PROJECT-5

June 15, 2023

#PROBLEM STATEMENT:- TO DIVIDE THE DATA INTO CLUSTERS BASED ON THE SIMILARITY

(The transactions made by a UK-based, registered, non-store online retailer between December 1, 2010, and December 9, 2011, are all included in the transnational data set known as online retail. The company primarily offers one-of-a-kind gifts for every occasion. The company has a large number of wholesalers as clients.Company ObjectiveUsing the global online retail dataset, we will design a clustering model and select the ideal group of clients for the business to target.)

[1]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

[2]:

```
df=pd.read_csv(r"/content/OnlineRetail.csv")
df
```

[2]:

```
      InvoiceNo StockCode                          Description  Quantity  \
0        536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER       6.0
1        536365     71053                  WHITE METAL LANTERN       6.0
2        536365    84406B       CREAM CUPID HEARTS COAT HANGER       8.0
3        536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
4        536365    84029E       RED WOOLLY HOTTIE WHITE HEART.       6.0
…           …        …                                   …       …
37000    539453    84659A              WHITE TRAVEL ALARM CLOCK       3.0
37001    539453     84685                 BEACH HUT KEY CABINET       1.0
37002    539453     84688              BEACH HUT DESIGN BLACKBOARD       1.0
37003    539453     84754      S/15 SILVER GLASS BAUBLES IN BAG       1.0
37004        53       NaN                                  NaN       NaN

            InvoiceDate  UnitPrice  CustomerID         Country
0      01-12-2010 08:26       2.55     17850.0  United Kingdom
1      01-12-2010 08:26       3.39     17850.0  United Kingdom
2      01-12-2010 08:26       2.75     17850.0  United Kingdom
3      01-12-2010 08:26       3.39     17850.0  United Kingdom
4      01-12-2010 08:26       3.39     17850.0  United Kingdom
…                   …         …         …               …
37000  17-12-2010 17:08       2.51         NaN  United Kingdom
37001  17-12-2010 17:08       7.62         NaN  United Kingdom
```

```
37002  17-12-2010 17:08       8.47         NaN  United Kingdom
37003  17-12-2010 17:08       2.51         NaN  United Kingdom
37004               NaN        NaN         NaN             NaN

[37005 rows x 8 columns]
```

[3]:
```
df.head()
```

[3]:
```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER       6.0
1    536365     71053                  WHITE METAL LANTERN       6.0
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER       8.0
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.       6.0

        InvoiceDate  UnitPrice  CustomerID         Country
0  01-12-2010 08:26       2.55     17850.0  United Kingdom
1  01-12-2010 08:26       3.39     17850.0  United Kingdom
2  01-12-2010 08:26       2.75     17850.0  United Kingdom
3  01-12-2010 08:26       3.39     17850.0  United Kingdom
4  01-12-2010 08:26       3.39     17850.0  United Kingdom
```

[4]:
```
df.tail()
```

[4]:
```
      InvoiceNo StockCode                          Description  Quantity  \
37000    539453    84659A           WHITE TRAVEL ALARM CLOCK       3.0
37001    539453     84685               BEACH HUT KEY CABINET       1.0
37002    539453     84688         BEACH HUT DESIGN BLACKBOARD       1.0
37003    539453     84754   S/15 SILVER GLASS BAUBLES IN BAG       1.0
37004        53       NaN                                 NaN       NaN

            InvoiceDate  UnitPrice  CustomerID         Country
37000  17-12-2010 17:08       2.51         NaN  United Kingdom
37001  17-12-2010 17:08       7.62         NaN  United Kingdom
37002  17-12-2010 17:08       8.47         NaN  United Kingdom
37003  17-12-2010 17:08       2.51         NaN  United Kingdom
37004               NaN        NaN         NaN             NaN
```

[5]:
```
df['InvoiceNo'].value_counts()
```

[5]:
```
537434     675
538071     652
538349     620
537638     601
537237     597
          …
C538680      1
```

```
C538681      1
C538682      1
537230       1
53           1
Name: InvoiceNo, Length: 1772, dtype: int64
```

```
[6]: df['CustomerID'].value_counts()
```

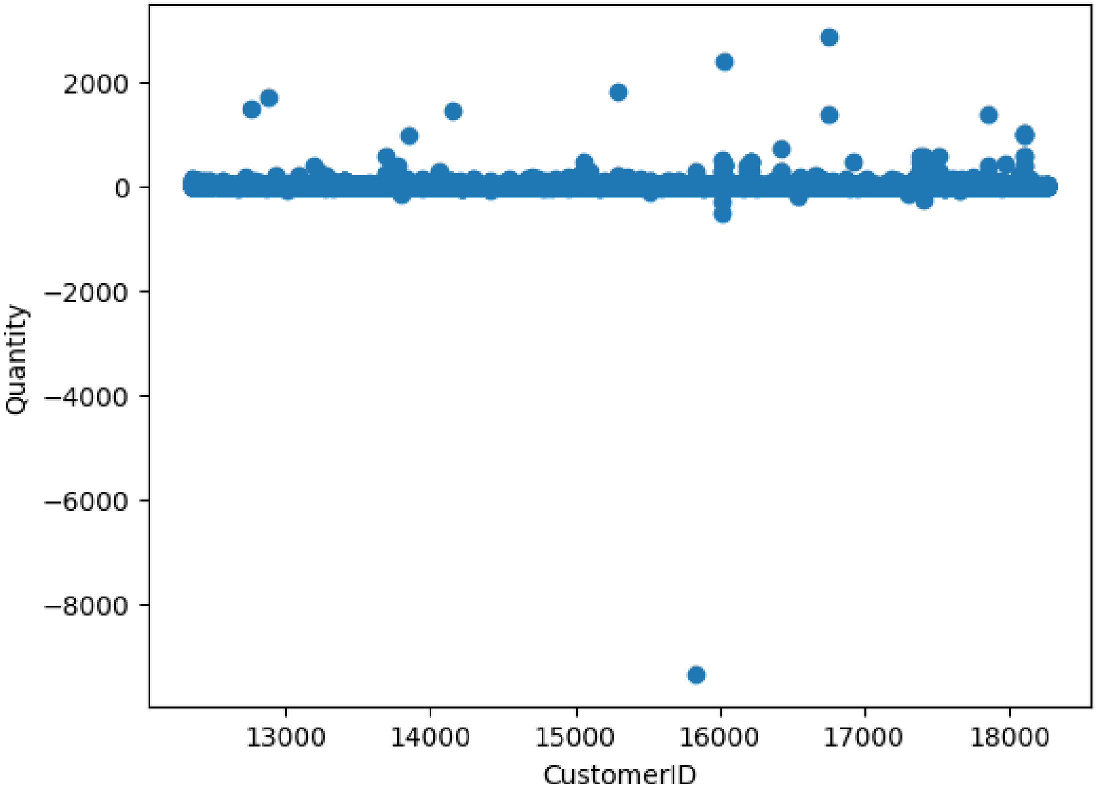
```
[6]: 12748.0    616
     17850.0    297
     17841.0    257
     14911.0    235
     14606.0    190
               …
     14576.0      1
     15179.0      1
     13145.0      1
     13481.0      1
     14867.0      1
     Name: CustomerID, Length: 897, dtype: int64
```

```
[7]: df['Quantity'].value_counts()
```

```
[7]:  1.0      12465
      2.0       5847
      12.0      3211
      3.0       2627
      6.0       2612
               …
     -19.0         1
     -43.0         1
     -35.0         1
     -72.0         1
     -500.0        1
     Name: Quantity, Length: 194, dtype: int64
```

```
[8]: plt.scatter(df["CustomerID"],df["Quantity"])
     plt.xlabel("CustomerID")
     plt.ylabel("Quantity")
```

```
[8]: Text(0, 0.5, 'Quantity')
```



[9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37005 entries, 0 to 37004
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   InvoiceNo    37005 non-null  object
 1   StockCode    37004 non-null  object
 2   Description  36887 non-null  object
 3   Quantity     37004 non-null  float64
 4   InvoiceDate  37004 non-null  object
 5   UnitPrice    37004 non-null  float64
 6   CustomerID   24364 non-null  float64
 7   Country      37004 non-null  object
dtypes: float64(3), object(5)
memory usage: 2.3+ MB
```

[10]:

```
df.isnull().sum()
```

```
[10]: InvoiceNo          0
      StockCode          1
      Description      118
      Quantity           1
      InvoiceDate        1
      UnitPrice          1
      CustomerID     12641
      Country            1
      dtype: int64
```

```
[11]: df.fillna(method='ffill',inplace=True)
```

```
[12]: from sklearn.cluster import KMeans
      km=KMeans()
      km
```

```
[12]: KMeans()
```

```
[13]: y_predicted=km.fit_predict(df[["CustomerID","Quantity"]])
      y_predicted
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
[13]: array([3, 3, 3, …, 2, 2, 2], dtype=int32)
```

```
[14]: df["cluster"]=y_predicted
      df.head()
```

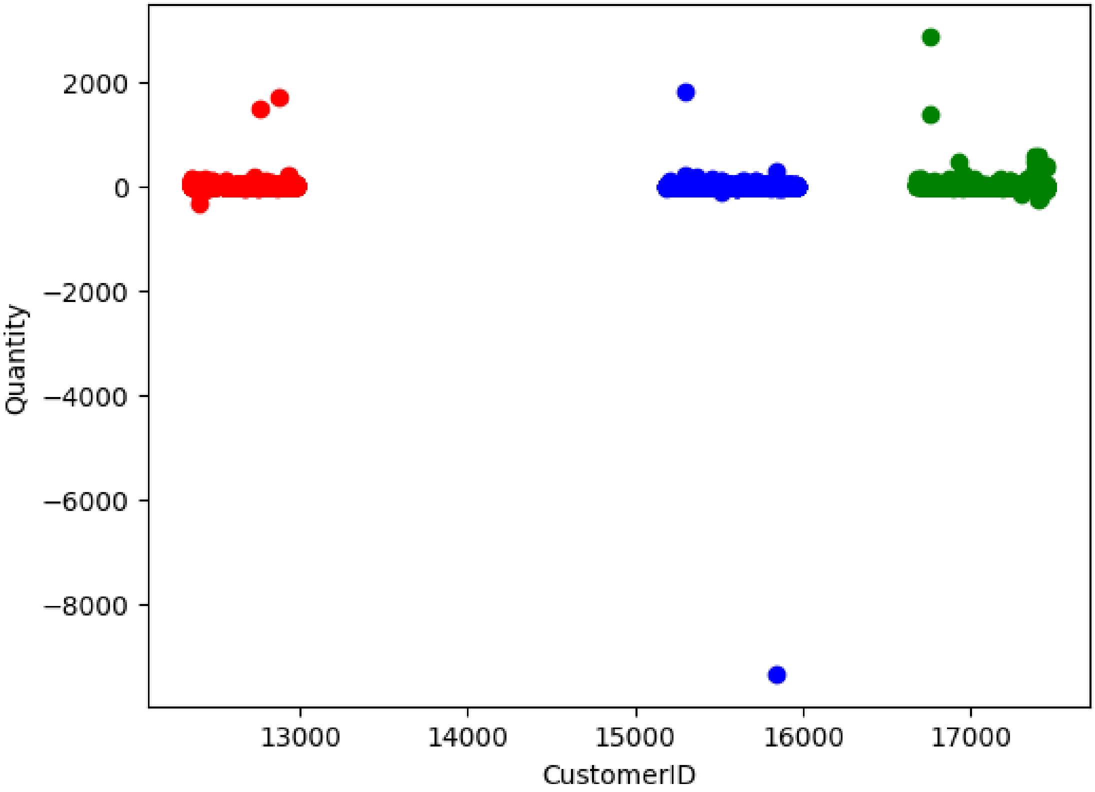
```
[14]:   InvoiceNo StockCode                          Description  Quantity  \
      0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER       6.0
      1    536365     71053                  WHITE METAL LANTERN       6.0
      2    536365    84406B       CREAM CUPID HEARTS COAT HANGER       8.0
      3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE       6.0
      4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.       6.0

                InvoiceDate  UnitPrice  CustomerID         Country  cluster
      0  01-12-2010 08:26       2.55     17850.0  United Kingdom        3
      1  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
      2  01-12-2010 08:26       2.75     17850.0  United Kingdom        3
      3  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
      4  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
```

```
[15]: df1=df[df.cluster==0]
      df2=df[df.cluster==1]
```

```
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["Quantity"],color="red")
plt.scatter(df2["CustomerID"],df2["Quantity"],color="green")
plt.scatter(df3["CustomerID"],df3["Quantity"],color="blue")
plt.xlabel("CustomerID")
plt.ylabel("Quantity")
```

[15]:
```
Text(0, 0.5, 'Quantity')
```



[16]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["Quantity"]])
df["Quantity"]=scaler.transform(df[["Quantity"]])
df.head()
```

[16]:
```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER  0.765196
1    536365     71053                  WHITE METAL LANTERN  0.765196
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER  0.765359
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE  0.765196
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.  0.765196
```

```
       InvoiceDate  UnitPrice  CustomerID         Country  cluster
0  01-12-2010 08:26       2.55     17850.0  United Kingdom        3
1  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
2  01-12-2010 08:26       2.75     17850.0  United Kingdom        3
3  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
4  01-12-2010 08:26       3.39     17850.0  United Kingdom        3
```

[17]:
```
scaler.fit(df[["CustomerID"]])
df["CustomerID"]=scaler.transform(df[["CustomerID"]])
df.head()
```

[17]:
```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER  0.765196
1    536365     71053                  WHITE METAL LANTERN  0.765196
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER  0.765359
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE  0.765196
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.  0.765196

       InvoiceDate  UnitPrice  CustomerID         Country  cluster
0  01-12-2010 08:26       2.55    0.929247  United Kingdom        3
1  01-12-2010 08:26       3.39    0.929247  United Kingdom        3
2  01-12-2010 08:26       2.75    0.929247  United Kingdom        3
3  01-12-2010 08:26       3.39    0.929247  United Kingdom        3
4  01-12-2010 08:26       3.39    0.929247  United Kingdom        3
```

#K-MeansClustering

[18]:
```
km=KMeans()
```

[19]:
```
y_predicted=km.fit_predict(df[["CustomerID","Quantity"]])
y_predicted
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

[19]:
```
array([4, 4, 4, …, 2, 2, 2], dtype=int32)
```

[20]:
```
df["New Cluster"]=y_predicted
df.head()
```

[20]:
```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER  0.765196
1    536365     71053                  WHITE METAL LANTERN  0.765196
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER  0.765359
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE  0.765196
```

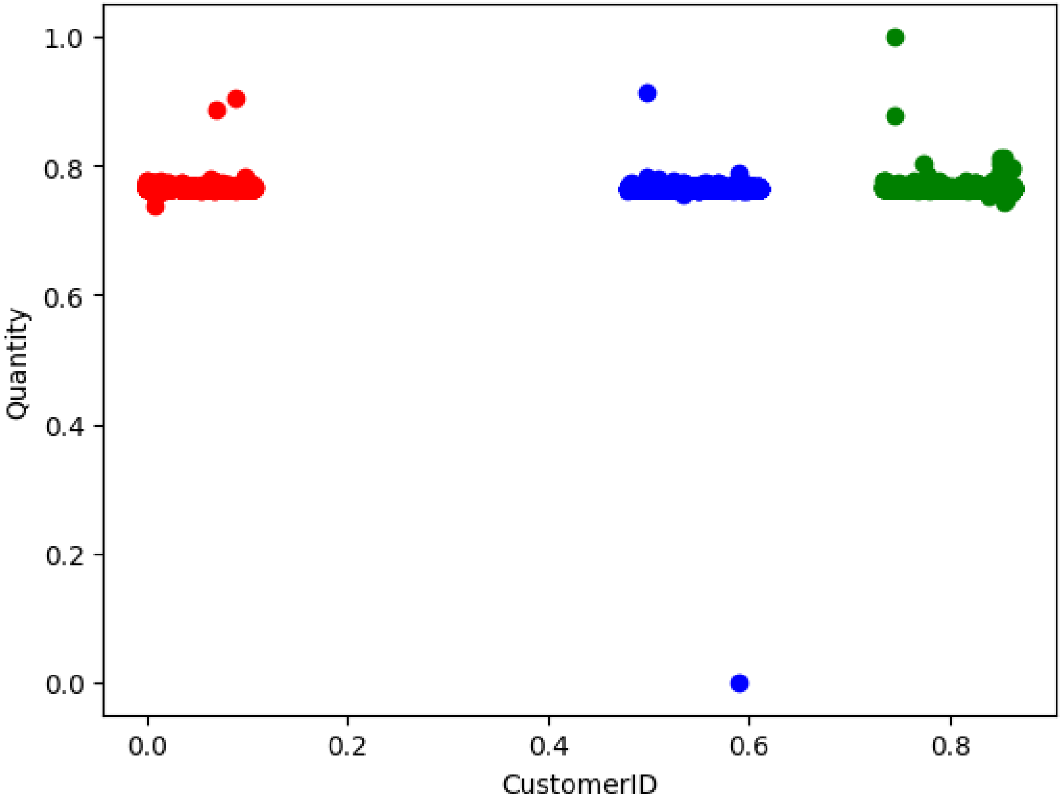
```
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.  0.765196

        InvoiceDate  UnitPrice  CustomerID         Country  cluster  \
0  01-12-2010 08:26       2.55    0.929247  United Kingdom        3
1  01-12-2010 08:26       3.39    0.929247  United Kingdom        3
2  01-12-2010 08:26       2.75    0.929247  United Kingdom        3
3  01-12-2010 08:26       3.39    0.929247  United Kingdom        3
4  01-12-2010 08:26       3.39    0.929247  United Kingdom        3

   New Cluster
0            4
1            4
2            4
3            4
4            4
```

[21]:
```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["CustomerID"],df1["Quantity"],color="red")
plt.scatter(df2["CustomerID"],df2["Quantity"],color="green")
plt.scatter(df3["CustomerID"],df3["Quantity"],color="blue")
plt.xlabel("CustomerID")
plt.ylabel("Quantity")
```

[21]:
```
Text(0, 0.5, 'Quantity')
```

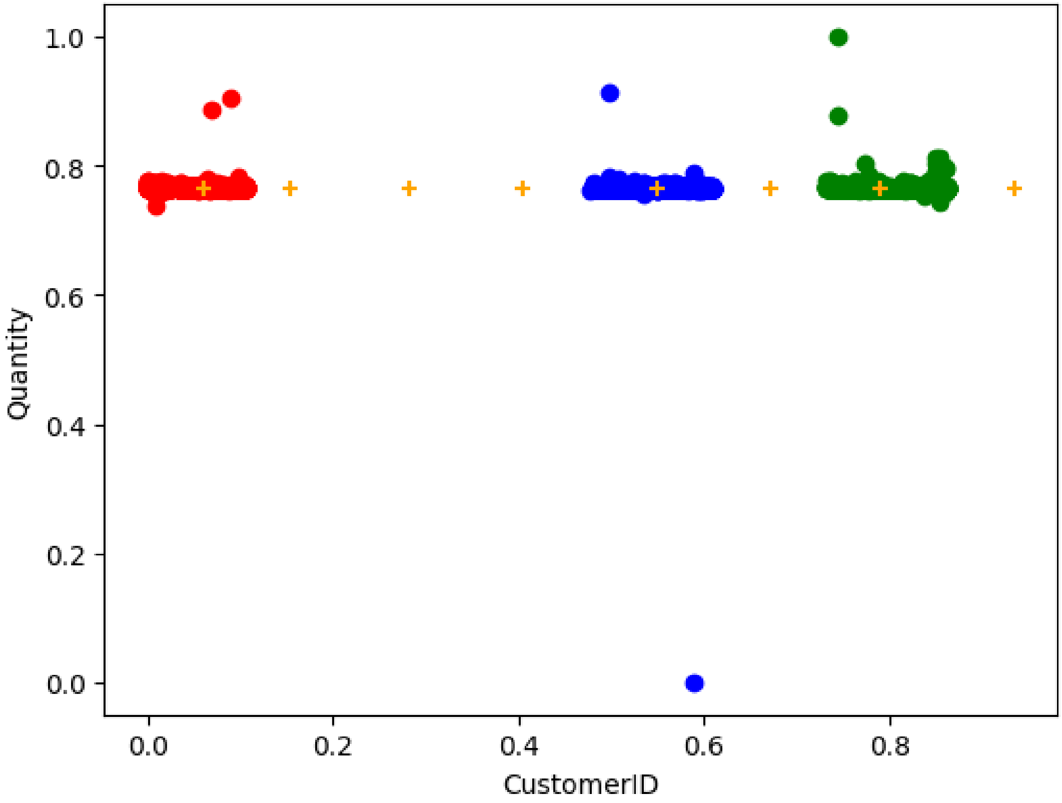


[22]:
```
km.cluster_centers_
```

[22]:
```
array([[0.06146662, 0.76539312],
       [0.79121684, 0.7652455 ],
       [0.55003698, 0.76509698],
       [0.28162316, 0.7661018 ],
       [0.93498295, 0.76535241],
       [0.6724887 , 0.76544962],
       [0.15279005, 0.76532485],
       [0.4037083 , 0.76545935]])
```

[23]:
```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["CustomerID"],df1["Quantity"],color="red")
plt.scatter(df2["CustomerID"],df2["Quantity"],color="green")
plt.scatter(df3["CustomerID"],df3["Quantity"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:
 ↪,1],color="orange",marker="+")
plt.xlabel("CustomerID")
plt.ylabel("Quantity")
```

[23]: Text(0, 0.5, 'Quantity')



[24]:
```
k_rng=range(1,10)
sse=[]
```

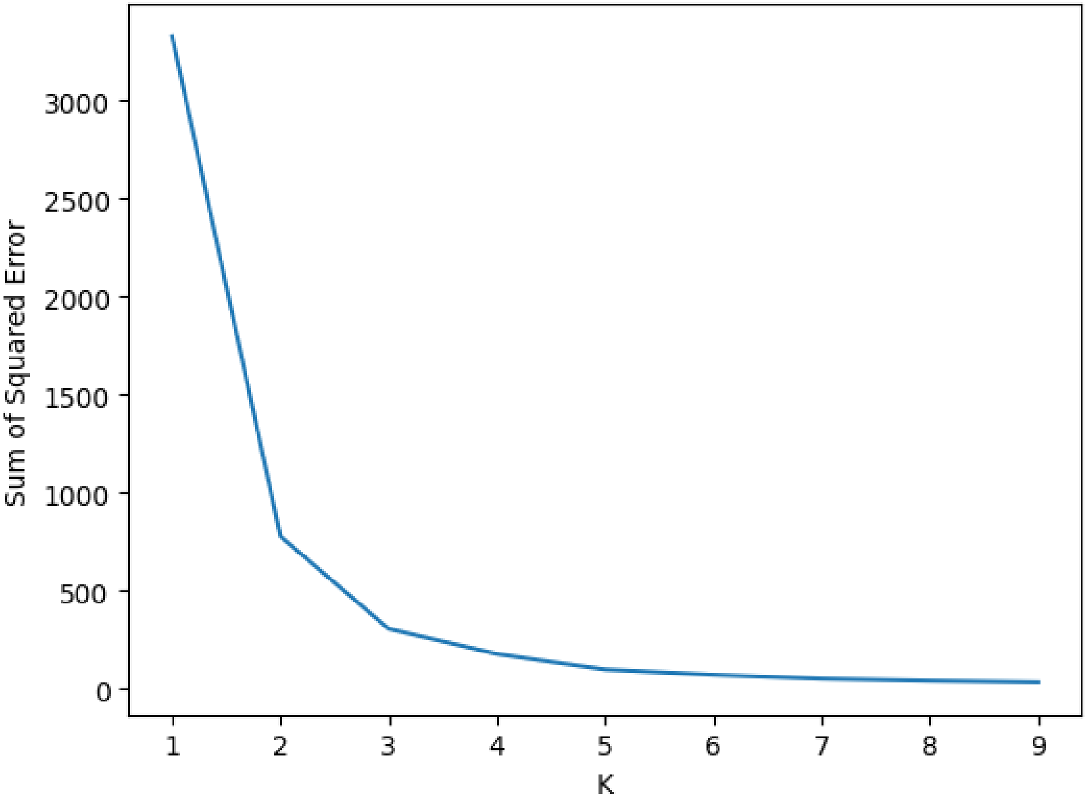
[25]:
```
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["CustomerID","Quantity"]])
 sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square error
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
```

```
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
[3326.9933911429544, 777.5097832607371, 305.84264678520316, 177.80203389783975,
98.38601031529377, 70.89747456762149, 51.23413647732975, 41.03276393124467,
33.40077310370238]
```

```
[25]: Text(0, 0.5, 'Sum of Squared Error')
```



#CONCLUSION:-

-> For the given dataset we use K-means Clustering and done the grouping based on the given data.In the above dataset we will take customer id and quantity based on that we make the clusters. When the K-value is low error rate is more and the K-value is high error rate is very high. So,finally we can Conclude the above dataset is bestfit for K-Means.