problem statement:To Predict the bestfit and to predict the online retail based on the given features

In [1]:
```
#importing libraries
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [3]:
```
df=pd.read_csv(r"C:\Users\pappu\Downloads\OnlineRetail.csv")
df
```

Out[3]:

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

2) Data cleaning and processing

In [4]:
```
1 df.head()
```

Out[4]:

	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 [5]:
```
1 df.tail()
```

Out[5]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cou
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

In [6]:
```
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 [8]:
```
1 df['InvoiceNo'].value_counts()
```

Out[8]:
```
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 [9]:
```
1 df['CustomerID'].value_counts()
```

Out[9]:
```
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 [10]:
```
1 df['Quantity'].value_counts()
```

Out[10]:
```
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 [11]:
```
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 [12]:
```
1 df.isnull().sum()
```

Out[12]:
```
InvoiceNo          0
StockCode          0
Description      1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID     135080
Country            0
dtype: int64
```

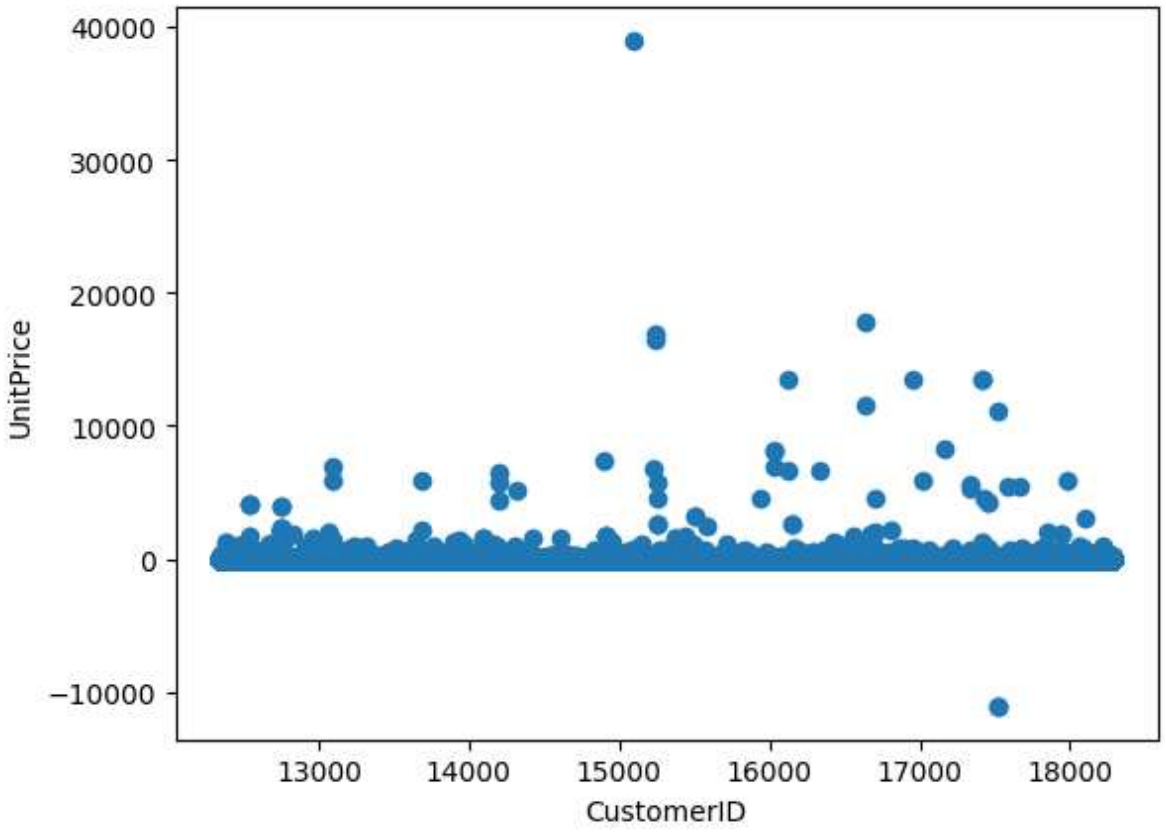
In [16]:
```
1 df.fillna(method='ffill',inplace=True)
2
```

In [17]:
```
1 df.isnull().sum()
```

Out[17]:
```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

3)Exploratory data analysis

In [18]:
```
1 plt.scatter(df["CustomerID"],df["UnitPrice"])
2 plt.xlabel("CustomerID")
3 plt.ylabel("UnitPrice")
4 plt.show()
```



4)Training our model

In [19]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[19]:
```
KMeans()
```

In [20]:
```
y_predicted=km.fit_predict(df[["CustomerID","UnitPrice"]])
y_predicted
```

```
C:\Users\teppa\AppData\Local\Programs\Python\Python310\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[20]:
```
array([4, 4, 4, ..., 2, 2, 2])
```

In [21]:
```
df["cluster"]=y_predicted
df.head()
```

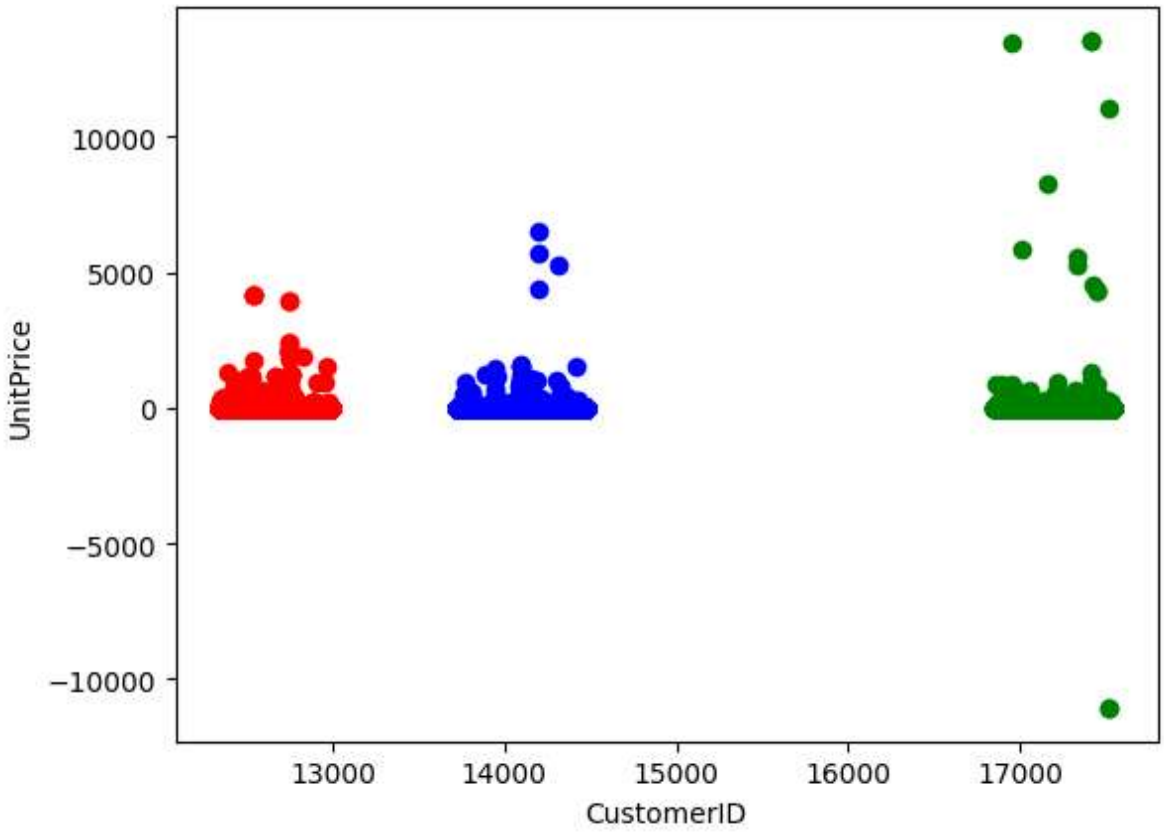
Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cl
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	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 [22]:

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

Out[22]: Text(0, 0.5, 'UnitPrice')



In [23]:

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

Out[23]:

	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	0.926443	United Kingdom	
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	0.926443	United Kingdom	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	

In [24]:
```
scaler=MinMaxScaler()
scaler.fit(df[["UnitPrice"]])
df["UnitPrice"]=scaler.transform(df[["UnitPrice"]])
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]:
```
km=KMeans()

```

In [26]:
```
y_predicted=km.fit_predict(df[["CustomerID","UnitPrice"]])
y_predicted
```

```
C:\Users\teppa\AppData\Local\Programs\Python\Python310\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[26]:
```
array([0, 0, 0, ..., 3, 3, 3])
```

In [27]:

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

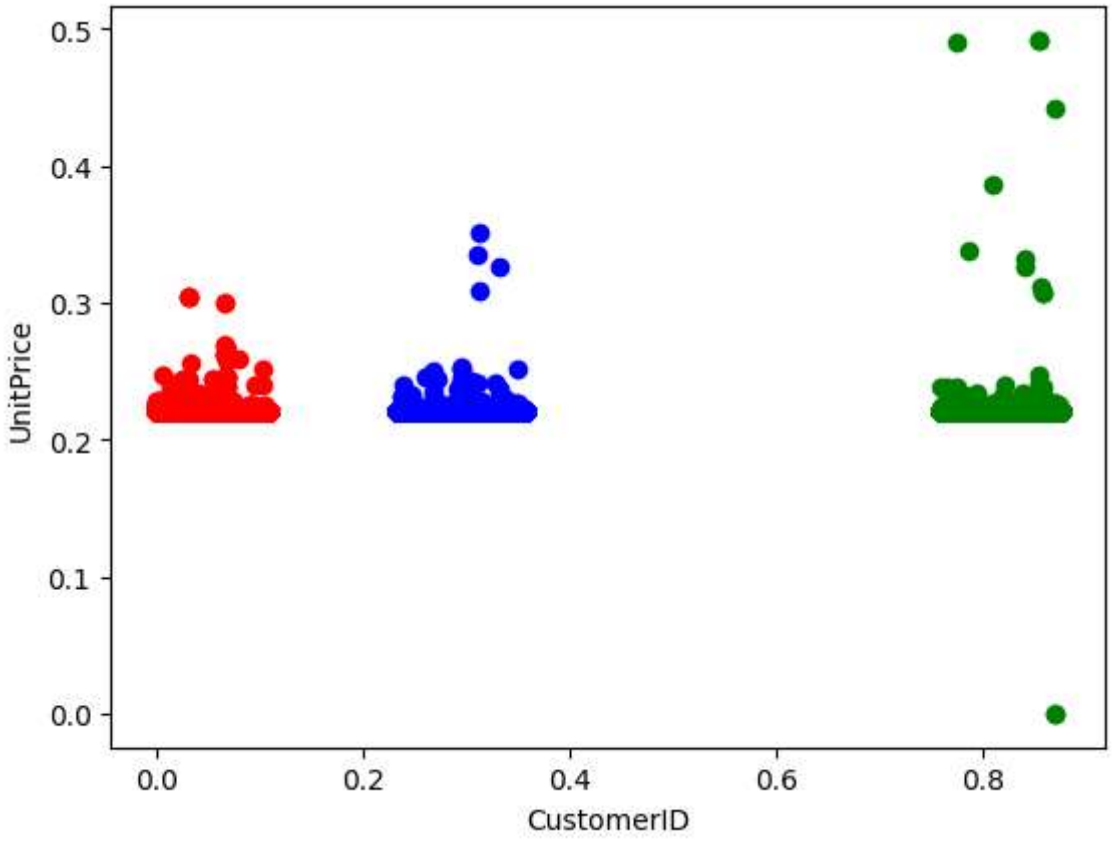
Out[27]:

	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 [28]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["UnitPrice"],color='green')
plt.scatter(df2["CustomerID"],df2["UnitPrice"],color='blue')
plt.scatter(df3["CustomerID"],df3["UnitPrice"],color='red')
plt.xlabel("CustomerID")
plt.ylabel("UnitPrice")

```

Out[28]:
```
Text(0, 0.5, 'UnitPrice')
```

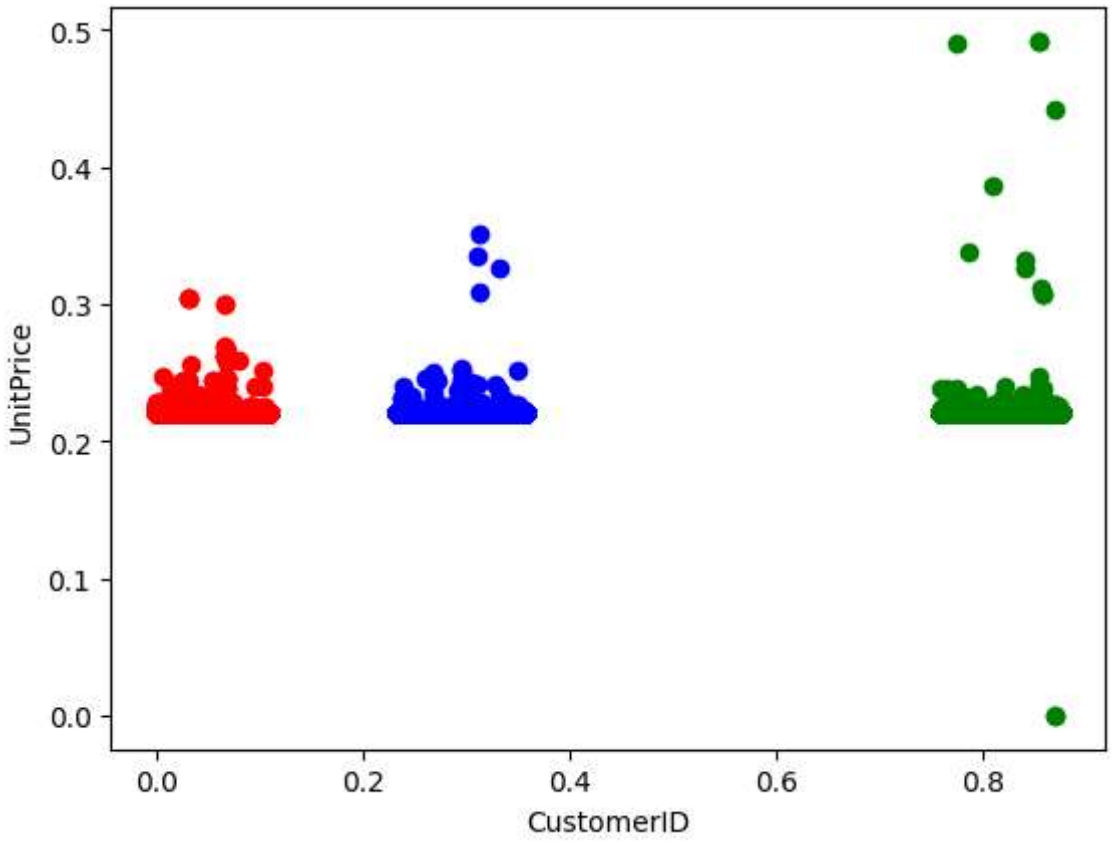


In [29]:
```
km.cluster_centers_
```

Out[29]:
```
array([[0.9328779 , 0.22117824],
       [0.29844314, 0.2211874 ],
       [0.69961249, 0.22119799],
       [0.05156814, 0.22120288],
       [0.55335879, 0.2211973 ],
       [0.81755206, 0.22119956],
       [0.41790724, 0.22118757],
       [0.16577916, 0.22118437]])
```

In [30]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["CustomerID"],df1["UnitPrice"],color='green')
plt.scatter(df2["CustomerID"],df2["UnitPrice"],color='blue')
plt.scatter(df3["CustomerID"],df3["UnitPrice"],color='red')
plt.xlabel("CustomerID")
plt.ylabel("UnitPrice")
```

Out[30]:
```
Text(0, 0.5, 'UnitPrice')
```



In [31]:
```
k_rng=range(1,10)
sse=[]
```

In [32]:

```python
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["CustomerID","UnitPrice"]])
    sse.append(km.inertia_)
sse
```

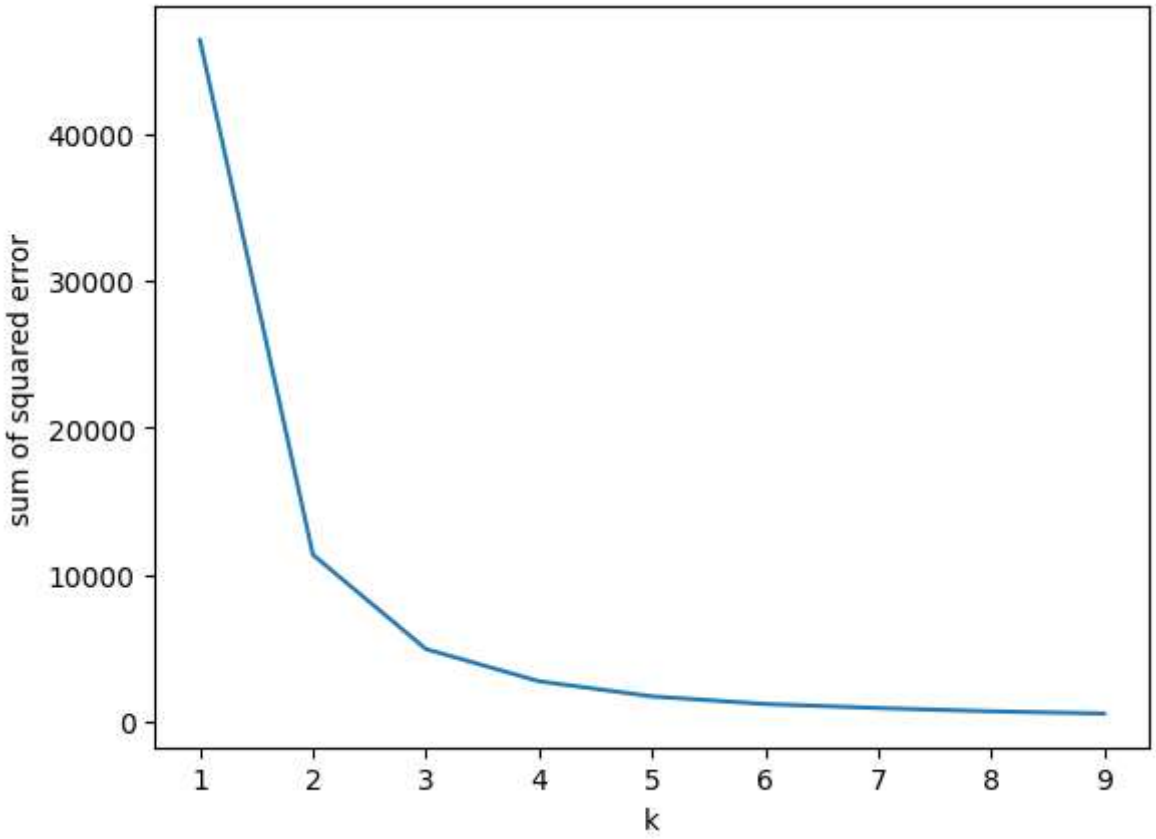
```
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\teppa\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
```

Out[32]:
```
[46375.89020547866,
 11337.109981610118,
 4919.481330963172,
 2724.56378187714,
 1696.1293309604941,
 1179.470634938091,
 903.5629718186494,
 678.596235871086,
 530.7293085521068]
```

Elbow Graph

In [33]:
```
1 plt.plot(k_rng,sse)
2 plt.xlabel("k")
3 plt.ylabel("sum of squared error")
4
```

Out[33]:
```
Text(0, 0.5, 'sum of squared error')
```



conclusion: The given data is "Online retail".For the above data set we have used K-means dataset and doneClustering based on given data set.If the k value is low the error rate

is more.if k value is high the error

In [ ]:
```
1
```