

MINI_PROJECT

PROBLEM STATEMENT :- 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 Objective Using the global online retail dataset, we will design a clustering model and select the ideal group of clients for the business to target.

importing the libraries

In [7]:

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

In [9]:

```
df=pd.read_csv(r"C:\Users\prajapath Arjun\OneDrive\Documents\online retail.csv")
df
```

Out[9]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	
0	536365	85123A	WHITE HANGING HEART T- LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	↓
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	↓
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	↓
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	↓
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	↓
...	
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 [10]:

```
df.head()
```

Out[10]:

	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	Unitec Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	Unitec Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	Unitec Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	Unitec Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	Unitec Kingdom

In [11]:

```
df.tail()
```

Out[11]:

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

In [12]:

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

Out[12]:

```
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 [13]:

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

Out[13]:

```
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 [14]:

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

Out[14]:

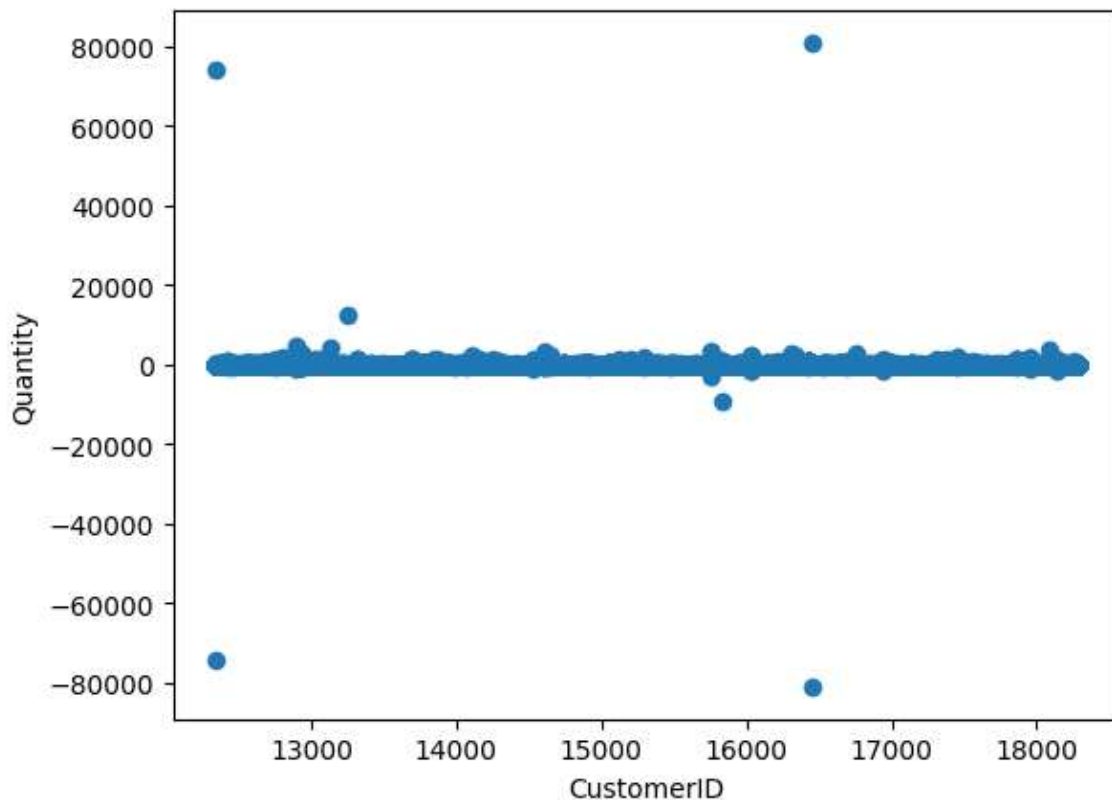
```
Quantity
1      148227
2      81829
12     61063
6      40868
4      38484
...
-472        1
-161        1
-1206       1
-272        1
-80995      1
Name: count, Length: 722, dtype: int64
```

In [15]:

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

Out[15]:

Text(0, 0.5, 'Quantity')



In [16]:

```
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 [17]:

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

Out[17]:

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

In [18]:

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

In [19]:

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

Out[19]:

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

In [20]:

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

Out[20]:

```
▼ KMeans
KMeans()
```

In [21]:

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

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

Out[21]:

```
array([4, 4, 4, ..., 3, 3, 3])
```

In [22]:

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

Out[22]:

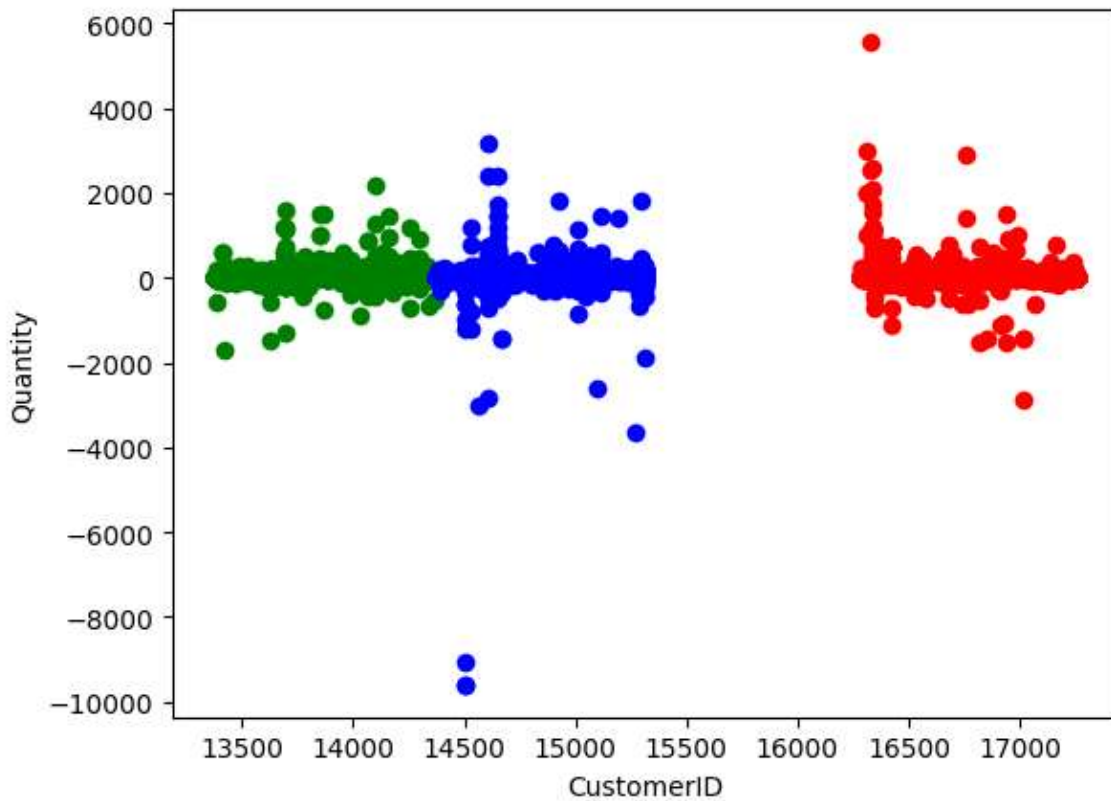
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T- LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	Unitec Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	Unitec Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT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

In [23]:

```
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[23]:

Text(0, 0.5, 'Quantity')



In [24]:

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

Out[24]:

	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	Unitec Kingdon
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	17850.0	Unitec Kingdon
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	17850.0	Unitec Kingdon
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	17850.0	Unitec Kingdon
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	17850.0	Unitec Kingdon

In [25]:

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

Out[25]:

	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	Unitec Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	0.926443	Unitec Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdom

K-MeansClustering

In [26]:

```
km=KMeans()
```

In [27]:

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

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

Out[27]:

```
array([5, 5, 5, ..., 2, 2, 2])
```

In [28]:

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

Out[28]:

	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	Unitec Kingdon
1	536365	71053	WHITE METAL LANTERN	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdon
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500049	01-12-2010 08:26	2.75	0.926443	Unitec Kingdon
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdon
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500037	01-12-2010 08:26	3.39	0.926443	Unitec Kingdon

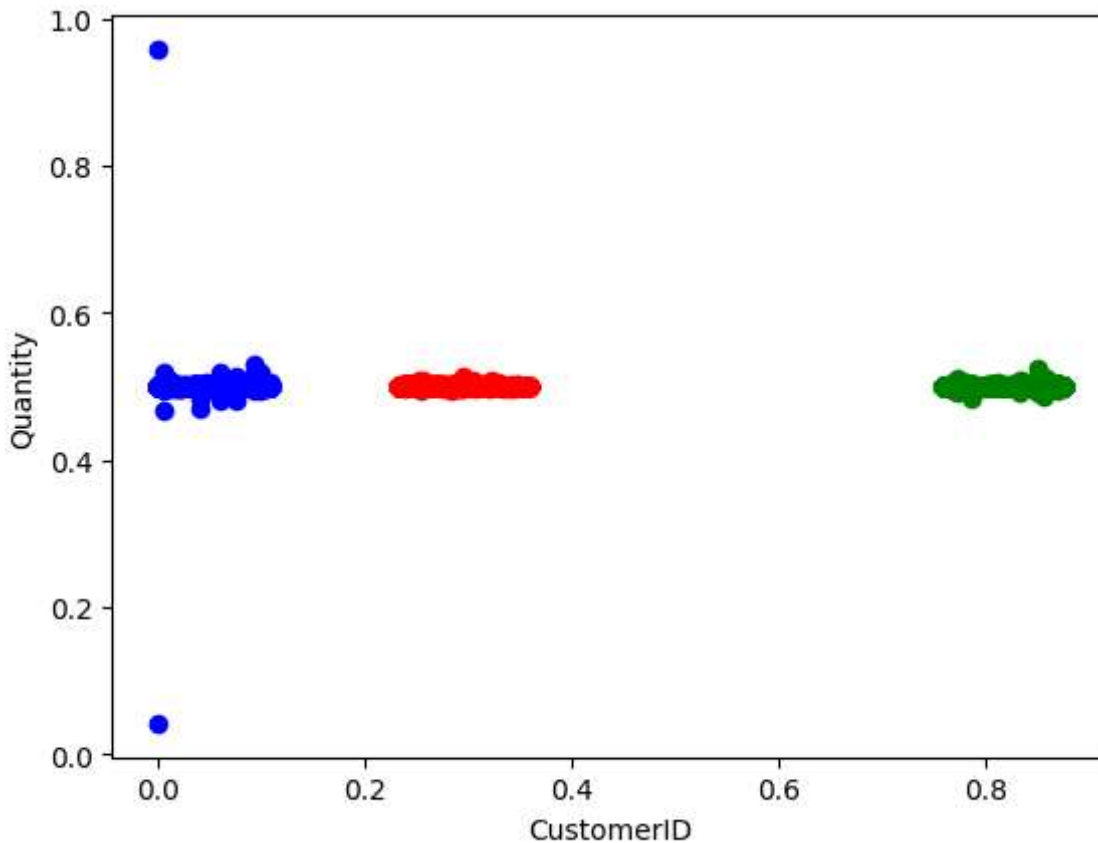


In [29]:

```
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[29]:

Text(0, 0.5, 'Quantity')



In [30]:

```
km.cluster_centers_
```

Out[30]:

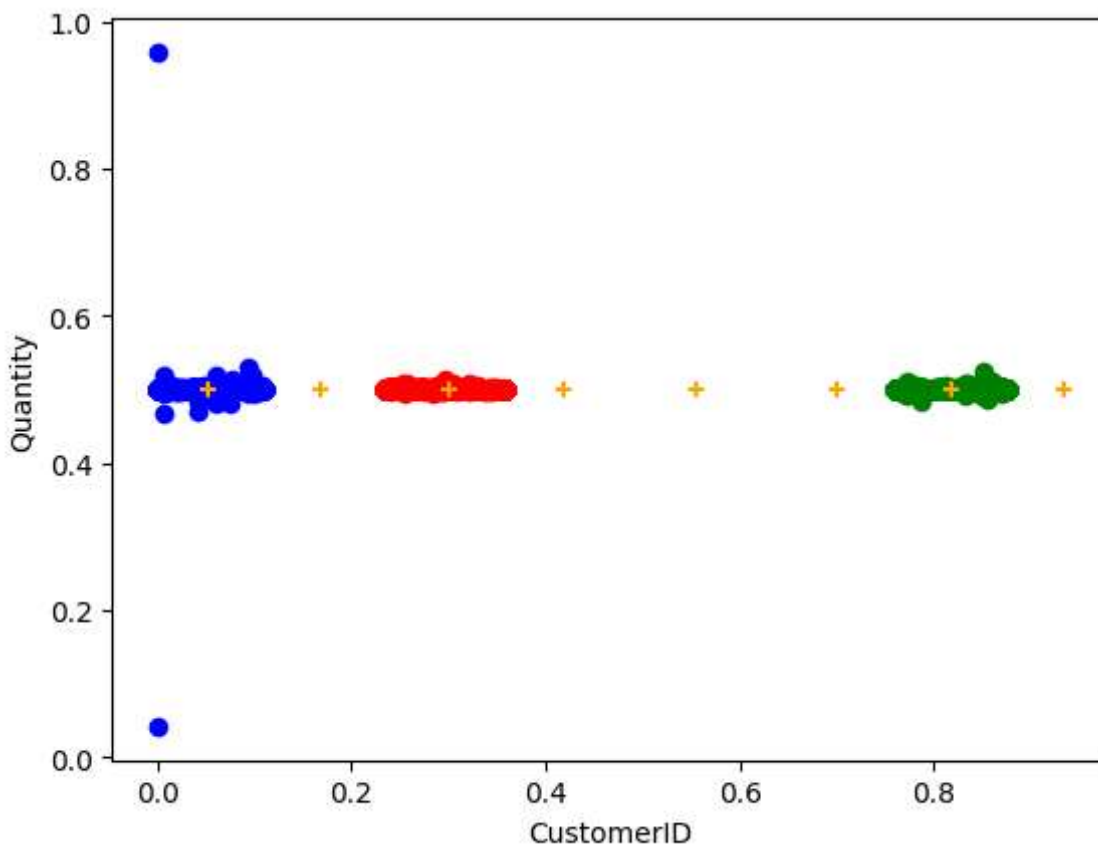
```
array([[0.29936477, 0.50006098],
       [0.81775848, 0.50005991],
       [0.0521429 , 0.50006713],
       [0.55424959, 0.50005371],
       [0.16699763, 0.5000602 ],
       [0.93299701, 0.50005095],
       [0.69997863, 0.50005818],
       [0.4186463 , 0.50006096]])
```

In [31]:

```
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[31]:

Text(0, 0.5, 'Quantity')



In [32]:

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

In [33]:

```

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\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-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(
```

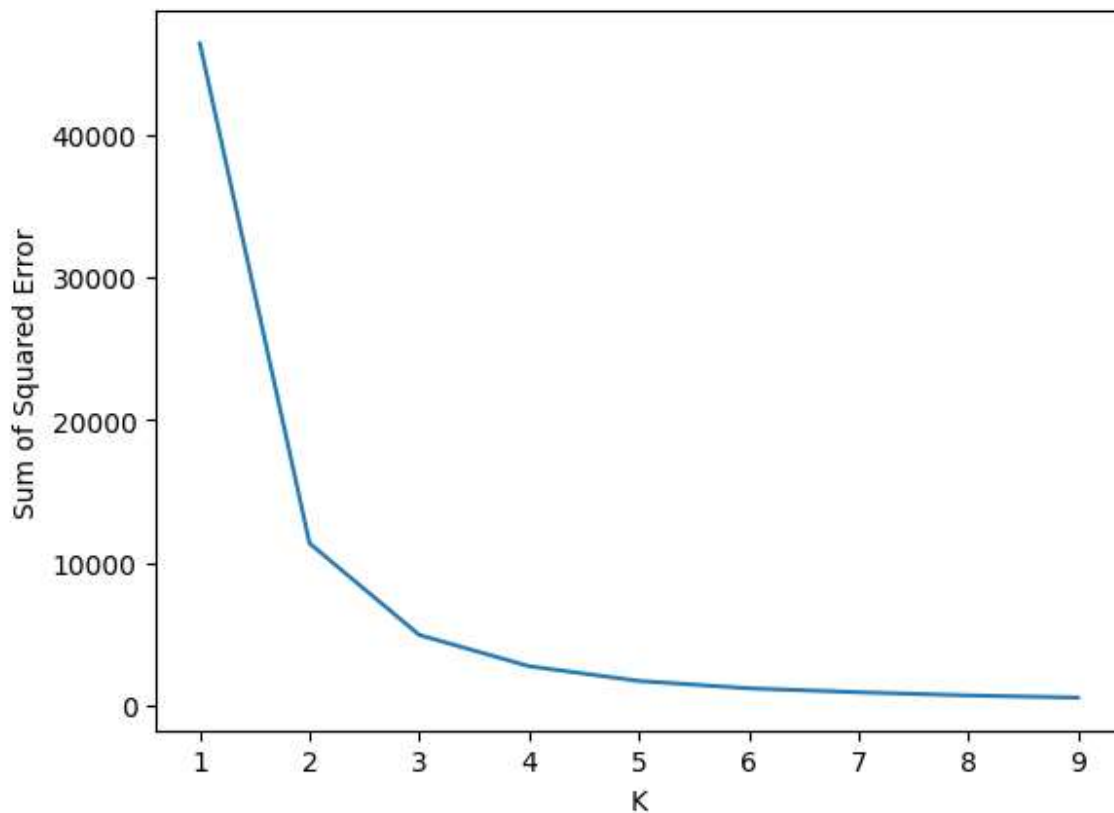
C:\Users\prajapath Arjun\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

```
warnings.warn(
```

```
[46374.84553398474, 11336.065820169119, 4915.901018401869, 2723.51910518953, 1695.077630848027, 1178.5876657857152, 902.6026843804057, 677.5036028786245, 528.8661304020511]
```

Out[33]:

```
Text(0, 0.5, 'Sum of Squared Error')
```



In []:

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