```
Student Name : Abhang Rushikesh
Roll No : BEA-30
```

Group B Machine Learning

Assignment 4 K Means Clustring

K Means Clustring

In [11]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
```

In [4]:
```
df=pd.read_csv('sales_data_sample.csv',encoding='latin-1')
```

In [5]:
```
df.sample(5)
```

Out[5]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDA
1045	10301	23	100.00	9	4011.66	10/5/20 0
2498	10308	21	100.00	12	2224.95	10/15/20 0
2275	10413	24	49.71	6	1193.04	5/5/2005 0
2485	10133	24	77.64	8	1863.36	6/27/20 0
428	10194	21	93.34	10	1960.14	11/25/20 0

5 rows × 25 columns

```
In
[6]:
df.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
  10PRODUCTLINE       2823 non-null   object
  11MSRP              2823 non-null   int64
  12PRODUCTCODE       2823 non-null   object
  13CUSTOMERNAME      2823 non-null   object
  14PHONE             2823 non-null   object
  15ADDRESSLINE1      2823 non-null   object
  16ADDRESSLINE2      302 non-null    object
  17CITY              2823 non-null   object
  18STATE             1337 non-null   object
  19POSTALCODE        2747 non-null   object
  20COUNTRY           2823 non-null   object
  21TERRITORY         1749 non-null   object
  22CONTACTLASTNAME   2823 non-null   object
  23CONTACTFIRSTNAME  2823 non-null   object
  24DEALSIZE          2823 non-null   object  dtypes: float64(2), int64(7),
 object(16) memory usage: 551.5+ KB
```

```
In [7]: df.isnull().sum()
```

```
Out[7]: 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 [8]: df.shape
```

```
Out[8]: (2823, 25)
```

```
In [9]: df.duplicated().sum()
```

```
Out[9]: 0
```

In

```
[10]: df.corr()
```

Out[10]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	S
ORDERNUMBER	1.000000	0.065543	-0.002935	-0.055550	0.0
QUANTITYORDERED	0.065543	1.000000	0.005564	-0.018397	0.5
PRICEEACH	-0.002935	0.005564	1.000000	-0.020965	0.6
ORDERLINENUMBER	-0.055550	-0.018397	-0.020965	1.000000	-0.0
SALES	0.039919	0.551426	0.657841	-0.058400	1.0
QTR_ID	-0.051383	-0.035323	0.008712	0.040716	-0.0
MONTH_ID	-0.039723	-0.039048	0.005152	0.034016	-0.0
YEAR_ID	0.904596	0.069535	-0.005938	-0.057367	0.0
MSRP	-0.010280	0.017881	0.670625	-0.021067	0.6

In [14]:

```
df['CITY'].value_counts()
```

Out[14]:

```
Madrid          304
San Rafael      180
NYC             152
Singapore        79
Paris            70
 ...
Graz             15
Los Angeles      14
```

In [16]:

```
df['STATE'].value_counts().count()
```

Out[16]:

```
16
```

In [18]:

```
df1=df.select_dtypes(exclude='object')
```

In [19]:

```
df1.shape
```

Out[19]:

```
(2823, 9)
Munich           14
Burbank          13
Charleroi         8
```

In

```
Name: CITY, Length: 73, dtype: int64
```

[20]:

```
df1.sample(5)
```

Out[20]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	QTR_ID
1307	10315	36	100.00	7	3602.16	4
2222	10104	35	47.62	11	1666.70	1
2500	10328	27	100.00	8	2762.10	4
1762	10328	48	58.92	1	2828.16	4
2242	10335	40	60.60	3	2424.00	4

In [21]:

```
std_scalar= StandardScaler()
df_scaled = std_scalar.fit_transform(df1)
```

In [23]:

```
df_scaled
```

Out[23]:

```
array([[-1.64794709, -0.52289086,  0.5969775 , ..., -1.39290889,
         -1.16517009, -0.14224584],
        [-1.4958875 , -0.11220131, -0.11445035, ..., -0.57233673,        -
 1.16517009, -0.14224584],
        [-1.35468931,  0.60650538,  0.54938372, ..., -0.02528862,
         -1.16517009, -0.14224584],
        ...,
        [ 1.38238338,  0.81185016,  0.81015797, ..., -1.11938483,
 1.69382614, -1.16263387],
        [ 1.50185877, -0.11220131, -1.06186404, ..., -1.11938483,
 1.69382614, -1.16263387],
        [ 1.68650256,  1.2225397 , -0.89925195, ..., -0.57233673,
```

In [24]:

```
df2=pd.DataFrame(df_scaled,columns=df1.columns)
```

In [26]:

```
df2.sample(5)
```

Out[26]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	QTR_I
1136	1.197740	1.017195	-0.524451	-0.583696	-0.144058	-1.42703
386	0.708977	-0.933580	-1.253231	0.126347	-1.105602	1.06535
1201	-0.073044	-0.420218	-1.527886	-0.347015	-1.040352	-0.59624
86	-0.855065	0.606505	0.810158	1.073072	2.899212	1.06535
1101	-0.225104	0.606505	-0.519989	-0.110334	-0.300791	-0.59624

In

```
        1.69382614, -1.16263387]])
```

[25]:
```
df2.corr()
```

Out[25]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	S
ORDERNUMBER	1.000000	0.065543	-0.002935	-0.055550	0.0
QUANTITYORDERED	0.065543	1.000000	0.005564	-0.018397	0.5
PRICEEACH	-0.002935	0.005564	1.000000	-0.020965	0.6
ORDERLINENUMBER	-0.055550	-0.018397	-0.020965	1.000000	-0.0
SALES	0.039919	0.551426	0.657841	-0.058400	1.0
QTR_ID	-0.051383	-0.035323	0.008712	0.040716	-0.0
MONTH_ID	-0.039723	-0.039048	0.005152	0.034016	-0.0
YEAR_ID	0.904596	0.069535	-0.005938	-0.057367	0.0
MSRP	-0.010280	0.017881	0.670625	-0.021067	0.6

In [27]:
```
from sklearn.cluster import KMeans
```

In [93]:
```
def WCSS(dataframe):
  wcss_list = []

  for k in range(1,8):

    kmeans_model = KMeans(n_clusters=k)
    kmeans_model.fit(dataframe)
    wcss_value= kmeans_model.inertia_
    wcss_list.append(wcss_value)
    print(f'for k == {k}, wcss is { wcss_value }')
  print("Cluster Centers:",kmeans_model.cluster_centers_)
  print("Feature Names:",kmeans_model.feature_names_in_)
  return wcss_list
```

[98]:
```
list1=WCSS(df2)
```

```
for k == 1, wcss is 25407.000000000022
for k == 2, wcss is 20090.88701217339 for
k == 3, wcss is 16909.327212616885 for k
== 4, wcss is 14818.002265126062 for k ==
5, wcss is 13539.084829579537 for k == 6,
```

In

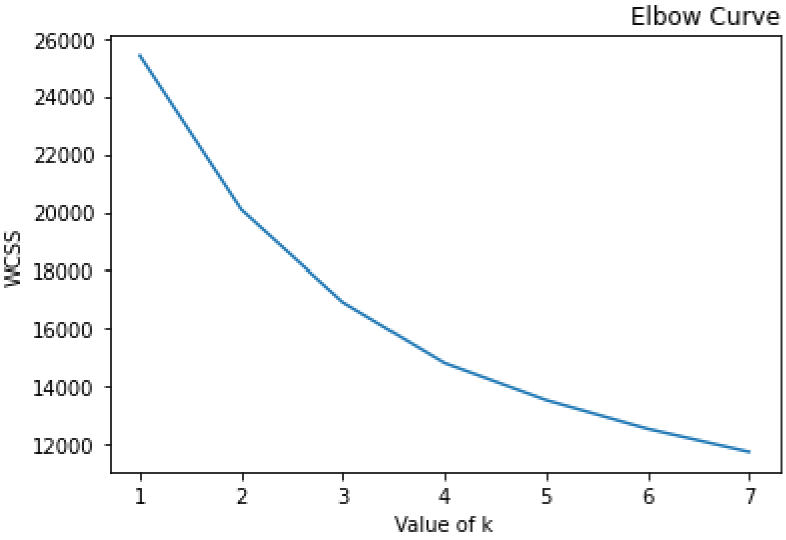
```
wcss is 12546.487375729277 for k == 7,
wcss is 11750.52826522585
Cluster Centers: [[-1.65229723e-01 -1.62463528e-01 -1.30802245e+00  5.70264992e -
02
  -8.96111961e-01  7.80306529e-01  7.65527784e-01 -4.29486958e-01  -
 9.69052753e-01]
 [-8.40236519e-01 -1.78149572e-01  6.28741655e-01 -6.37777720e-04
   1.18321519e-01 -1.02733575e+00 -1.00501458e+00 -3.62202633e-01
 3.66600355e-01]
 [ 1.43907888e+00 -3.35333519e-02 -9.31828018e-02 -9.11582949e-02
  -1.56612779e-01 -1.15395289e+00 -1.14407798e+00  1.69382614e+00  -
 1.10099474e-01]
 [-7.53639205e-01 -1.28891156e-02 -1.17312351e+00  5.85510281e-03
  -7.95246500e-01 -1.00106684e+00 -9.85109390e-01 -2.76282170e-01  -
 9.75117770e-01]
 [ 6.30810799e-01 -1.96967940e-01  4.70065354e-01  3.94032376e-02
   1.16368961e-02  8.31061956e-01  8.28268814e-01  2.51067374e-01
 1.89649224e-01]
 [ 2.02392884e-01  1.11986731e+00  7.97229433e-01 -2.53219181e-01
   2.03435595e+00 -1.77623653e-02 -2.93408295e-02  2.27266963e-01
 1.27265534e+00]
 [-9.17895108e-01 -6.89458182e-02  6.24701521e-01  1.34017581e-01
   2.59650437e-01  8.46111679e-01  8.32639833e-01 -1.16517009e+00
   4.34142038e-01]]
Feature Names: ['ORDERNUMBER' 'QUANTITYORDERED' 'PRICEEACH' 'ORDERLINENUMBER'
'SALES'
 'QTR_ID' 'MONTH_ID' 'YEAR_ID' 'MSRP']
```

In [101]:

```
def ElbowCurve(wcss_list):
  k=[1,2,3,4,5,6,7]
  plt.plot(k,wcss_list)
  plt.xlabel('Value of k')
  plt.ylabel('WCSS')
  plt.title('Elbow Curve',loc='right')
```

In [102]:

```
ElbowCurve(list1)
```



wcss (within cluster sum of square) >> sum of square of distances of points from the respective centroids Elbow Graph >> elbow shaped graph that helps us decide the optimal value of k. Silhouette score >> calculated from silhouette co-efficient. Whichever value of k has highest silhouette score that would be decided for k value.

In [39]:

```
from sklearn.metrics import silhouette_score
```

In [103]:

```
def SilhoutteScore(dataframe):
  silhouette_score_list = []

  for k in range(2,6):

      kmeans_model_new = KMeans(n_clusters=k)

      y_pred_new = kmeans_model.fit_predict(dataframe)

      silhouette_coefficient = silhouette_score(dataframe,y_pred_new)

      silhouette_score_list.append(silhouette_coefficient)

      print(f'for k == {k},& silhouette score is {silhouette_coefficient}')

  return silhouette_score_list
```

In [104]:

```
slist=SilhoutteScore(df2)
```

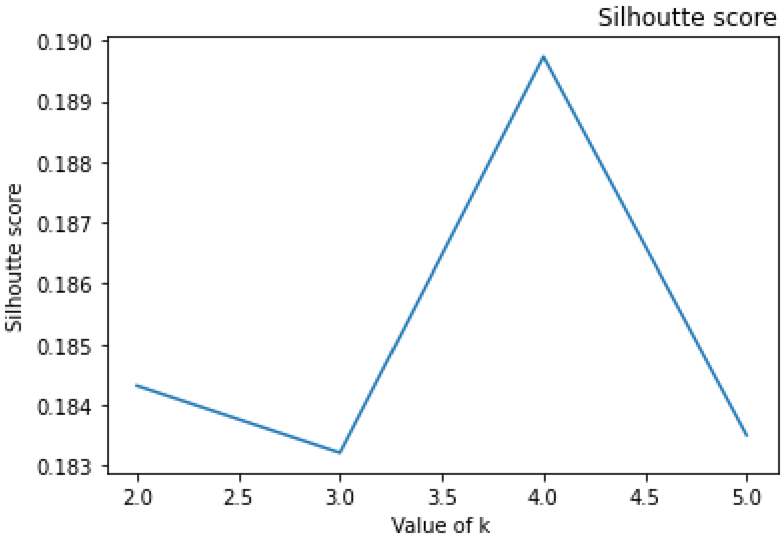
```
for k == 2,& silhouette score is 0.18431723406990635
for k == 3,& silhouette score is 0.18321172557165
for k == 4,& silhouette score is 0.18973662495542307
for k == 5,& silhouette score is 0.1835025449952827
```

In [105]:

```
def plotSilhoutte(silhouette_score_list):
  k=range(2,6)
  plt.plot(k,silhouette_score_list)
  plt.xlabel('Value of k')
  plt.ylabel('Silhoutte score')
  plt.title('Silhoutte score',loc='right')
```

In [106]:

```
plotSilhoutte(slist)
```



In [107]:

