Vaibhav Kumar[¶](#gjdgxs)

### RollNo 19[¶](#30j0zll)

K\_mean\_Customer\_dataset[¶](#1fob9te)

In [1]:

**import** numpy **as** np  
**import** pandas **as** pd  
**from** sklearn.datasets **import** make\_blobs  
**import** matplotlib.pyplot **as** plt  
**import** seaborn **as** sns  
**from** sklearn.cluster **import** KMeans  
**from** sklearn.metrics **import** silhouette\_score  
**from** sklearn.preprocessing **import** StandardScaler

In [2]:

df**=**pd**.**read\_csv("D:\\vk\\TRIM 3\\ML\\DATASET\\Mall\_Customers.csv")

In [3]:

df

Out[3]:

|  | **CustomerID** | **Gender** | **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 |
| **...** | ... | ... | ... | ... | ... |
| **195** | 196 | Female | 35 | 120 | 79 |
| **196** | 197 | Female | 45 | 126 | 28 |
| **197** | 198 | Male | 32 | 126 | 74 |
| **198** | 199 | Male | 32 | 137 | 18 |
| **199** | 200 | Male | 30 | 137 | 83 |

200 rows × 5 columns

In [4]:

df**.**info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 200 entries, 0 to 199  
Data columns (total 5 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 CustomerID 200 non-null int64   
 1 Gender 200 non-null object  
 2 Age 200 non-null int64   
 3 Annual Income (k$) 200 non-null int64   
 4 Spending Score (1-100) 200 non-null int64   
dtypes: int64(4), object(1)  
memory usage: 7.9+ KB

In [5]:

df**.**describe()

Out[5]:

|  | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| --- | --- | --- | --- | --- |
| **count** | 200.000000 | 200.000000 | 200.000000 | 200.000000 |
| **mean** | 100.500000 | 38.850000 | 60.560000 | 50.200000 |
| **std** | 57.879185 | 13.969007 | 26.264721 | 25.823522 |
| **min** | 1.000000 | 18.000000 | 15.000000 | 1.000000 |
| **25%** | 50.750000 | 28.750000 | 41.500000 | 34.750000 |
| **50%** | 100.500000 | 36.000000 | 61.500000 | 50.000000 |
| **75%** | 150.250000 | 49.000000 | 78.000000 | 73.000000 |
| **max** | 200.000000 | 70.000000 | 137.000000 | 99.000000 |

In [6]:

sns**.**heatmap(df**.**isnull())

Out[6]:

<AxesSubplot:>

we need to convert Gender into int[¶](#3znysh7)

In [7]:

df["Gender"]**.**dtypes

Out[7]:

dtype('O')

In [8]:

df['Gender'] **=** df['Gender']**.**map({'Male':1,'Female':0})

In [9]:

df

Out[9]:

|  | **CustomerID** | **Gender** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1 | 1 | 19 | 15 | 39 |
| **1** | 2 | 1 | 21 | 15 | 81 |
| **2** | 3 | 0 | 20 | 16 | 6 |
| **3** | 4 | 0 | 23 | 16 | 77 |
| **4** | 5 | 0 | 31 | 17 | 40 |
| **...** | ... | ... | ... | ... | ... |
| **195** | 196 | 0 | 35 | 120 | 79 |
| **196** | 197 | 0 | 45 | 126 | 28 |
| **197** | 198 | 1 | 32 | 126 | 74 |
| **198** | 199 | 1 | 32 | 137 | 18 |
| **199** | 200 | 1 | 30 | 137 | 83 |

200 rows × 5 columns

In [10]:

X**=**df[['Gender','Age','Annual Income (k$)','Spending Score (1-100)']]**.**values

In [11]:

X

Out[11]:

