

Course: Laboratory Practice III

Course Code: 410246

Name: Ahire Kalpesh Bapurao

Class: BE - A

Roll No. :12

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>

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

```
dataset = pd.read_csv('/content/sales_data_sample.csv', sep=",", encoding='Latin-1')
```

```
dataset.head()
```

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

5 rows × 25 columns



```
dataset.tail()
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE
2818	10350	20	100.00	15	2244.40	12/2/2004 0:00
2819	10373	29	100.00	1	3978.51	1/31/2005 0:00
2820	10386	43	100.00	4	5417.57	3/1/2005 0:00
2821	10397	34	62.24	1	2116.16	3/28/2005 0:00
2822	10414	47	65.52	9	3079.44	5/6/2005 0:00

5 rows x 25 columns

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
```

dataset.shape

(2823, 25)

```
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      0
CITY              0
STATE             0
POSTALCODE        0
COUNTRY           0
TERRITORY         0
CONTACTLASTNAME   0
CONTACTFIRSTNAME  0
DEALSIZE          0
dtype: int64
```

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

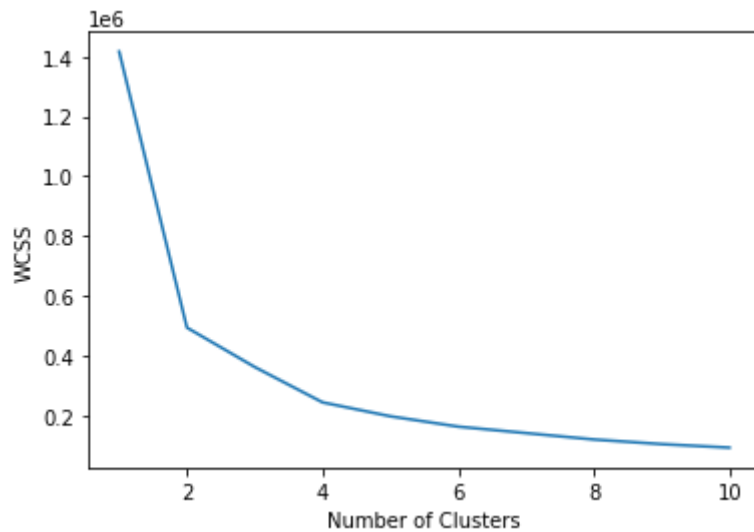
```
X
```

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

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

```
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_)
```

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



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

```
y_kmeans
```

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

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 60, c = 'red', label = 'Cluster1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 60, c = 'blue', label = 'Cluster2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 60, c = 'green', label = 'Cluster3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 60, c = 'violet', label = 'Cluster4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 60, c = 'yellow', label = 'Cluster5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'black', label = 'Centroids')
plt.xlabel('Quantity Ordered')
plt.ylabel('Price Each')
plt.legend()

plt.show()
```

