```
import pandas as pd
from pandas import read_excel
df = pd.DataFrame()
```

```
df = pd.read_excel("onlineretail.xlsx", index_col =
'InvoiceDate',parse_dates =True)
```

```
df.head()
```

```
                     SUM of Quantity  CustomerID
InvoiceDate
2010-01-12 08:26:00               40     17850.0
2010-01-12 08:28:00               12     13047.0
2010-01-12 08:34:00               98     12583.0
2010-01-12 08:35:00                3     13748.0
2010-01-12 09:00:00               80     15100.0
```

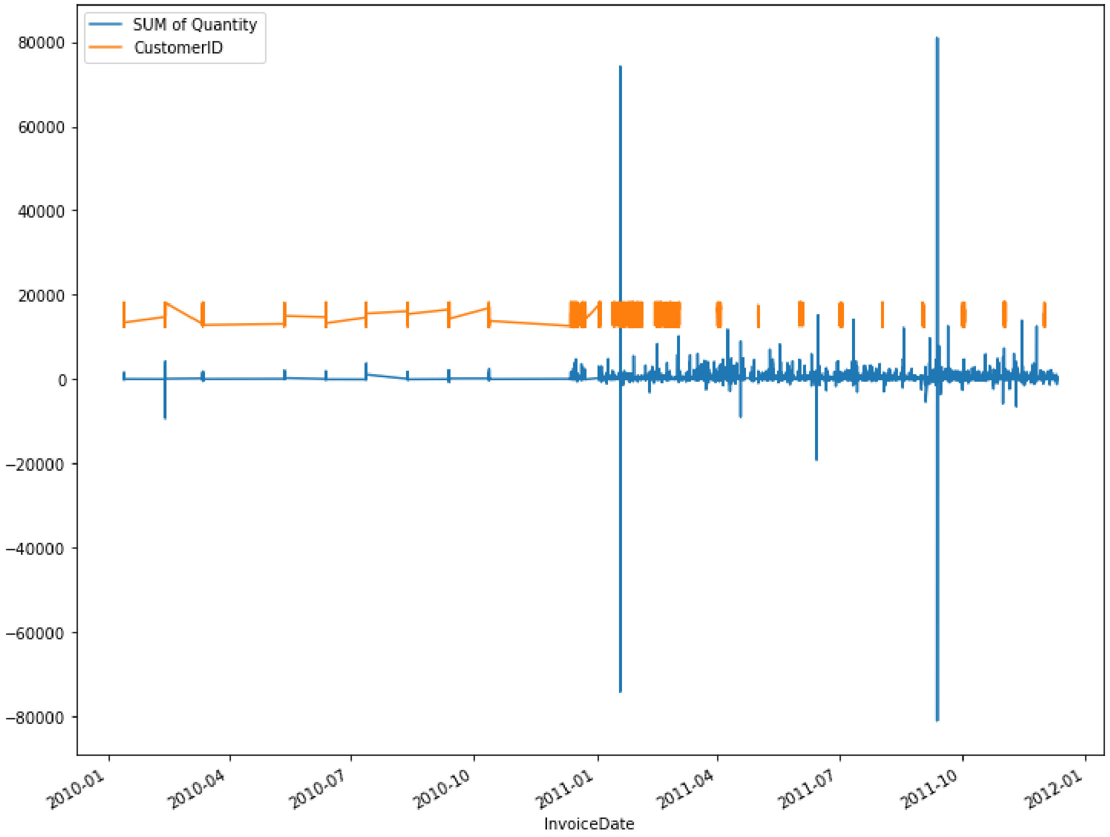
```
df.tail()
```

```
                     SUM of Quantity  CustomerID
InvoiceDate
2011-09-12 12:21:00               18         NaN
2011-09-12 12:23:00               76         NaN
2011-09-12 12:25:00              120         NaN
2011-09-12 12:31:00              278         NaN
2011-09-12 12:49:00               66         NaN
```

```
import matplotlib.pyplot as plt
```

```
df.columns = ['SUM of Quantity','CustomerID']
df.plot(figsize=(12,10))
```

```
<AxesSubplot:xlabel='InvoiceDate'>
```



```
df['Quantity_LastMonth'] = df['SUM of Quantity'].shift(+1)
df['Quantity_2Monthsback'] = df['SUM of Quantity'].shift(+2)
df['Quantity_3Month'] = df['SUM of Quantity'].shift(+3)
df
```

```
                     SUM of Quantity  CustomerID
Quantity_LastMonth  \
InvoiceDate

2010-01-12 08:26:00               40     17850.0                 NaN

2010-01-12 08:28:00               12     13047.0                40.0

2010-01-12 08:34:00               98     12583.0                12.0

2010-01-12 08:35:00                3     13748.0                98.0

2010-01-12 09:00:00               80     15100.0                 3.0

...                              ...         ...                 ...

2011-09-12 12:21:00               18         NaN               104.0

2011-09-12 12:23:00               76         NaN                18.0
```

```
2011-09-12 12:25:00              120         NaN                76.0

2011-09-12 12:31:00              278         NaN               120.0

2011-09-12 12:49:00               66         NaN               278.0


                     Quantity_2Monthsback  Quantity_3Month
InvoiceDate
2010-01-12 08:26:00                   NaN              NaN
2010-01-12 08:28:00                   NaN              NaN
2010-01-12 08:34:00                  40.0              NaN
2010-01-12 08:35:00                  12.0             40.0
2010-01-12 09:00:00                  98.0             12.0
...                                   ...              ...
2011-09-12 12:21:00                 476.0             60.0
2011-09-12 12:23:00                 104.0            476.0
2011-09-12 12:25:00                  18.0            104.0
2011-09-12 12:31:00                  76.0             18.0
2011-09-12 12:49:00                 120.0             76.0

[21304 rows x 5 columns]
```

```
df = df.dropna()
df
```

```
                     SUM of Quantity  CustomerID
Quantity_LastMonth  \
InvoiceDate

2010-01-12 08:35:00                3     13748.0                98.0

2010-01-12 09:00:00               80     15100.0                 3.0

2010-01-12 09:01:00               12     15291.0                80.0

2010-01-12 09:02:00               88     14688.0                12.0

2010-01-12 09:09:00               32     17809.0                88.0

...                              ...         ...                 ...

2011-07-03 12:06:00               -2     13436.0               -18.0

2011-07-03 12:12:00                2     15520.0                -2.0

2011-07-03 12:17:00              176     13298.0                 2.0
```

```
2011-07-03 12:26:00              473     14569.0               176.0

2011-07-03 12:31:00               44     12713.0               473.0
```

```
                     Quantity_2Monthsback  Quantity_3Month
InvoiceDate
2010-01-12 08:35:00                  12.0             40.0
2010-01-12 09:00:00                  98.0             12.0
2010-01-12 09:01:00                   3.0             98.0
2010-01-12 09:02:00                  80.0              3.0
2010-01-12 09:09:00                  12.0             80.0
...                                   ...              ...
2011-07-03 12:06:00                 130.0            204.0
2011-07-03 12:12:00                 -18.0            130.0
2011-07-03 12:17:00                  -2.0            -18.0
2011-07-03 12:26:00                   2.0             -2.0
2011-07-03 12:31:00                 176.0              2.0

[4369 rows x 5 columns]
```

```
from sklearn.linear_model import LinearRegression
lin = LinearRegression()
```

```
from sklearn.ensemble import RandomForestRegressor
model=RandomForestRegressor(n_estimators=100,max_features=3,
random_state=1)
```

```
import numpy as np
x1,x2,x3,y=df['Quantity_LastMonth'],df['Quantity_2Monthsback'],df['Qua
ntity_3Month'],df['SUM of Quantity']
x1,x2,x3,y=np.array(x1),np.array(x2),np.array(x3),np.array(y)
x1,x2,x3,y=x1.reshape(-1,1),x2.reshape(-1,1),x3.reshape(-
1,1),y.reshape(-1,1)
final_x=np.concatenate((x1,x2,x3),axis=1)
print(final_x)
```

```
[[ 98.  12.  40.]
 [  3.  98.  12.]
 [ 80.   3.  98.]
 ...
 [  2.  -2. -18.]
 [176.   2.  -2.]
 [473. 176.   2.]]
```

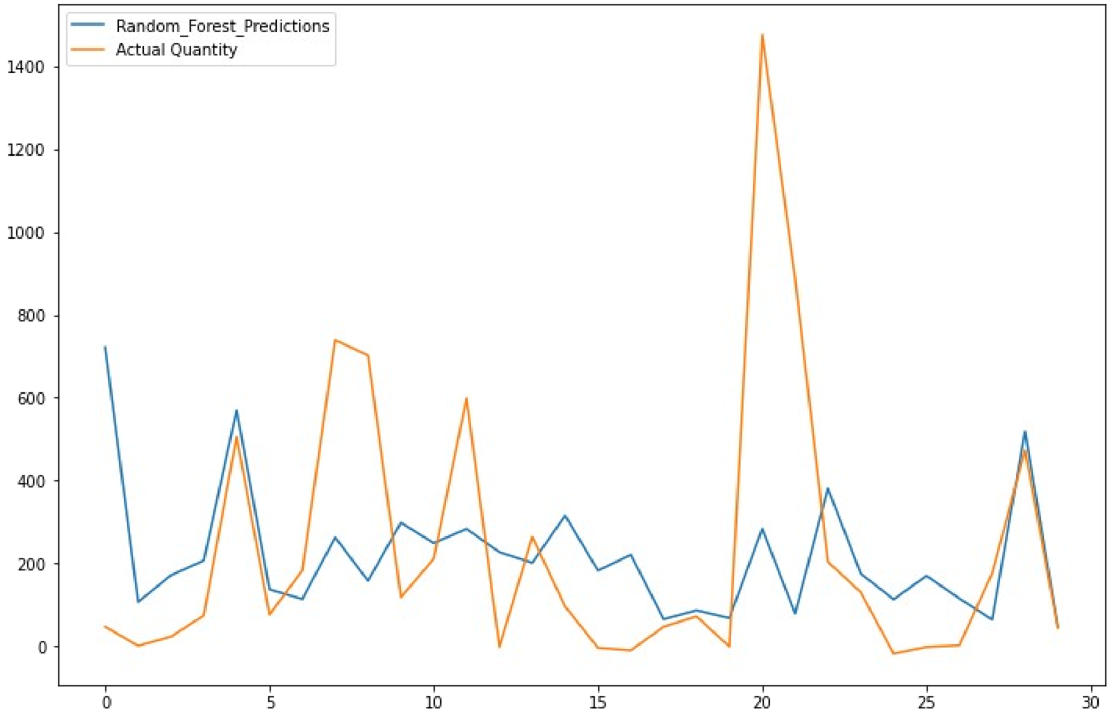
```
X_train,X_test,y_train,y_test=final_x[:-30],final_x[-30:],y[:-30],y[-
30:]
```

```
model.fit(X_train,y_train)
lin.fit(X_train,y_train)
```

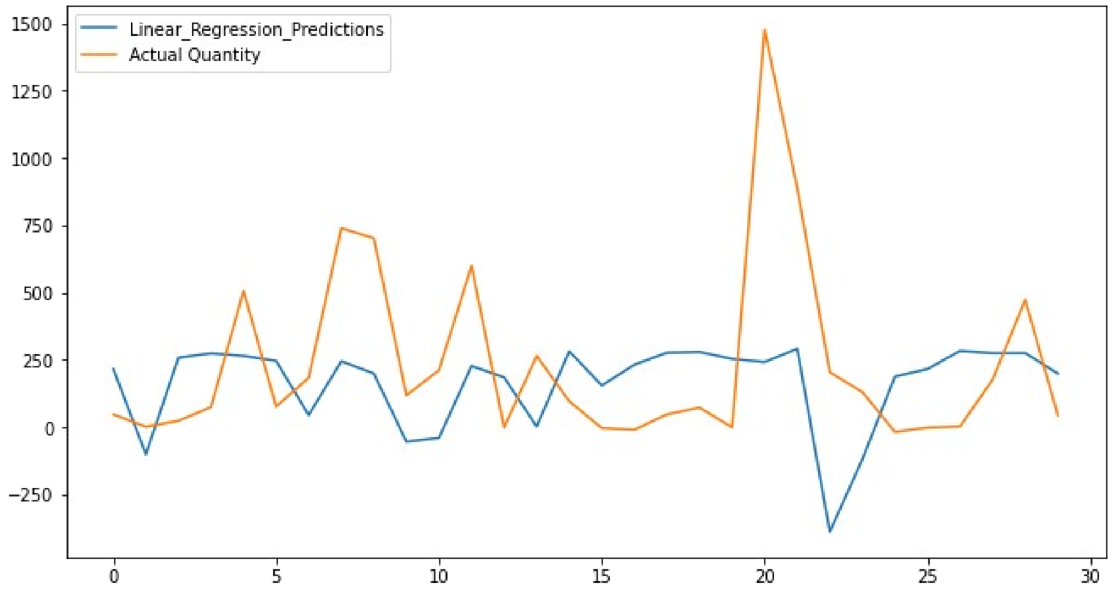
```
<ipython-input-13-f21a97a0524a>:1: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().
  model.fit(X_train,y_train)
```

```
LinearRegression()
```

```
pred=model.predict(X_test)
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (12,8)
plt.plot(pred,label='Random_Forest_Predictions')
plt.plot(y_test,label='Actual Quantity')
plt.legend(loc="upper left")
plt.show()
```



```
lin_pred=lin.predict(X_test)
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (11,6)
plt.plot(lin_pred,label='Linear_Regression_Predictions')
plt.plot(y_test,label='Actual Quantity')
plt.legend(loc="upper left")
plt.show()
```



```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse_rf=sqrt(mean_squared_error(pred,y_test))
rmse_lr=sqrt(mean_squared_error(lin_pred,y_test))

print('Mean Squared Error for Random Forest Model is:',rmse_rf)
print('Mean Squared Error for Linear Regression Model is:',rmse_lr)
```

```
Mean Squared Error for Random Forest Model is: 344.2949082896328
Mean Squared Error for Linear Regression Model is: 360.02251046519234
```

Thank You