array([[ 1, 19, 15, 39],  
 [ 1, 21, 15, 81],  
 [ 0, 20, 16, 6],  
 [ 0, 23, 16, 77],  
 [ 0, 31, 17, 40],  
 [ 0, 22, 17, 76],  
 [ 0, 35, 18, 6],  
 [ 0, 23, 18, 94],  
 [ 1, 64, 19, 3],  
 [ 0, 30, 19, 72],  
 [ 1, 67, 19, 14],  
 [ 0, 35, 19, 99],  
 [ 0, 58, 20, 15],  
 [ 0, 24, 20, 77],  
 [ 1, 37, 20, 13],  
 [ 1, 22, 20, 79],  
 [ 0, 35, 21, 35],  
 [ 1, 20, 21, 66],  
 [ 1, 52, 23, 29],  
 [ 0, 35, 23, 98],  
 [ 1, 35, 24, 35],  
 [ 1, 25, 24, 73],  
 [ 0, 46, 25, 5],  
 [ 1, 31, 25, 73],  
 [ 0, 54, 28, 14],  
 [ 1, 29, 28, 82],  
 [ 0, 45, 28, 32],  
 [ 1, 35, 28, 61],  
 [ 0, 40, 29, 31],  
 [ 0, 23, 29, 87],  
 [ 1, 60, 30, 4],  
 [ 0, 21, 30, 73],  
 [ 1, 53, 33, 4],  
 [ 1, 18, 33, 92],  
 [ 0, 49, 33, 14],  
 [ 0, 21, 33, 81],  
 [ 0, 42, 34, 17],  
 [ 0, 30, 34, 73],  
 [ 0, 36, 37, 26],  
 [ 0, 20, 37, 75],  
 [ 0, 65, 38, 35],  
 [ 1, 24, 38, 92],  
 [ 1, 48, 39, 36],  
 [ 0, 31, 39, 61],  
 [ 0, 49, 39, 28],  
 [ 0, 24, 39, 65],  
 [ 0, 50, 40, 55],  
 [ 0, 27, 40, 47],  
 [ 0, 29, 40, 42],  
 [ 0, 31, 40, 42],  
 [ 0, 49, 42, 52],  
 [ 1, 33, 42, 60],  
 [ 0, 31, 43, 54],  
 [ 1, 59, 43, 60],  
 [ 0, 50, 43, 45],  
 [ 1, 47, 43, 41],  
 [ 0, 51, 44, 50],  
 [ 1, 69, 44, 46],  
 [ 0, 27, 46, 51],  
 [ 1, 53, 46, 46],  
 [ 1, 70, 46, 56],  
 [ 1, 19, 46, 55],  
 [ 0, 67, 47, 52],  
 [ 0, 54, 47, 59],  
 [ 1, 63, 48, 51],  
 [ 1, 18, 48, 59],  
 [ 0, 43, 48, 50],  
 [ 0, 68, 48, 48],  
 [ 1, 19, 48, 59],  
 [ 0, 32, 48, 47],  
 [ 1, 70, 49, 55],  
 [ 0, 47, 49, 42],  
 [ 0, 60, 50, 49],  
 [ 0, 60, 50, 56],  
 [ 1, 59, 54, 47],  
 [ 1, 26, 54, 54],  
 [ 0, 45, 54, 53],  
 [ 1, 40, 54, 48],  
 [ 0, 23, 54, 52],  
 [ 0, 49, 54, 42],  
 [ 1, 57, 54, 51],  
 [ 1, 38, 54, 55],  
 [ 1, 67, 54, 41],  
 [ 0, 46, 54, 44],  
 [ 0, 21, 54, 57],  
 [ 1, 48, 54, 46],  
 [ 0, 55, 57, 58],  
 [ 0, 22, 57, 55],  
 [ 0, 34, 58, 60],  
 [ 0, 50, 58, 46],  
 [ 0, 68, 59, 55],  
 [ 1, 18, 59, 41],  
 [ 1, 48, 60, 49],  
 [ 0, 40, 60, 40],  
 [ 0, 32, 60, 42],  
 [ 1, 24, 60, 52],  
 [ 0, 47, 60, 47],  
 [ 0, 27, 60, 50],  
 [ 1, 48, 61, 42],  
 [ 1, 20, 61, 49],  
 [ 0, 23, 62, 41],  
 [ 0, 49, 62, 48],  
 [ 1, 67, 62, 59],  
 [ 1, 26, 62, 55],  
 [ 1, 49, 62, 56],  
 [ 0, 21, 62, 42],  
 [ 0, 66, 63, 50],  
 [ 1, 54, 63, 46],  
 [ 1, 68, 63, 43],  
 [ 1, 66, 63, 48],  
 [ 1, 65, 63, 52],  
 [ 0, 19, 63, 54],  
 [ 0, 38, 64, 42],  
 [ 1, 19, 64, 46],  
 [ 0, 18, 65, 48],  
 [ 0, 19, 65, 50],  
 [ 0, 63, 65, 43],  
 [ 0, 49, 65, 59],  
 [ 0, 51, 67, 43],  
 [ 0, 50, 67, 57],  
 [ 1, 27, 67, 56],  
 [ 0, 38, 67, 40],  
 [ 0, 40, 69, 58],  
 [ 1, 39, 69, 91],  
 [ 0, 23, 70, 29],  
 [ 0, 31, 70, 77],  
 [ 1, 43, 71, 35],  
 [ 1, 40, 71, 95],  
 [ 1, 59, 71, 11],  
 [ 1, 38, 71, 75],  
 [ 1, 47, 71, 9],  
 [ 1, 39, 71, 75],  
 [ 0, 25, 72, 34],  
 [ 0, 31, 72, 71],  
 [ 1, 20, 73, 5],  
 [ 0, 29, 73, 88],  
 [ 0, 44, 73, 7],  
 [ 1, 32, 73, 73],  
 [ 1, 19, 74, 10],  
 [ 0, 35, 74, 72],  
 [ 0, 57, 75, 5],  
 [ 1, 32, 75, 93],  
 [ 0, 28, 76, 40],  
 [ 0, 32, 76, 87],  
 [ 1, 25, 77, 12],  
 [ 1, 28, 77, 97],  
 [ 1, 48, 77, 36],  
 [ 0, 32, 77, 74],  
 [ 0, 34, 78, 22],  
 [ 1, 34, 78, 90],  
 [ 1, 43, 78, 17],  
 [ 1, 39, 78, 88],  
 [ 0, 44, 78, 20],  
 [ 0, 38, 78, 76],  
 [ 0, 47, 78, 16],  
 [ 0, 27, 78, 89],  
 [ 1, 37, 78, 1],  
 [ 0, 30, 78, 78],  
 [ 1, 34, 78, 1],  
 [ 0, 30, 78, 73],  
 [ 0, 56, 79, 35],  
 [ 0, 29, 79, 83],  
 [ 1, 19, 81, 5],  
 [ 0, 31, 81, 93],  
 [ 1, 50, 85, 26],  
 [ 0, 36, 85, 75],  
 [ 1, 42, 86, 20],  
 [ 0, 33, 86, 95],  
 [ 0, 36, 87, 27],  
 [ 1, 32, 87, 63],  
 [ 1, 40, 87, 13],  
 [ 1, 28, 87, 75],  
 [ 1, 36, 87, 10],  
 [ 1, 36, 87, 92],  
 [ 0, 52, 88, 13],  
 [ 0, 30, 88, 86],  
 [ 1, 58, 88, 15],  
 [ 1, 27, 88, 69],  
 [ 1, 59, 93, 14],  
 [ 1, 35, 93, 90],  
 [ 0, 37, 97, 32],  
 [ 0, 32, 97, 86],  
 [ 1, 46, 98, 15],  
 [ 0, 29, 98, 88],  
 [ 0, 41, 99, 39],  
 [ 1, 30, 99, 97],  
 [ 0, 54, 101, 24],  
 [ 1, 28, 101, 68],  
 [ 0, 41, 103, 17],  
 [ 0, 36, 103, 85],  
 [ 0, 34, 103, 23],  
 [ 0, 32, 103, 69],  
 [ 1, 33, 113, 8],  
 [ 0, 38, 113, 91],  
 [ 0, 47, 120, 16],  
 [ 0, 35, 120, 79],  
 [ 0, 45, 126, 28],  
 [ 1, 32, 126, 74],  
 [ 1, 32, 137, 18],  
 [ 1, 30, 137, 83]], dtype=int64)

In [12]:

kmeans**=**KMeans(n\_clusters**=**2)  
kmeans**.**fit(X)  
y\_kmeans**=**kmeans**.**predict(X)

In [13]:

print(y\_kmeans)

[1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1  
 0 1 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1  
 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1  
 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0  
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1  
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]

In [14]:

kmeans**.**cluster\_centers\_

Out[14]:

array([[ 0.44705882, 28.95294118, 62.17647059, 73.62352941],  
 [ 0.43478261, 46.16521739, 59.36521739, 32.88695652]])

In [15]:

n\_cluster**=**6  
silhouette\_avg**=**silhouette\_score(X,y\_kmeans)  
print("For n\_cluster= ",n\_cluster,"average silhouette\_score is :",silhouette\_avg)

For n\_cluster= 6 average silhouette\_score is : 0.29307334005502633

In [16]:

plt**.**scatter(X[:,0],X[:,1],c**=**y\_kmeans,s**=**50,cmap**=**'rainbow')

Out[16]:

<matplotlib.collections.PathCollection at 0x215e6aeffd0>

In [17]:

Error **=**[]  
**for** i **in** range (2,11):  
 kmeans **=** KMeans(n\_clusters **=** i)**.**fit(X)  
*# kmeans.fit(x)*  
 Error**.**append(kmeans**.**inertia\_)  
  
plt**.**plot(range(2, 11), Error)  
plt**.**title( 'Elbow method')  
plt**.**xlabel('No of clusters')  
plt**.**ylabel('Error')  
plt**.**show()

6 clusters[¶](#2et92p0)

In [ ]:

In [ ]: