In [3]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
#Heading data
df = pd.read_csv('Orders.csv')
df
```

Out[4]:

	OrderID	InvoiceDate	CustomerID	OrderVolume	ShipMode	Segment	Category	PaymentTerm	Discount	ExistingPurchaseOrder	FirstCustomerOrder	CashInDate
0	O-100	2011-01-03	1024	276.1	Same Day	Corporate	Office Supplies	30	5%	Yes	Yes	2011-02-18
1	O-101	2011-01-03	1024	35.88	Same Day	Corporate	Office Supplies	30	5%	Yes	No	2011-02-18
2	O-102	2011-01-05	1006	66.12	Second Class	Consumer	Office Supplies	30	3%	Yes	Yes	2011-02-06
3	O-104	2011-01-06	1023	408.3	Standard Class	Consumer	Office Supplies	30	5%	No	Yes	2011-02-17
4	O-105	2011-01-06	1009	314.22	Standard Class	Consumer	Technology	45	3%	Yes	Yes	2011-02-10
...	...	...	...	...	...	...	...	...	...	...	...	...
37637	O-48951	2014/30/11	1029	50.09	Standard Class	Corporate	Office Supplies	14	3%	No	No	2014-12-31
37638	O-48953	2014/30/11	1010	21.12	Same Day	Consumer	Office Supplies	14	5%	Yes	No	2014-12-31
37639	O-48960	2014-11-30	1031	33.57	First Class	Consumer	Office Supplies	14	5%	Yes	No	2014-12-16
37640	O-48972	2014-11-30	1031	20.72	Second Class	Consumer	Office Supplies	30	3%	Yes	No	2014-12-15
37641	O-48977	2014-11-30	1000	1.2	Standard Class	Consumer	Office Supplies	14	0%	Yes	No	2014-12-30

37642 rows × 12 columns

In [5]:
```
df['value'] = np.arange(len(df))
df
```

Out[5]:

	OrderID	InvoiceDate	CustomerID	OrderVolume	ShipMode	Segment	Category	PaymentTerm	Discount	ExistingPurchaseOrder	FirstCustomerOrder	CashInDate	value
0	O-100	2011-01-03	1024	276.1	Same Day	Corporate	Office Supplies	30	5%	Yes	Yes	2011-02-18	0
1	O-101	2011-01-03	1024	35.88	Same Day	Corporate	Office Supplies	30	5%	Yes	No	2011-02-18	1
2	O-102	2011-01-05	1006	66.12	Second Class	Consumer	Office Supplies	30	3%	Yes	Yes	2011-02-06	2
3	O-104	2011-01-06	1023	408.3	Standard Class	Consumer	Office Supplies	30	5%	No	Yes	2011-02-17	3
4	O-105	2011-01-06	1009	314.22	Standard Class	Consumer	Technology	45	3%	Yes	Yes	2011-02-10	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
37637	O-48951	2014/30/11	1029	50.09	Standard Class	Corporate	Office Supplies	14	3%	No	No	2014-12-31	37637
37638	O-48953	2014/30/11	1010	21.12	Same Day	Consumer	Office Supplies	14	5%	Yes	No	2014-12-31	37638
37639	O-48960	2014-11-30	1031	33.57	First Class	Consumer	Office Supplies	14	5%	Yes	No	2014-12-16	37639
37640	O-48972	2014-11-30	1031	20.72	Second Class	Consumer	Office Supplies	30	3%	Yes	No	2014-12-15	37640
37641	O-48977	2014-11-30	1000	1.2	Standard Class	Consumer	Office Supplies	14	0%	Yes	No	2014-12-30	37641

37642 rows × 13 columns

In [17]:
```
df['OrderVolume'].value_counts()
```

Out[17]:
```
12.96      47
25.92      39
32.4       32
15.55      29
19.44      29
           ..
346.29      1
521.96      1
258.07      1
461.48      1
1000.91     1
Name: OrderVolume, Length: 16952, dtype: int64
```

In [18]:
```
df['OrderVolume'] = pd.to_numeric(df['OrderVolume'],errors = 'coerce')
df.OrderVolume.dtypes
```

Out[18]:
```
dtype('float64')
```

In [20]:
```
fill = df['OrderVolume'].mean()
fill
```

Out[20]:
```
245.15100329559627
```

In [28]:
```
df['OrderVolume'] = df['OrderVolume'].fillna(df['OrderVolume'].mean())
df['OrderVolume'].isnull().sum()
```

Out[28]:
```
0
```

In [29]:
```
final = df[["InvoiceDate","value","OrderVolume"]]
final
```

Out[29]:

	InvoiceDate	value	OrderVolume
0	2011-01-03	0	276.10
1	2011-01-03	1	35.88
2	2011-01-05	2	66.12
3	2011-01-06	3	408.30
4	2011-01-06	4	314.22
...	...	...	...
37637	2014/30/11	37637	50.09
37638	2014/30/11	37638	21.12
37639	2014-11-30	37639	33.57
37640	2014-11-30	37640	20.72
37641	2014-11-30	37641	1.20

37642 rows × 3 columns

In [30]:
```
index = final.set_index('InvoiceDate')
index
```

Out[30]:

InvoiceDate	value	OrderVolume
2011-01-03	0	276.10
2011-01-03	1	35.88
2011-01-05	2	66.12
2011-01-06	3	408.30
2011-01-06	4	314.22
...	...	...
2014/30/11	37637	50.09
2014/30/11	37638	21.12
2014-11-30	37639	33.57
2014-11-30	37640	20.72
2014-11-30	37641	1.20

37642 rows × 2 columns

In [32]:
```
index.isnull()
```

Out[32]:

InvoiceDate	value	OrderVolume
2011-01-03	False	False
2011-01-03	False	False
2011-01-05	False	False
2011-01-06	False	False
2011-01-06	False	False
...	...	...
2014/30/11	False	False
2014/30/11	False	False
2014-11-30	False	False
2014-11-30	False	False
2014-11-30	False	False

37642 rows × 2 columns

In [33]:
```
import numpy as np
x,y = index['value'],index['OrderVolume']
x,y = np.array(x),np.array(y)
x,y= x.reshape(-1,1), y.reshape(-1,1)
print(x)
```

Out[33]:
```
[[    0]
 [    1]
 [    2]
 ...
 [37639]
 [37640]
 [37641]]
```

In [34]:
```
print(y)
```

Out[34]:
```
[[276.1 ]
 [ 35.88]
 [ 66.12]
 ...
 [ 33.57]
 [ 20.72]
 [  1.2 ]]
```

In [35]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size = 20,random_state = 1)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(xtrain,ytrain)
```

Out[36]:
```
LinearRegression()
```

In [37]:
```
reg.predict(xtest)
```

Out[37]:
```
array([[242.95180734],
       [246.59592987],
       [247.82825674],
       [245.40623672],
       [248.31208201],
       [242.64218232],
       [242.42727748],
       [247.25993567],
       [242.78281574],
       [243.22227187],
       [245.02176153],
       [244.19667355],
       [242.70159946],
       [247.41812416],
       [242.45293488],
       [247.01860178],
       [246.62467387],
       [248.70369496],
       [242.7700835 ],
       [248.20964532]])
```

In [38]:
```
num = 20
trange = pd.date_range('2014-12-1', periods=num, freq='d')
trange
```

Out[38]:
```
DatetimeIndex(['2014-12-01', '2014-12-02', '2014-12-03', '2014-12-04',
               '2014-12-05', '2014-12-06', '2014-12-07', '2014-12-08',
               '2014-12-09', '2014-12-10', '2014-12-11', '2014-12-12',
               '2014-12-13', '2014-12-14', '2014-12-15', '2014-12-16',
               '2014-12-17', '2014-12-18', '2014-12-19', '2014-12-20'],
              dtype='datetime64[ns]', freq='D')
```

In [39]:
```
Forecast = reg.predict(xtest)
print(Forecast)
```

Out[39]:
```
[[242.95180734]
 [246.59592987]
 [247.82825674]
 [245.40623672]
 [248.31208201]
 [242.64218232]
 [242.42727748]
 [247.25993567]
 [242.78281574]
 [243.22227187]
 [245.02176153]
 [244.19667355]
 [242.70159946]
 [247.41812416]
 [242.45293488]
 [247.01860178]
 [246.62467387]
 [248.70369496]
 [242.7700835 ]
 [248.20964532]]
```

In [40]:
```
Predict_df = pd.DataFrame(Forecast, index=trange)
Predict_df.columns = ['forecast']
Predict_df
```

Out[40]:

	forecast
2014-12-01	242.951807
2014-12-02	246.595930
2014-12-03	247.828257
2014-12-04	245.406237
2014-12-05	248.312082
2014-12-06	242.642182
2014-12-07	242.427277
2014-12-08	247.259936
2014-12-09	242.782816
2014-12-10	243.222271
2014-12-11	245.021762
2014-12-12	244.196674
2014-12-13	242.701599
2014-12-14	247.418124
2014-12-15	242.452935
2014-12-16	247.018602
2014-12-17	246.624674
2014-12-18	248.703695
2014-12-19	242.770084
2014-12-20	248.209645

In [ ]:
```

```