```
df3=df2[['QUANTITYORDERED','SALES']]
```

In [89]:
```
df3.shape
```

```
Out[89]: (2823, 2) Out[90]:
```

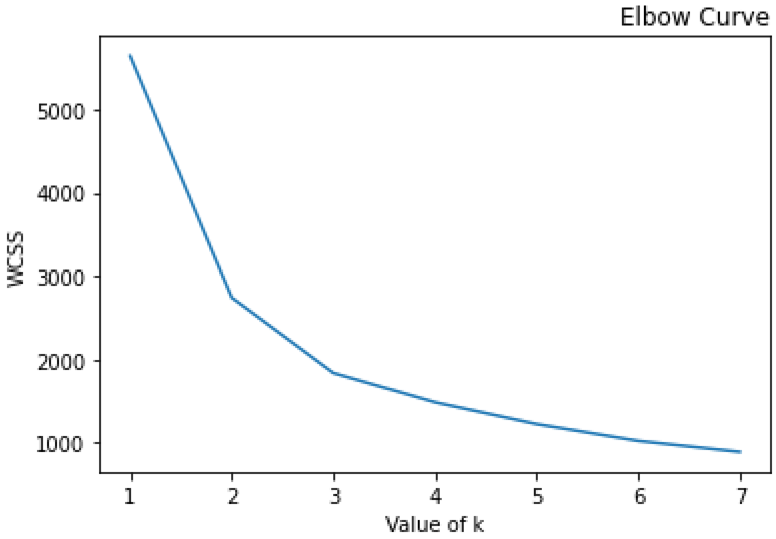
In [90]:
```
df3.sample(5)
```

	QUANTITYORDERED	SALES
2343	1.530557	0.369860
733	-0.420218	0.509868
2000	-0.728236	-0.923450
2611	0.195816	0.794814
2445	-1.446942	-1.122707

In [108]:
```
list2=WCSS(df3)
```

```
for k == 1, wcss is 5646.000000000013 for
k == 2, wcss is 2742.1326113529076 for k
== 3, wcss is 1836.773942782872 for k ==
4, wcss is 1484.9056875574397 for k == 5,
wcss is 1222.2810663248342 for k == 6,
wcss is 1020.4214771528574 for k == 7,
wcss is 888.3489682710209 Cluster Centers:
[[-0.9436959  -0.06492512]
 [ 1.06927822 -0.18543648]
 [-1.20680617 -0.98181626]
 [ 0.04988416  0.59361836]
 [-0.06830791 -0.67748743]
 [ 1.69369187  2.92676821]
 [ 0.99929307  1.35167433]]
Feature Names: ['QUANTITYORDERED' 'SALES']
```

In [109]:
```
ElbowCurve(list2)
```

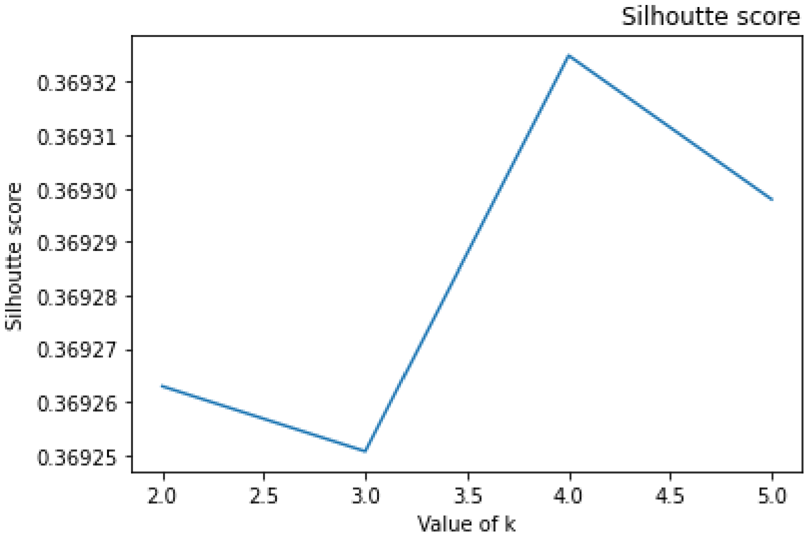


In [110]:
```
slist1=SilhoutteScore(df3)
```

```
for k == 2,& silhouette score is 0.36926295964297356
for k == 3,& silhouette score is 0.36925075040605526
for k == 4,& silhouette score is 0.3693249255482588
for k == 5,& silhouette score is 0.36929794842789970
```

In [111]:

```
plotSilhoutte(slist1)
```



In [ ]: