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

1)Data collection

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

In [2]:
```
#Reading data
df=pd.read_csv(r"C:\Users\Jayadeep\Documents\online retail1.csv")
df
```

Out[2]:

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

541909 rows × 8 columns

2)Data cleaning and processing

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

Out[3]:

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

Out[4]:

	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	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	09-12-2011 12:50	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	09-12-2011 12:50	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	09-12-2011 12:50	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	09-12-2011 12:50	4.95	12680.0	France

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

```
Out[5]: 573585     1114
        581219      749
        581492      731
        580729      721
        558475      705
                   ...
        554023        1
        554022        1
        554021        1
        554020        1
        C558901       1
        Name: InvoiceNo, Length: 25900, dtype: int64
```

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

```
Out[6]: 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: CustomerID, Length: 4372, dtype: int64
```

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

Out[7]:
```
 1        148227
 2         81829
 12        61063
 6         40868
 4         38484
           ...  
-472           1
-161           1
-1206          1
-272           1
-80995         1
Name: Quantity, Length: 722, dtype: int64
```

In [8]:
```
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 [9]:
```
df.isnull().sum()
```

Out[9]:
```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

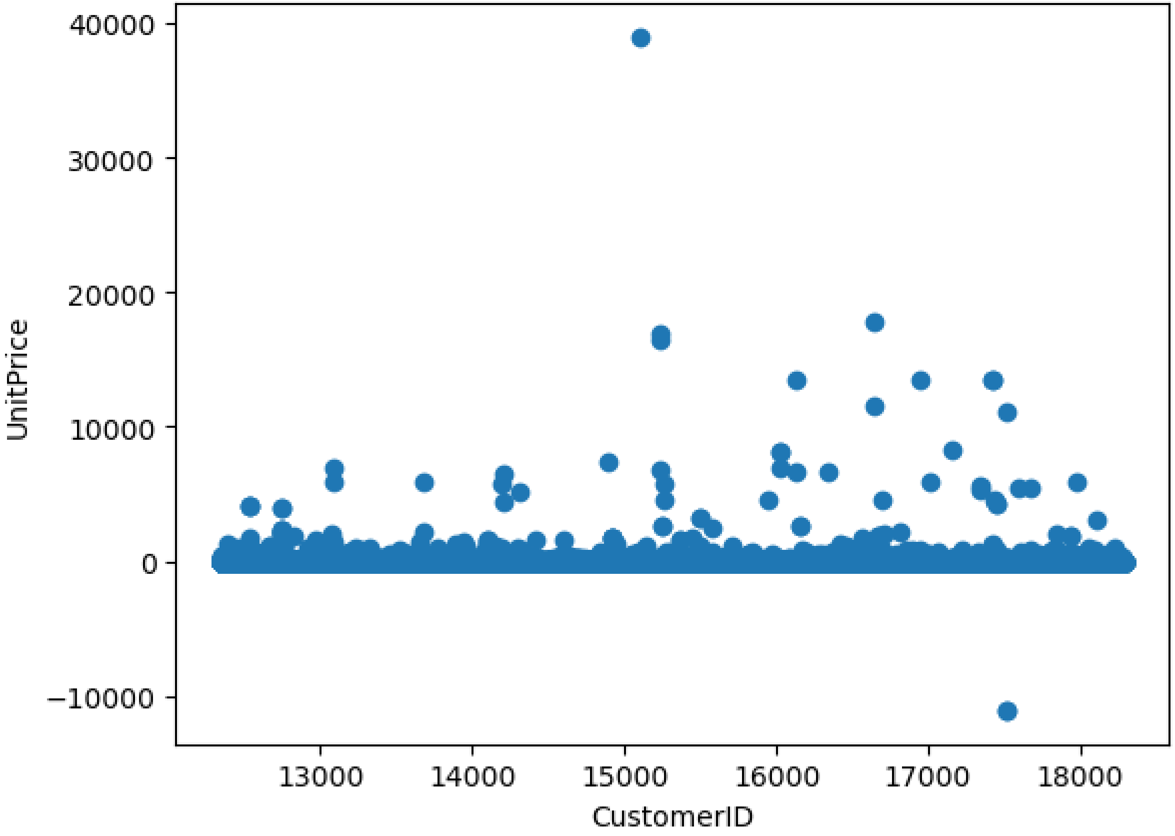
In [10]:
```
df.fillna(method='ffill',inplace=True)
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

3)Exploratory data analysis

In [12]:
```
plt.scatter(df["CustomerID"],df["UnitPrice"])
plt.xlabel("CustomerID")
plt.ylabel("UnitPrice")
plt.show()
```



4)Training our model

In [13]:

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

Out[13]:

```
KMeans()
```

In [14]:

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

Out[14]:

```
array([7, 7, 7, ..., 6, 6, 6])
```

In [15]:

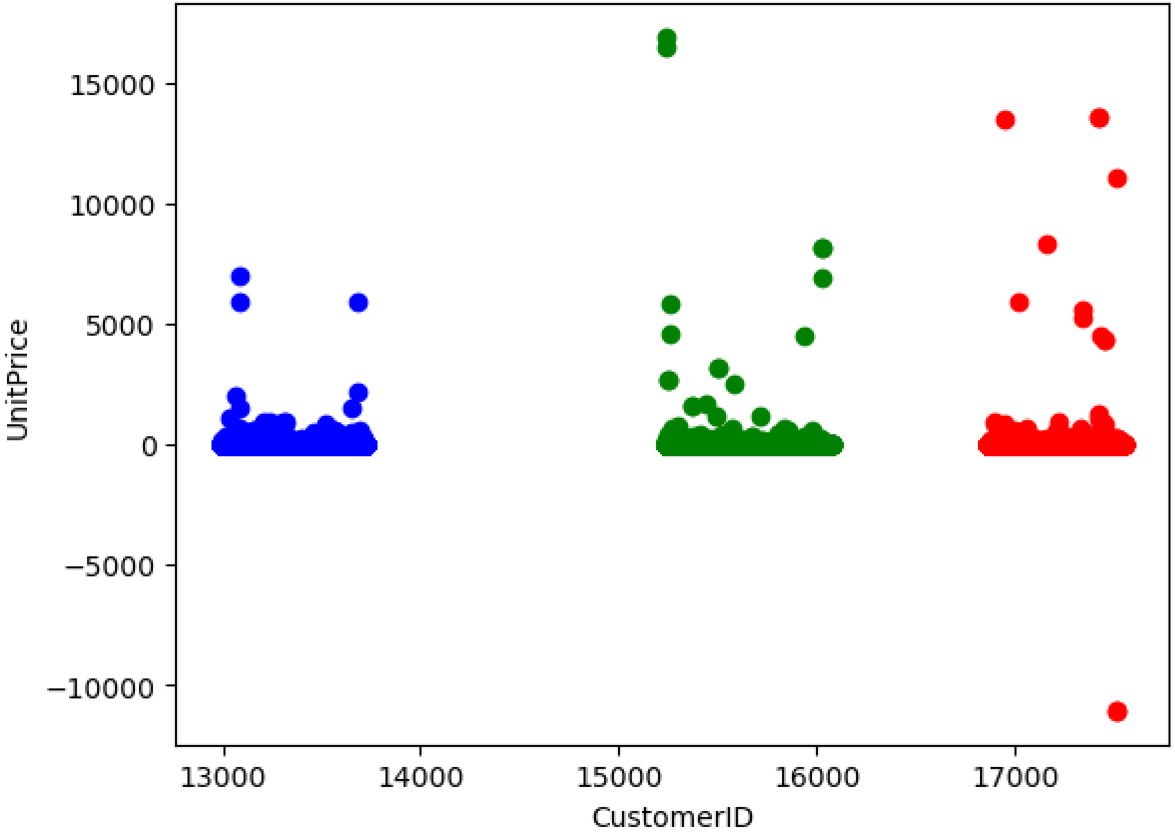
```
df["cluster"]=y_predicted
df.head()
```

Out[15]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom	7
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom	7
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom	7
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom	7
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom	7

In [16]:
```
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[16]:
```
Text(0, 0.5, 'UnitPrice')
```



In [17]:

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

Out[17]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	0.926443	United Kingdom	7
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	7
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	0.926443	United Kingdom	7
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	7
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	0.926443	United Kingdom	7

In [18]:

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

Out[18]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	0.221150	0.926443	United Kingdom	7
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	0.221154	0.926443	United Kingdom	7
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7

In [19]:

```
km=KMeans()
```

In [20]:

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

Out[20]:

```
array([1, 1, 1, ..., 5, 5, 5])
```

In [21]:

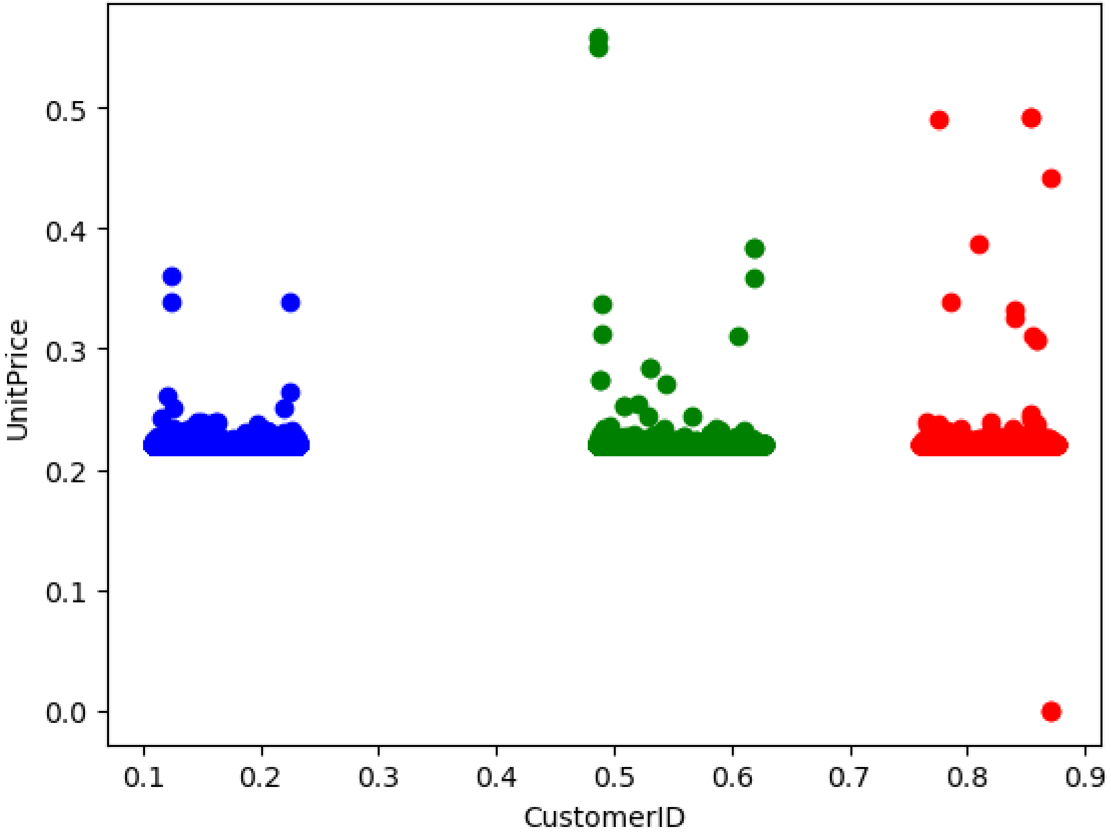
```
df["New Cluster"]=y_predicted
df.head()
```

Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	cluster	New Cluster
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	0.221150	0.926443	United Kingdom	7	1
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7	1
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	0.221154	0.926443	United Kingdom	7	1
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7	1
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	0.221167	0.926443	United Kingdom	7	1

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')
```

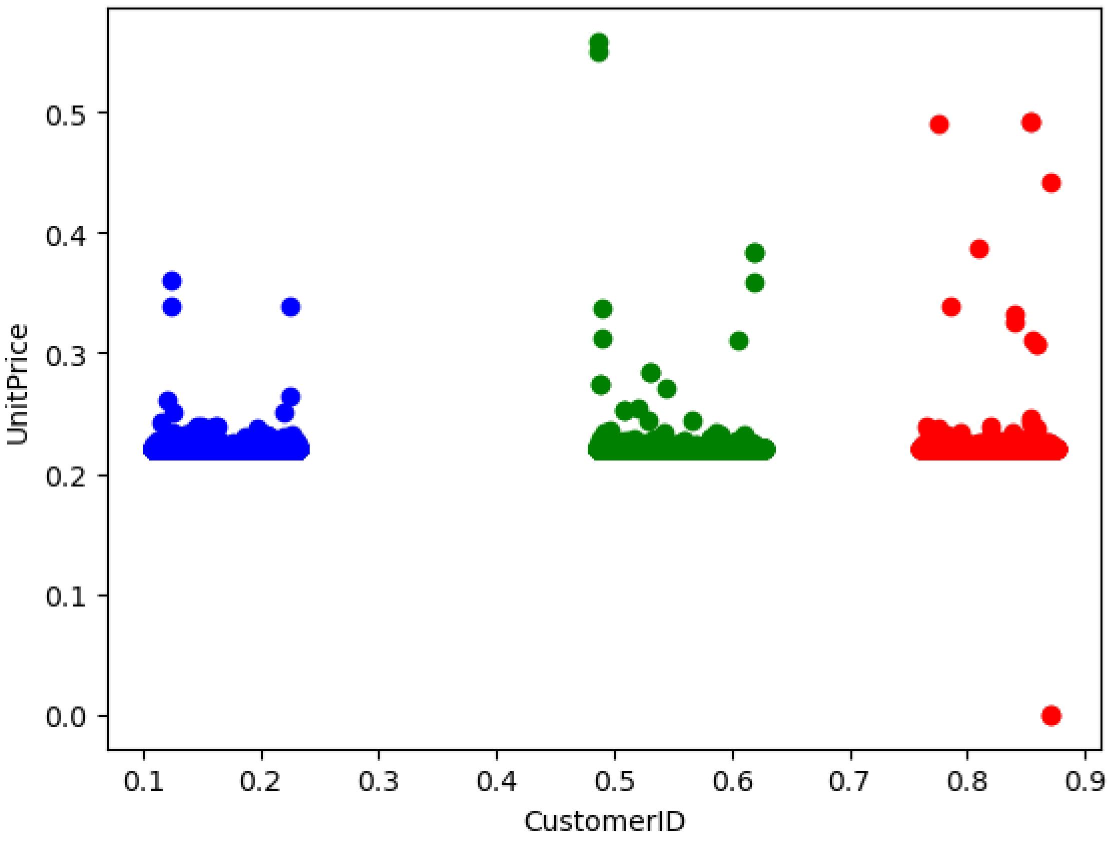


In [23]:
```
km.cluster_centers_
```

Out[23]:
```
array([[0.16573636, 0.22118441],
       [0.93308721, 0.22117835],
       [0.55502897, 0.22118727],
       [0.4185919 , 0.22119564],
       [0.70124399, 0.22119847],
       [0.05156814, 0.22120288],
       [0.81857738, 0.22119924],
       [0.29849213, 0.22118735]])
```

In [24]:
```
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[24]:
```
Text(0, 0.5, 'UnitPrice')
```



In [25]:

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

In [28]:

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

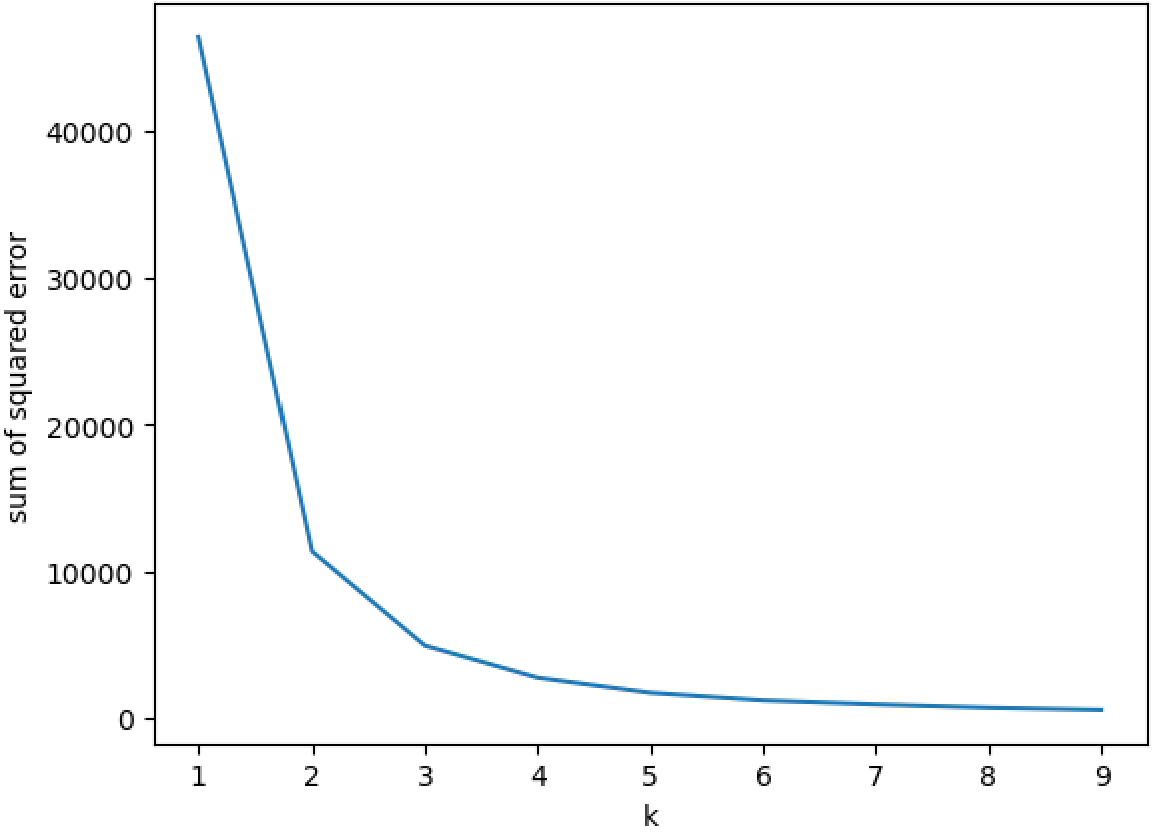
Out[28]:

```
[46375.89020547945,
 11337.10998161026,
 4916.917350291193,
 2724.563781877091,
 1696.1222875898384,
 1179.518375472868,
 903.5755836413746,
 678.5741459311167,
 529.7715143287168,
 46375.89020547945,
 11337.11049629344,
 4922.75156740312,
 2724.5637818770924,
 1696.560227864234,
 1179.4708386922298,
 913.7776872660106,
 678.3061613175081,
 529.8266116978539]
```

Elbow Graph

In [27]:
```
plt.plot(k_rng,sse)
plt.xlabel("k")
plt.ylabel("sum of squared error")
```

Out[27]:
```
Text(0, 0.5, 'sum of squared error')
```



conclusion: The given data is "Online retail".For this data set we have used K-means dataset and done Clustering based on given data set.If the k value is low the error rate is more,if k value is high the error is low.Therefore KMeans Clustering is the Bestfit for this Dataset