cluster3
Cluster4
Cluster5
Centroids