Import Libraries

In [12]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
```

In [2]:
```
# Upload the dataset
df= pd.read_csv("D:\\customer_data.csv")
df.head()
```

Out[2]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Rece
0	5524	1957	Graduation	Single	58138.0	0	0	04/09/12	
1	2174	1954	Graduation	Single	46344.0	1	1	08/03/14	
2	4141	1965	Graduation	Together	71613.0	0	0	21/08/13	
3	6182	1984	Graduation	Together	26646.0	1	0	10/02/14	
4	5324	1981	PhD	Married	58293.0	1	0	19/01/14	

5 rows × 29 columns

In [3]:
```
# Find the Shape
df.shape
```

Out[3]: (2240, 29)

In [4]:
```
# Description of the Dataset
df.describe()
```

Out[4]:

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntW
count	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.000000	2240.00
mean	5592.159821	1968.805804	52247.251354	0.444196	0.506250	49.109375	303.93
std	3246.662198	11.984069	25173.076661	0.538398	0.544538	28.962453	336.59
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.00
25%	2828.250000	1959.000000	35303.000000	0.000000	0.000000	24.000000	23.75
50%	5458.500000	1970.000000	51381.500000	0.000000	0.000000	49.000000	173.50
75%	8427.750000	1977.000000	68522.000000	1.000000	1.000000	74.000000	504.25
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	99.000000	1493.00

8 rows × 26 columns

In [21]:
```
# Find the Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

In [16]:
```
df.head()
```

Out[16]:

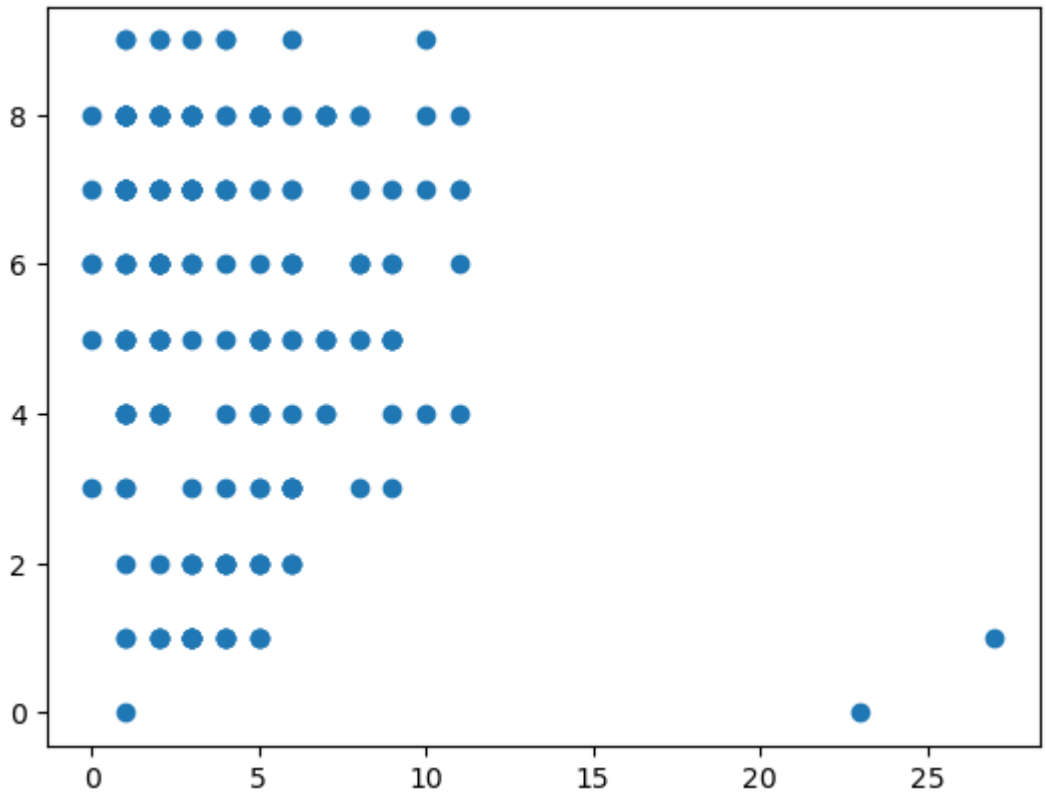
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Rece
0	5524	1957	Graduation	4	58138.0	0	0	04/09/12	
1	2174	1954	Graduation	4	46344.0	1	1	08/03/14	
2	4141	1965	Graduation	5	71613.0	0	0	21/08/13	
3	6182	1984	Graduation	5	26646.0	1	0	10/02/14	
4	5324	1981	PhD	3	58293.0	1	0	19/01/14	

5 rows × 29 columns

In [67]:
```
R= df.sample(frac= 0.1)
print(R)
```

```
         ID  Year_Birth   Education  Marital_Status   Income  Kidhome  \
1469   4406        1970  Graduation               5  67419.0        0
216    7264        1978    2n Cycle               4  52195.0        2
1515   2853        1980  Graduation               4  51766.0        1
266    1225        1963  Graduation               3  80124.0        0
1351   5080        1993  Graduation               4  70515.0        0
...     ...         ...         ...             ...      ...      ...
1806   6237        1966         PhD               4   7144.0        0
1028  10175        1958         PhD               2  32173.0        0
696    8315        1995  Graduation               4  34824.0        0
1179   5735        1991      Master               4  90638.0        0
2200   7620        1990       Basic               4  16185.0        1

      Teenhome Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  \
1469         1    16/01/13       29       846  ...                  5
216          1    12/05/14        2        12  ...                  8
1515         0    11/03/14       74        60  ...                  5
266          0    26/06/14       47       483  ...                  1
1351         0    21/10/13       12       420  ...                  2
...        ...         ...      ...       ...  ...                ...
1806         2    07/12/13       92        81  ...                  0
1028         1    01/08/13        0        18  ...                  4
696          0    26/03/14       65         4  ...                  6
1179         0    13/02/14       29      1156  ...                  1
2200         0    05/08/13       71         5  ...                  8

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  \
1469             0             0             0             0             0
216              0             0             0             0             0
1515             0             0             0             0             0
266              1             0             0             0             0
1351             0             0             0             0             0
...            ...           ...           ...           ...           ...
1806             0             0             0             0             0
1028             0             0             0             0             0
696              0             0             0             0             0
1179             0             0             1             0             0
2200             0             0             0             0             0

      Complain  Z_CostContact  Z_Revenue  Response
1469         0              3         11         0
216          0              3         11         0
1515         0              3         11         0
266          0              3         11         0
1351         0              3         11         1
...        ...            ...        ...       ...
1806         0              3         11         0
1028         0              3         11         0
696          0              3         11         0
1179         0              3         11         0
2200         0              3         11         0

[224 rows x 29 columns]
```

In [91]:
```
# Scatter plot
plt.scatter(R['NumWebPurchases'], R['NumWebVisitsMonth'])
plt.show()
```



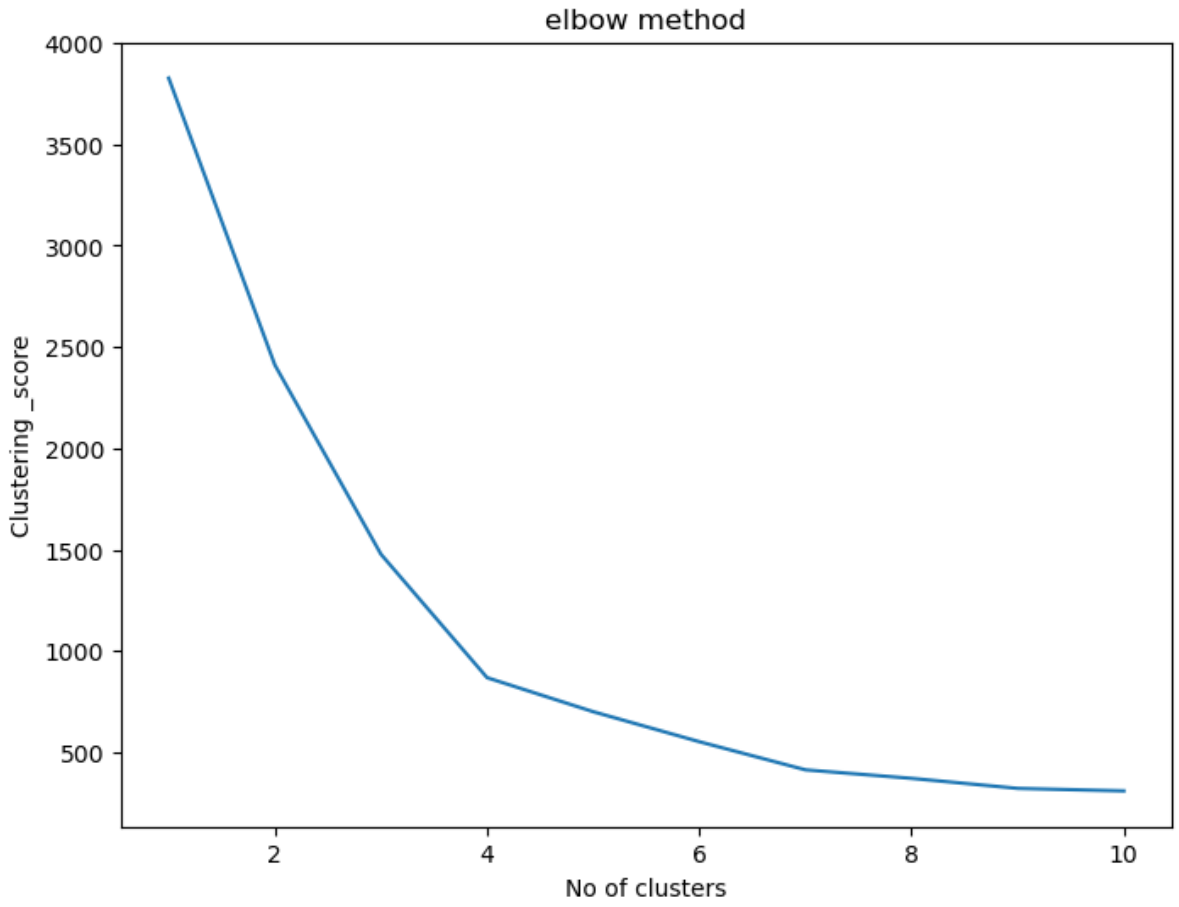
In [92]:
```
X = R.iloc[:,[16,19]].values
X.shape
```

Out[92]:
```
(224, 2)
```

In [93]:
```
Clustering_score= []

for i in range(1,11):
    kmeans= KMeans(n_clusters=i,init = 'random', random_state=42)
    kmeans.fit(X)
    Clustering_score.append(kmeans.inertia_)
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: User
Warning: KMeans is known to have a memory leak on Windows with MKL, when ther
e are less chunks than available threads. You can avoid it by setting the env
ironment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [94]:
```
# create elbow for find how many cluster we have
plt.figure(figsize = (8,6))
plt.plot(range(1,11), Clustering_score)
plt.xlabel('No of clusters')
plt.ylabel('Clustering _score')
plt.title("elbow method")
plt.show()
```



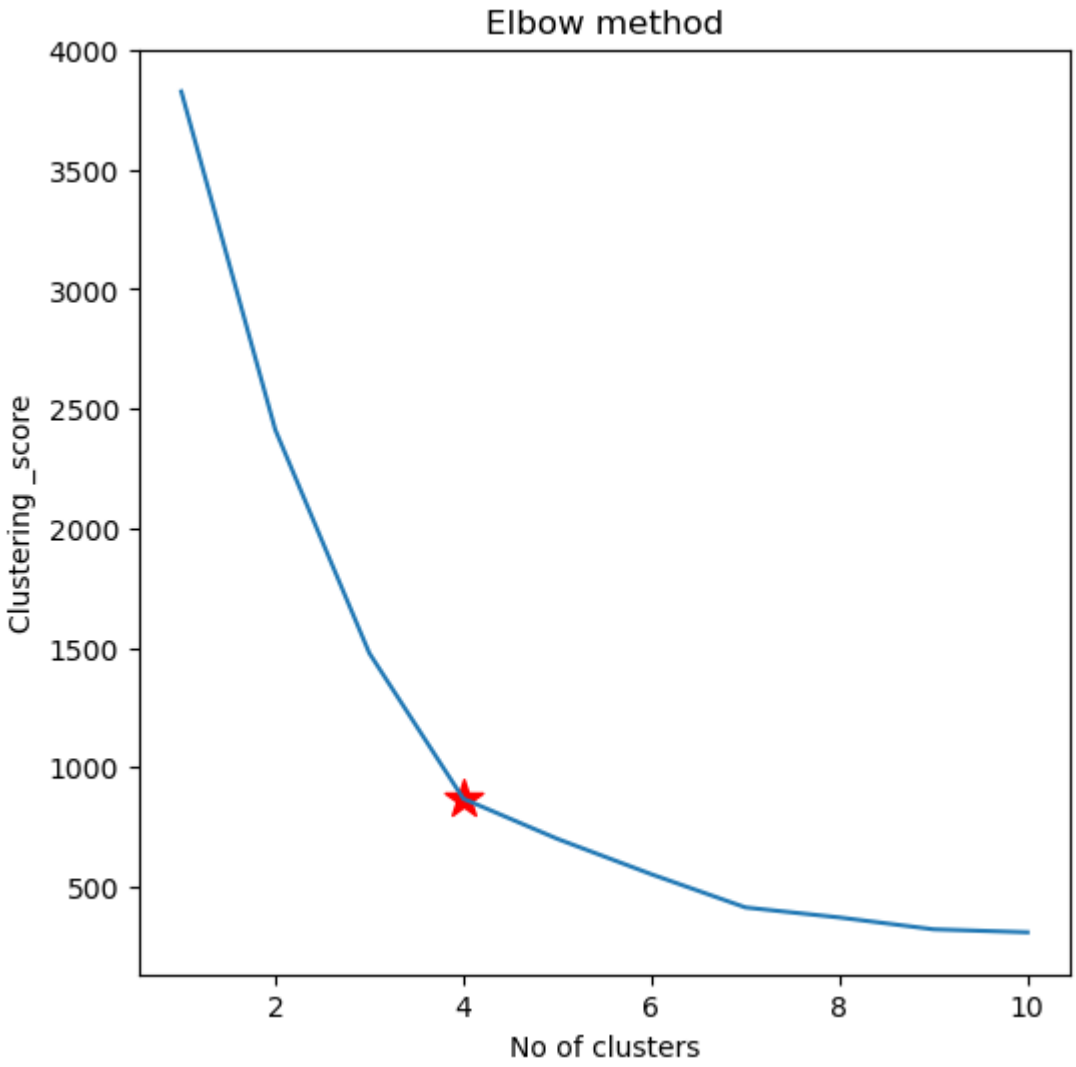
In [95]:
```
# Find Clustering_Score
Clustering_score
```

Out[95]:
```
[3827.0267857142862,
 2412.001680848829,
 1477.8262591256428,
 868.8733016124768,
 700.9578043907313,
 552.916925804828,
 414.11758723732396,
 372.1021475060474,
 322.6175940391149,
 310.09842773898504]
```

In [101]:
```
Clustering_score[3]
```

Out[101]:
```
868.8733016124768
```

In [103]:
```
plt.figure(figsize = (6,6))
plt.plot(range(1,11), Clustering_score)
plt.scatter(4, Clustering_score[3], s= 200, c='red', marker='*')
plt.xlabel('No of clusters')
plt.ylabel('Clustering _score')
plt.title("Elbow method")
plt.show()
```

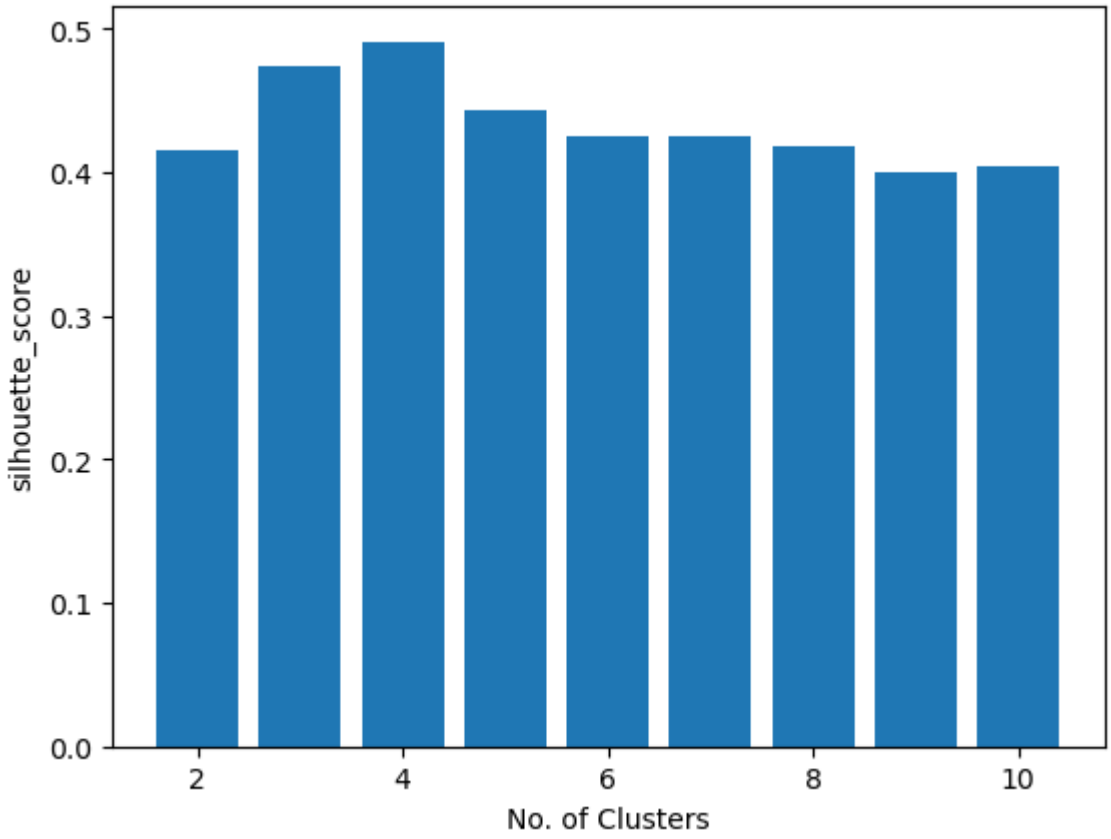


In [104]:
```
from sklearn.metrics import silhouette_score

silhouette_score_lst= []


for i in range (2,11):
    silhouette_score_lst.append(silhouette_score(X,(KMeans(n_clusters=i).fit_p
```

In [105]:
```
print(silhouette_score_lst)
```

```
[0.41559815492509106, 0.47463857889704186, 0.4911921071008035, 0.443753641874
8661, 0.4250932935585357, 0.42590439919761486, 0.41801507206637617, 0.4006124
005594317, 0.40477276144392454]
```

In [106]:
```
#plotting
k=[2,3,4,5,6,7,8,9,10]

plt.bar(k,silhouette_score_lst)
plt.xlabel('No. of Clusters')
plt.ylabel('silhouette_score')
plt.show()
```



1-highest value of bar from given clusters values will be selected

2-selecting number of clusters = 4

In [107]:
```
#set up a model
kmeans= KMeans(n_clusters=4, random_state=42)

#fit the model
kmeans.fit(X)

pred= kmeans.predict(X)
print(pred)
```

```
[1 0 0 3 3 3 0 3 3 1 0 3 0 1 1 3 0 1 0 0 0 0 3 0 1 1 1 1 1 0 1 3 1 0 0 3 1
 0 0 0 0 3 3 0 1 0 3 1 0 0 0 0 0 0 0 0 3 3 1 0 0 3 3 3 0 1 0 1 3 1 0 1 3 3
 0 0 0 0 3 1 0 0 0 1 0 0 0 3 0 3 3 3 3 3 0 0 3 1 0 0 3 0 3 0 0 3 0 0 0 1 3
 3 0 0 3 1 1 3 0 0 1 1 1 0 1 1 3 0 0 0 0 0 1 0 3 1 0 0 3 3 3 3 1 0 0 0 0 0
 0 1 3 3 0 3 1 0 0 3 0 0 0 1 3 1 0 0 1 0 0 3 1 1 3 0 1 0 1 0 1 0 0 1 0 0 0
 0 0 3 0 1 0 3 0 1 0 3 1 0 0 1 3 3 1 3 0 3 3 1 0 3 3 0 0 3 3 2 0 3 1 2 3 0
 3 0]
```

In [108]:
```
df['Cluster']= pd.DataFrame(pred, columns= ['cluster'])
df.head(10)
```

Out[108]:

ecency	MntWines	...	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCr
58	635	...	0	0	0	0	
38	11	...	0	0	0	0	
26	426	...	0	0	0	0	
26	11	...	0	0	0	0	
94	173	...	0	0	0	0	
16	520	...	0	0	0	0	
34	235	...	0	0	0	0	
32	76	...	0	0	0	0	
19	14	...	0	0	0	0	
68	28	...	1	0	0	0	

In [109]:
```
# Counts value in Cluster columns
df['Cluster'].value_counts()
```

Out[109]:
```
0.0    106
3.0     64
1.0     52
2.0      2
Name: Cluster, dtype: int64
```

In [110]:
```
#centroid of each clusters

kmeans.cluster_centers_
```

Out[110]:
```
array([[ 2.20754717,  7.05660377],
       [ 7.73076923,  5.96153846],
       [25.        ,  0.5       ],
       [ 3.390625  ,  2.25      ]])
```

In [113]:
```
kmeans.cluster_centers_[:,0]
```

Out[113]:
```
array([ 2.20754717,  7.73076923, 25.        ,  3.390625  ])
```

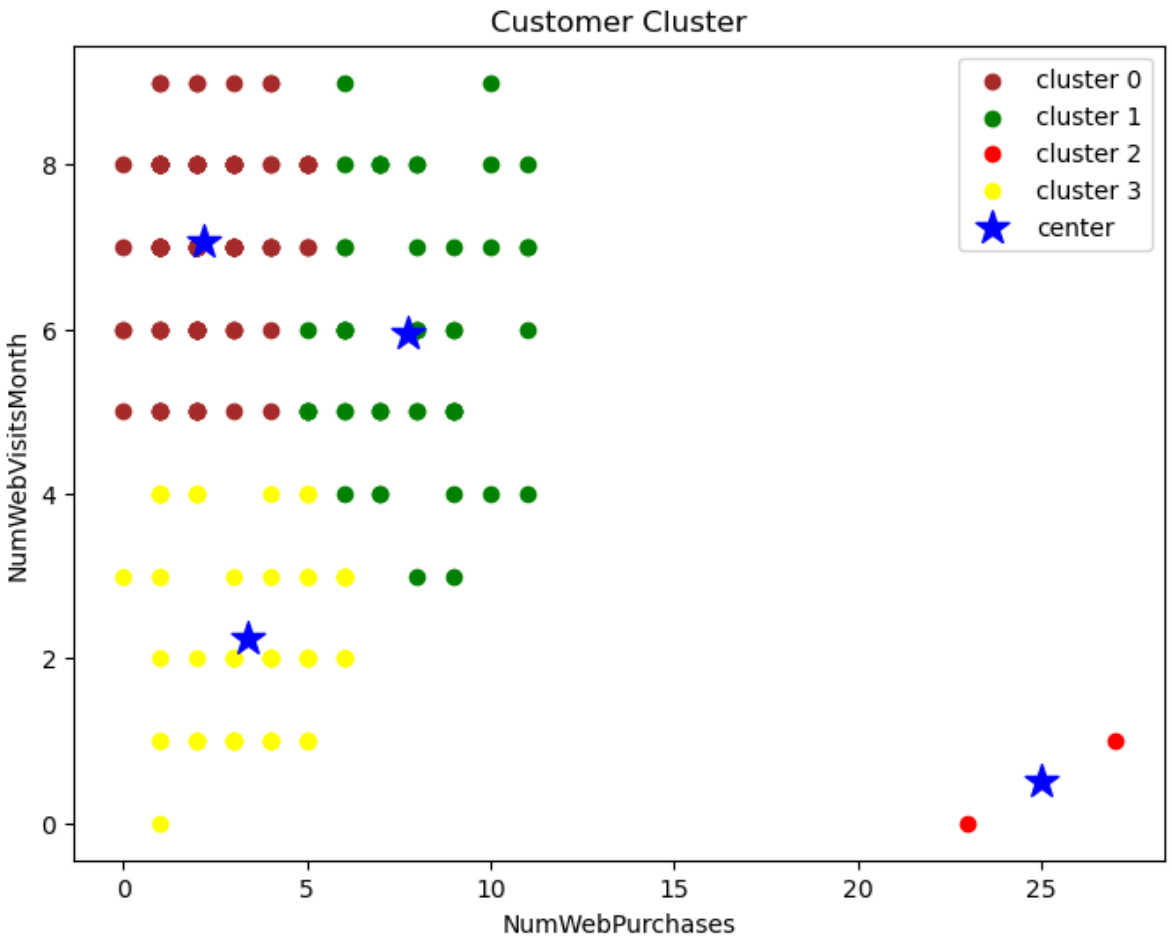
In [112]:
```
X[pred == 0,0]
```

Out[112]:
```
array([1, 4, 4, 3, 2, 2, 1, 2, 1, 2, 1, 3, 4, 3, 1, 1, 5, 2, 0, 1, 1, 5,
       1, 3, 3, 2, 1, 4, 2, 1, 3, 1, 1, 3, 3, 4, 2, 4, 2, 2, 1, 1, 2, 3,
       3, 2, 3, 0, 2, 3, 3, 2, 1, 2, 3, 2, 1, 3, 3, 2, 4, 1, 5, 2, 4, 1,
       1, 0, 5, 1, 1, 3, 1, 2, 2, 0, 5, 5, 2, 1, 5, 2, 0, 3, 3, 1, 5, 4,
       2, 2, 2, 1, 1, 2, 3, 2, 1, 1, 3, 2, 1, 1, 4, 1, 1, 2], dtype=int64)
```

In [114]:
```
X[pred == 0,1]
```

Out[114]:
```
array([8, 5, 7, 7, 6, 7, 7, 5, 5, 6, 5, 8, 8, 8, 7, 7, 8, 9, 7, 8, 7, 8,
       8, 8, 7, 8, 9, 8, 8, 8, 8, 7, 8, 5, 6, 9, 7, 9, 5, 7, 8, 5, 6, 7,
       6, 6, 7, 8, 8, 7, 8, 8, 7, 9, 6, 5, 7, 7, 7, 7, 7, 6, 8, 8, 6, 8,
       7, 5, 8, 6, 8, 8, 5, 7, 6, 6, 7, 8, 6, 7, 7, 8, 6, 7, 9, 7, 8, 7,
       7, 5, 7, 8, 8, 6, 7, 5, 6, 7, 8, 6, 9, 8, 7, 7, 6, 8], dtype=int64)
```

In [120]:
```
plt.figure(figsize= (8,6))
plt.scatter(X[pred==0,0],X[pred==0,1],c = 'brown',label = 'cluster 0')
plt.scatter(X[pred==1,0],X[pred==1,1],c = 'green',label = 'cluster 1')
plt.scatter(X[pred==2,0],X[pred==2,1],c = 'red',label = 'cluster 2')
plt.scatter(X[pred==3,0],X[pred==3,1],c = 'yellow',label = 'cluster 3')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], s= 200
plt.xlabel('NumWebPurchases')
plt.ylabel('NumWebVisitsMonth')
plt.title('Customer Cluster')
plt.legend()
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: