Project - 5 (DATASET: Online Retail) 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.

In [1]:

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

In [6]:

```python
df=pd.read_csv(r"C:\Users\Dheepack\OneDrive\Desktop\Project\OnlineRetail1.csv")
df
```

Out[6]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	C
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	Ki
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	Ki
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	Ki
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	Ki
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	Ki
...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	09-12-2011 12:50	0.85	12680.0	
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	09-12-2011 12:50	2.10	12680.0	
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	09-12-2011 12:50	4.15	12680.0	
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	09-12-2011 12:50	4.15	12680.0	
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	09-12-2011 12:50	4.95	12680.0	

541909 rows × 8 columns

In [7]:

```
df.head()
```

Out[7]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom

In [8]:

```
df.tail()
```

Out[8]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	C
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	09-12-2011 12:50	0.85	12680.0	
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	09-12-2011 12:50	2.10	12680.0	
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	09-12-2011 12:50	4.15	12680.0	
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	09-12-2011 12:50	4.15	12680.0	
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	09-12-2011 12:50	4.95	12680.0	

In [20]:

```python
df['InvoiceNo'].value_counts()
```

Out[20]:

```
InvoiceNo
573585    1114
581219     749
581492     731
580729     721
558475     705
          ...
554023       1
554022       1
554021       1
554020       1
C558901      1
Name: count, Length: 25900, dtype: int64
```

In [21]:

```python
df['CustomerID'].value_counts()
```

Out[21]:

```
CustomerID
17841.0    7983
14911.0    5903
14096.0    5128
12748.0    4642
14606.0    2782
           ...
15070.0       1
15753.0       1
17065.0       1
16881.0       1
16995.0       1
Name: count, Length: 4372, dtype: int64
```

In [22]:

```python
df['Quantity'].value_counts()
```

Out[22]:
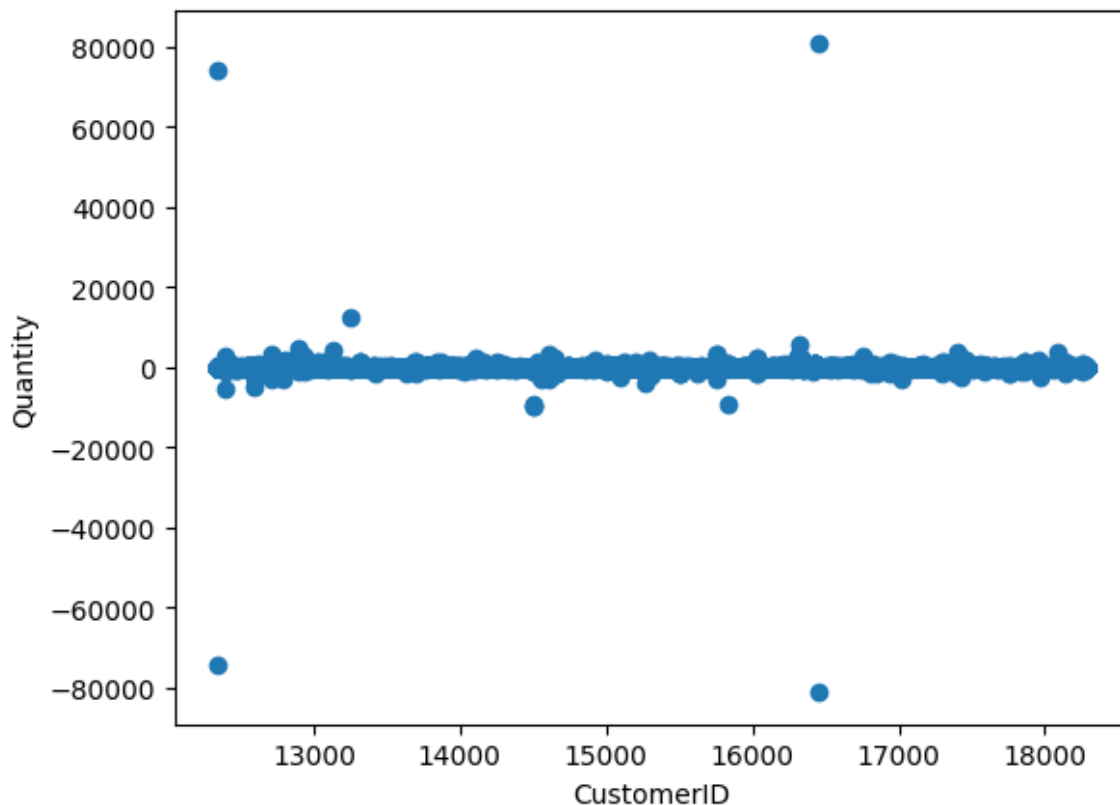
```
Quantity
 1     148227
 2      81829
 12     61063
 6      40868
 4      38484
          ...
-472        1
-161        1
-1206       1
-272        1
-80995      1
Name: count, Length: 722, dtype: int64
```

In [33]:

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

Out[33]:

Text(0, 0.5, 'Quantity')

In [27]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

In [28]:

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

Out[28]:

```
InvoiceNo           0
StockCode           0
Description       1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID      135080
Country             0
dtype: int64
```

In [29]:

```python
df.fillna(method='ffill',inplace=True)
```

In [30]:

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

Out[30]:

```
InvoiceNo       0
StockCode       0
Description     0
Quantity        0
InvoiceDate     0
UnitPrice       0
CustomerID      0
Country         0
dtype: int64
```

In [31]:

```python
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[31]:

```
KMeans()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [32]:

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

```
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
```

Out[32]:

```
array([0, 0, 0, ..., 4, 4, 4])
```

In [34]:

```python
df["cluster"]=y_predicted
df.head()
```
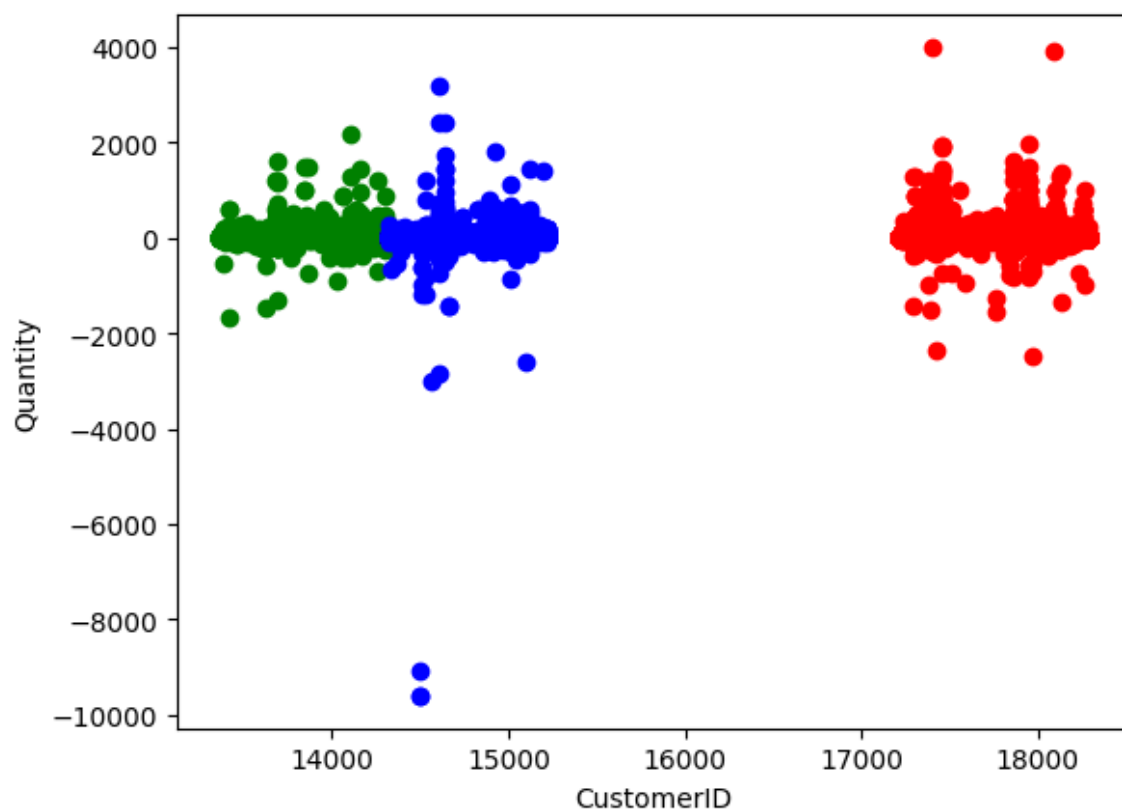
Out[34]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom

In [35]:

```
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")
```

Out[35]:

Text(0, 0.5, 'Quantity')

In [36]:

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

Out[36]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.500037	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	17850.0	United Kingdom

In [37]:

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

Out[37]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.500037	01-12-2010 08:26	2.55	0.926443	United Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	0.926443	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom

K-MeansClustering

In [38]:

```python
km=KMeans()
```

In [39]:

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

```
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
```

Out[39]:

```
array([6, 6, 6, ..., 3, 3, 3])
```

In [40]:

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

Out[40]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.500037	01-12-2010 08:26	2.55	0.926443	United Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	0.926443	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	0.926443	United Kingdom

In [41]:

```python
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")
```

Out[41]:

Text(0, 0.5, 'Quantity')
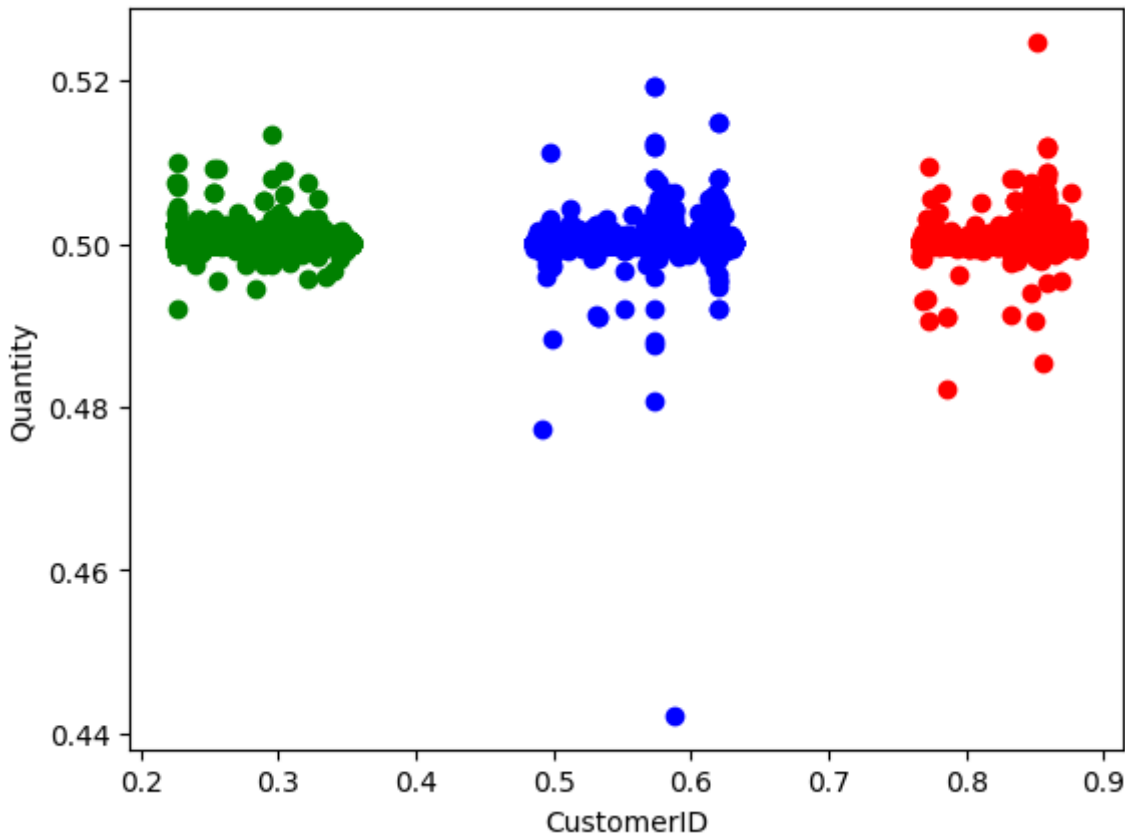
In [42]:

```python
km.cluster_centers_
```

Out[42]:

```
array([[0.8271312 , 0.50006239],
       [0.2901948 , 0.50006572],
       [0.55551996, 0.50005331],
       [0.05057095, 0.50006665],
       [0.70711435, 0.50005581],
       [0.41544492, 0.50005974],
       [0.93592371, 0.50005146],
       [0.15896113, 0.50005712]])
```

In [43]:

```python
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")
```

Out[43]:

Text(0, 0.5, 'Quantity')
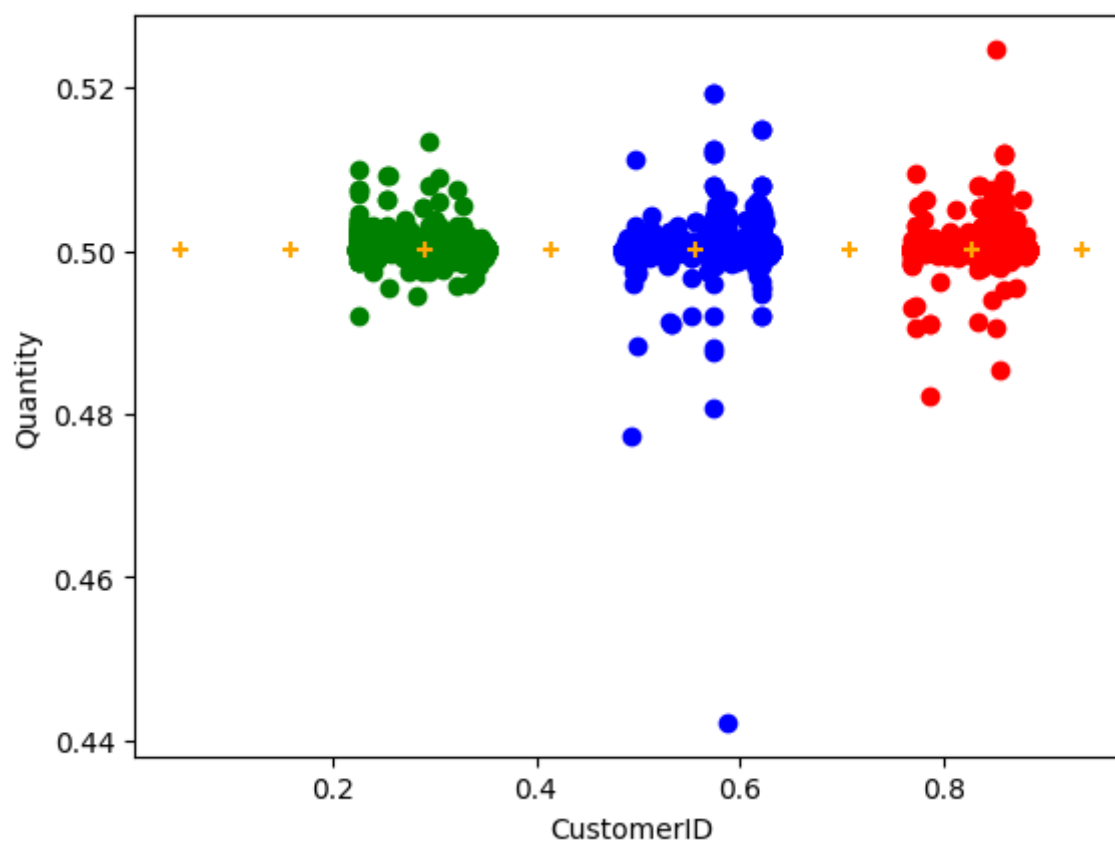
In [44]:

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

In [45]:

```python
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")
```

```
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(
C:\Users\krish\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\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` explicit
ly to suppress the warning
  warnings.warn(

[46374.84553398371, 11336.065305485055, 4915.894028081869, 2723.5191051895
64, 1695.163078963402, 1178.4246703192575, 902.5741514357653, 677.21678425
97247, 528.4339881385184]
```
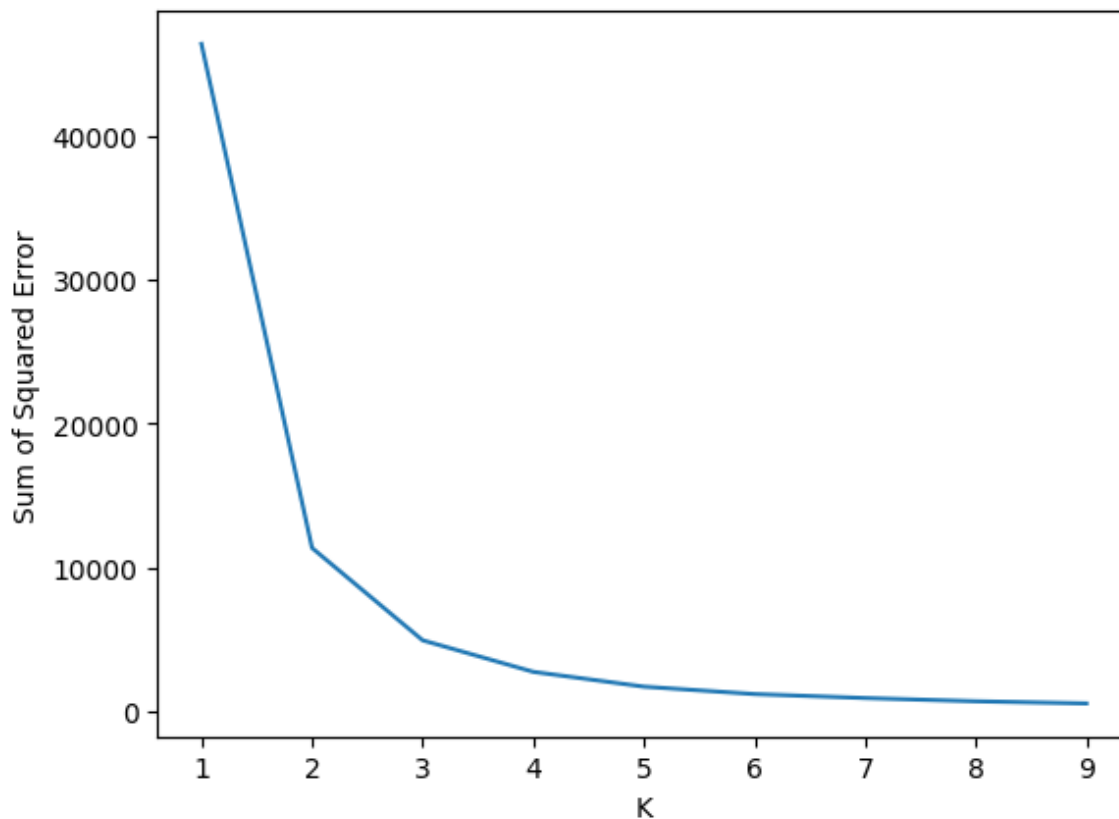
Out[45]:

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.

In [ ]: