PROJECT5

Probelem Statement:Which model is suitable for OnlineRetailDataset

Importing Libraries

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Reading Dataset

In [5]:
```
df=pd.read_csv(r"C:\Users\P. VIJAY KUMAR\Downloads\OnlineRetailData\Online
df
```

Out[5]:

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

541909 rows × 8 columns

Data Preprocessing

In [6]:
```
1 df.head()
```

Out[6]:

	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 [7]:
```
1 df.describe()
```

Out[7]:

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

In [8]:
```
1 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 [10]:
```
1 df.isnull().sum()
```

Out[10]:
```
InvoiceNo          0
StockCode          0
Description      1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID     135080
Country            0
dtype: int64
```

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

In [12]:
```
1 df.isnull().sum()
```

Out[12]:
```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

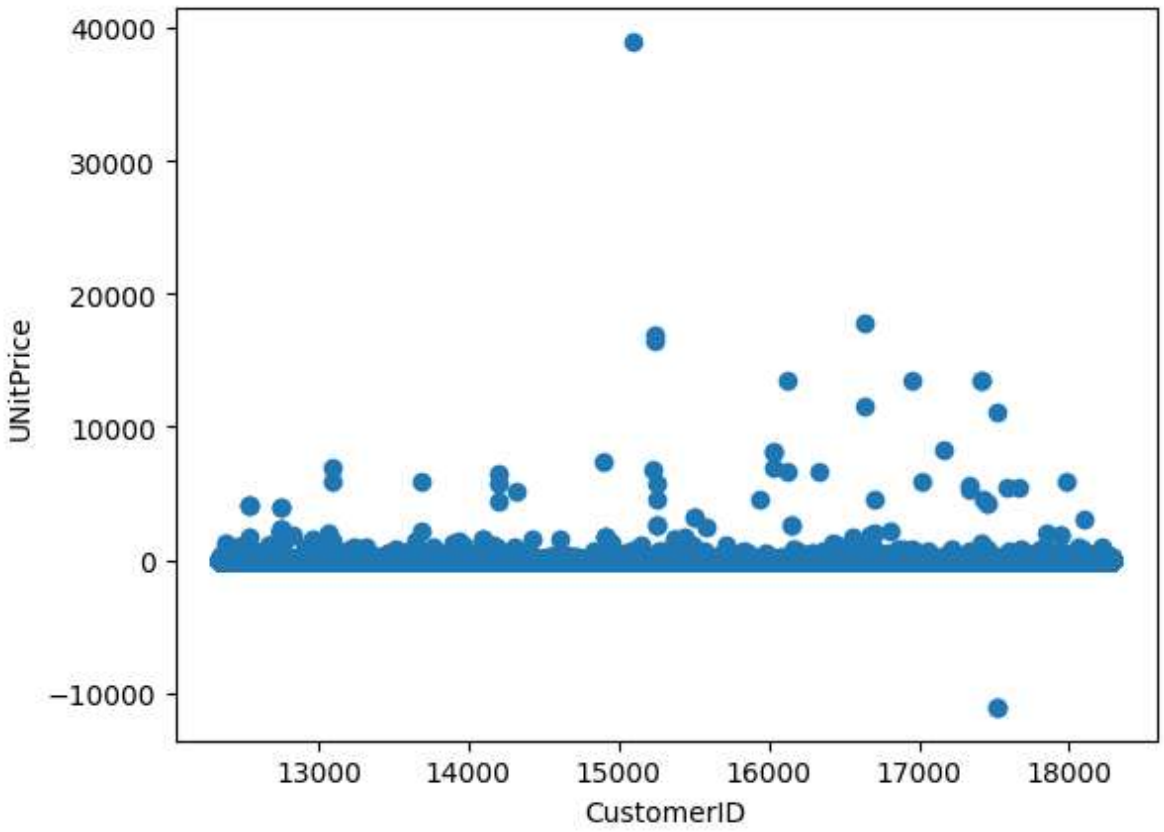
In [13]:
```
1 df.columns
```

Out[13]:
```
Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
       'UnitPrice', 'CustomerID', 'Country'],
      dtype='object')
```

In [14]:
```
plt.scatter(df["CustomerID"],df["UnitPrice"])
plt.xlabel("CustomerID")
plt.ylabel("UNitPrice")

```

Out[14]:
```
Text(0, 0.5, 'UNitPrice')
```



In [15]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[15]:
```
KMeans()
```

In [16]:
```
y_predicted=km.fit_predict(df[["CustomerID","UnitPrice"]])
y_predicted
```

```
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
```

Out[16]:
```
array([0, 0, 0, ..., 3, 3, 3])
```

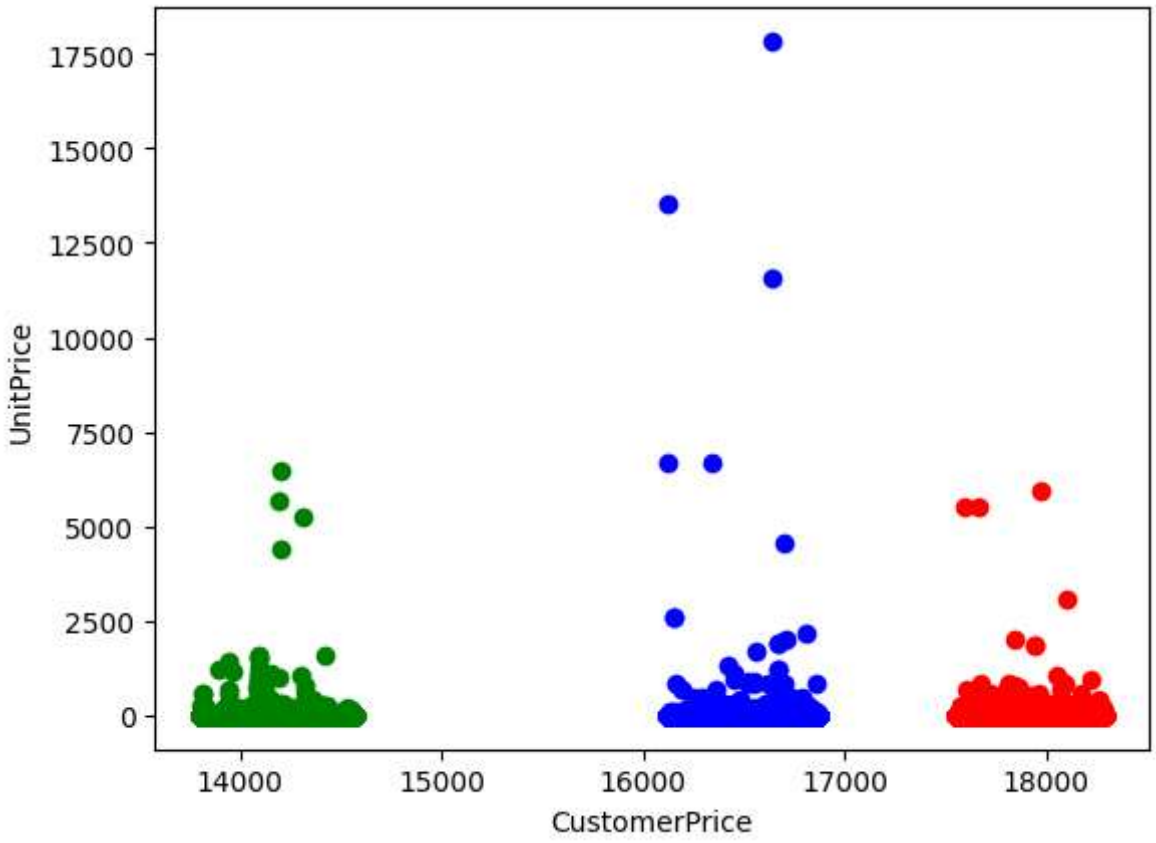
In [17]:
```
df["cluster"]=y_predicted
df.head()
```

Out[17]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cl
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 [19]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["UnitPrice"],color="red")
plt.scatter(df2["CustomerID"],df2["UnitPrice"],color="green")
plt.scatter(df3["CustomerID"],df3["UnitPrice"],color="blue")
plt.xlabel("CustomerPrice")
plt.ylabel("UnitPrice")
```

Out[19]: Text(0, 0.5, 'UnitPrice')



In [20]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["UnitPrice"]])
df["UnitPrice"]=scaler.transform(df[["UnitPrice"]])
df.head()
```

Out[20]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cl
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	0.221150	17850.0	United Kingdom	
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	0.221167	17850.0	United Kingdom	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	0.221154	17850.0	United Kingdom	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	0.221167	17850.0	United Kingdom	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	0.221167	17850.0	United Kingdom	

In [21]:
```
scaler.fit(df[["CustomerID"]])
df["CustomerID"]=scaler.transform(df[["CustomerID"]])
df.head()

```

Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cl
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	0.221150	0.926443	United Kingdom	
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	0.221154	0.926443	United Kingdom	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	

In [22]:
```
km=KMeans()
```

In [23]:
```
y_predicted=km.fit_predict(df[["CustomerID","UnitPrice"]])
y_predicted
```

```
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
```

Out[23]:
```
array([2, 2, 2, ..., 6, 6, 6])
```

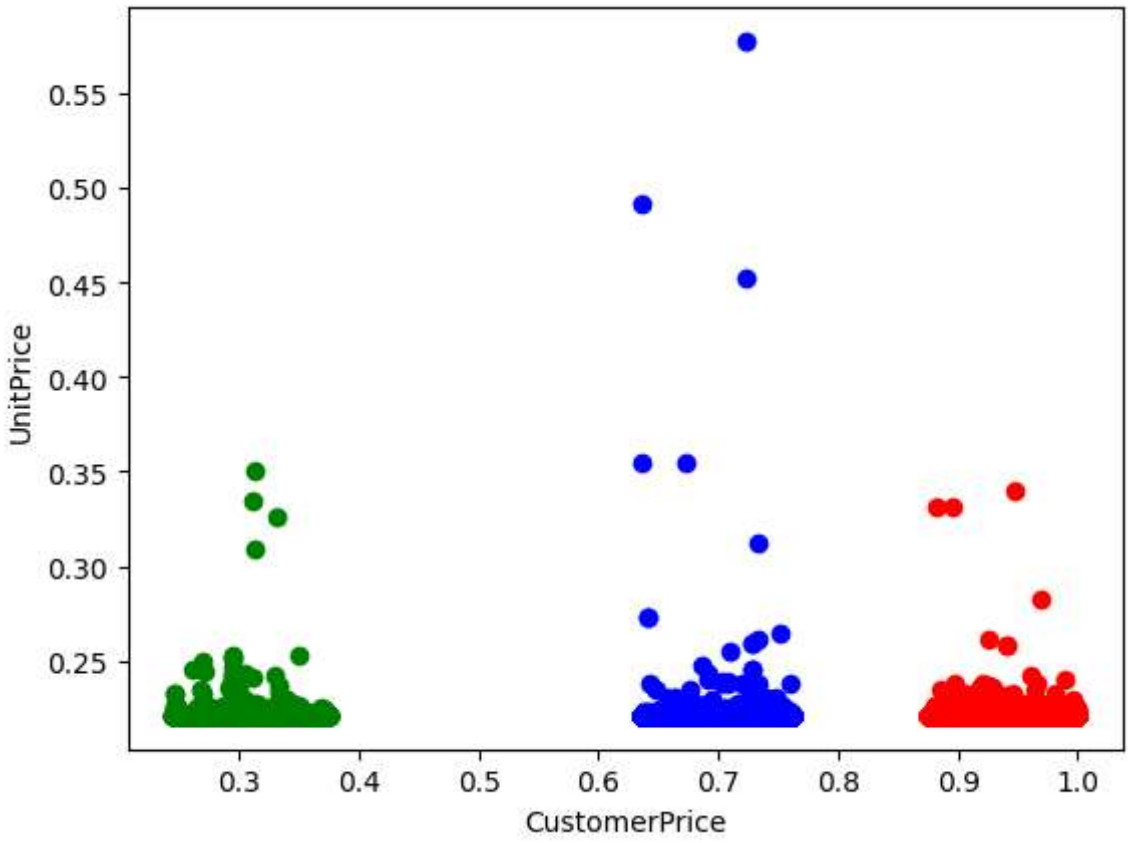
In [24]:
```
df["New Cluster"]=y_predicted
df.head()
```

Out[24]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cl
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	0.221150	0.926443	United Kingdom	
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	0.221154	0.926443	United Kingdom	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	

In [25]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["UnitPrice"],color="red")
plt.scatter(df2["CustomerID"],df2["UnitPrice"],color="green")
plt.scatter(df3["CustomerID"],df3["UnitPrice"],color="blue")
plt.xlabel("CustomerPrice")
plt.ylabel("UnitPrice")
```

Out[25]:
```
Text(0, 0.5, 'UnitPrice')
```

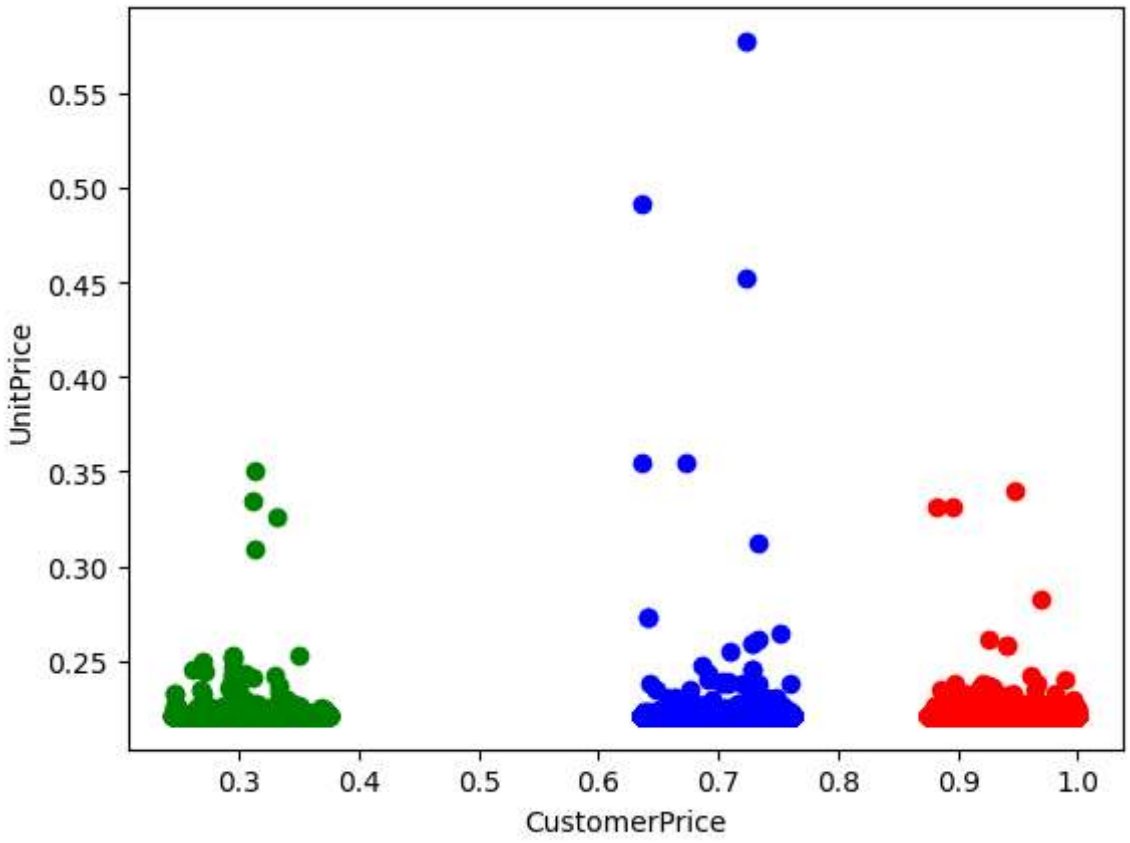


In [26]:
```
km.cluster_centers_
```

Out[26]:
```
array([[0.69989479, 0.22119809],
       [0.16600151, 0.2211844 ],
       [0.93299701, 0.2211783 ],
       [0.29865733, 0.2211874 ],
       [0.81775848, 0.22119948],
       [0.55360153, 0.22119718],
       [0.05166198, 0.22120282],
       [0.41800307, 0.22118756]])
```

In [27]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["UnitPrice"],color="red")
plt.scatter(df2["CustomerID"],df2["UnitPrice"],color="green")
plt.scatter(df3["CustomerID"],df3["UnitPrice"],color="blue")
plt.xlabel("CustomerPrice")
plt.ylabel("UnitPrice")
```

Out[27]:
```
Text(0, 0.5, 'UnitPrice')
```



In [28]:
```
k_rng=range(1,10)
sse=[]
```

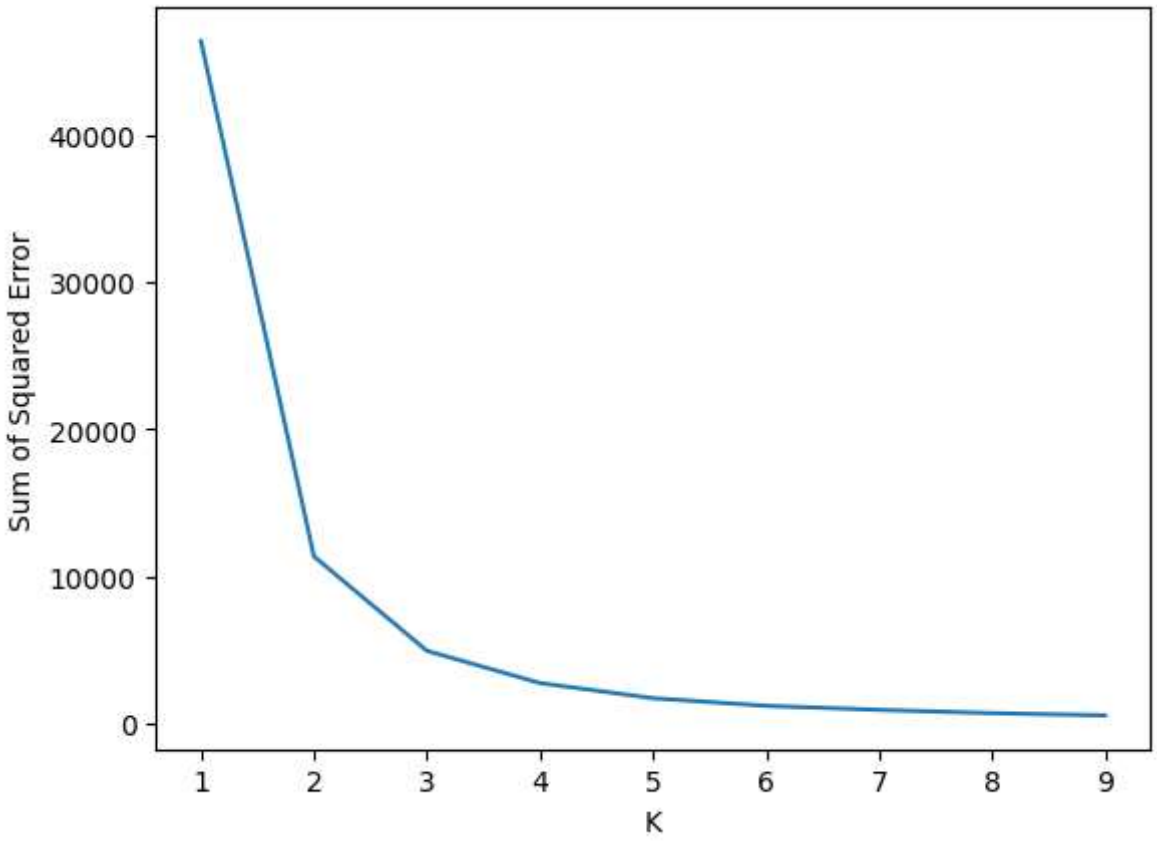
In [29]:

```python
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["CustomerID","UnitPrice"]])
 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\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
C:\Users\P. VIJAY KUMAR\AppData\Roaming\Python\Python310\site-packages\sklear
n\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will c
hange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr
ess the warning
  warnings.warn(
```

```
[46375.89020547898, 11337.109981610289, 4916.94569319009, 2724.5637818770265,
1696.0983656825579, 1179.5484782320732, 903.6409426351693, 678.329140993817,
529.8625992679783]
```

Out[29]: Text(0, 0.5, 'Sum of Squared Error')



CONCLUSION: The KMeans algorithm is best fitted to given dataset

In [ ]:
```
1
```