```
# Importing the required libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from google.colab import files
uploaded = files.upload()
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving BhartiAirtel.csv to BhartiAirtel (1).csv
```

```
data=pd.read_csv("BhartiAirtel.csv")
data=data[['Date', 'Price']]
data.head()
```

	Date	Price
0	29-11-2024	1,627.15
1	28-11-2024	1,560.40
2	27-11-2024	1,577.65
3	26-11-2024	1,577.25
4	25-11-2024	1,578.75

Annomaly detection

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2699 entries, 0 to 2698
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Date    2699 non-null   object
 1   Price   2699 non-null   object
dtypes: object(2)
memory usage: 42.3+ KB
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2699 entries, 0 to 2698
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Date    2699 non-null   object
 1   Price   2699 non-null   object
dtypes: object(2)
memory usage: 42.3+ KB
```

```
data['Date'] = pd.to_datetime(data['Date'])

data.head()
```

```
<ipython-input-14-9a6f85947b91>:1: UserWarning: Parsing dates in %d-%m-%Y format when dayfirst=False (the default) was specified. Pa
  data['Date'] = pd.to_datetime(data['Date']) # Assuming the column name is in fact 'Date'
```

	Date	Price
0	2024-11-29	1,627.15
1	2024-11-28	1,560.40
2	2024-11-27	1,577.65
3	2024-11-26	1,577.25
4	2024-11-25	1,578.75

```
data.info()
```

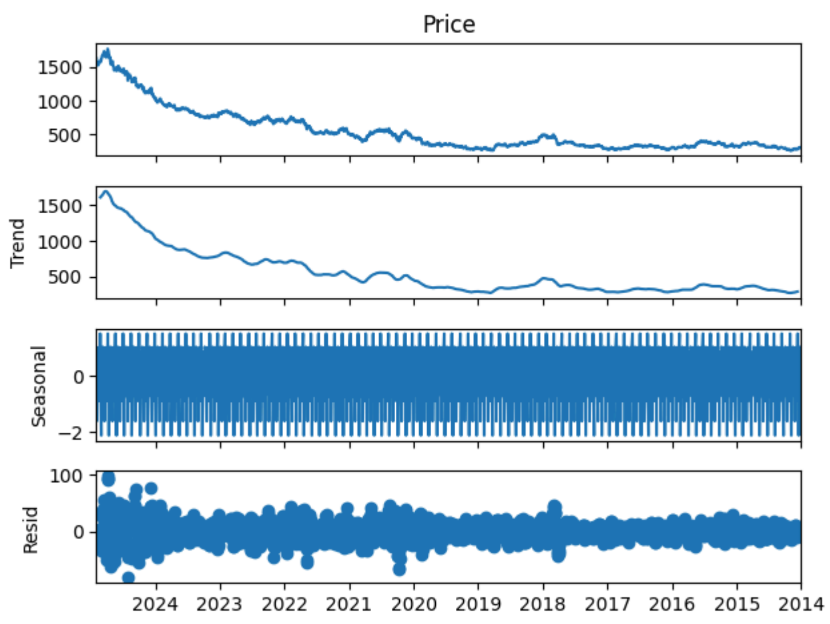
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2699 entries, 0 to 2698
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype         
---  ------  --------------  -----         
 0   Date    2699 non-null   datetime64[ns]
 1   Price   2699 non-null   object
dtypes: datetime64[ns](1), object(1)
memory usage: 42.3+ KB
```

Convert the "Date" column to datetime and set it as the index

```
data.set_index('Date', inplace=True)
```

Decompose time series

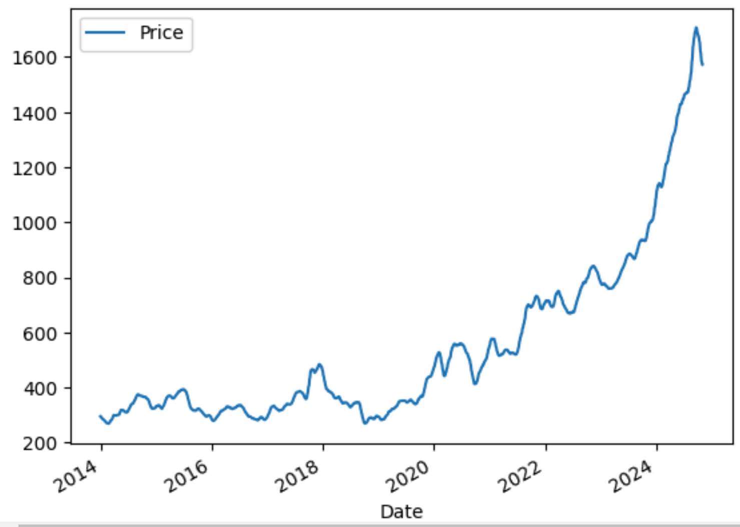
```
# Import necessary libraries
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt


# Convert the 'Price' column to numeric, removing commas
data['Price'] = pd.to_numeric(data['Price'].str.replace(',', ''))

# Now you can perform seasonal decomposition
try:
    decompose_result = seasonal_decompose(data['Price'], model='additive') # Access the 'Price' column
except ValueError:
    # If frequency inference fails, specify the period
    decompose_result = seasonal_decompose(data['Price'], model='additive', period=30) # Access the 'Price' column

# Plot the decomposed components
decompose_result.plot()
plt.show()
```



More decomposition using moving averagre to check trend and seasonality in the data

```
data_mean=data.rolling(window=20).mean()
data_mean.plot()
```

```
<Axes: xlabel='Date'>
```



Check Stationarity in the data

```
def difference_series(series, lag=1):
    return series.diff(lag).dropna()

# Perform first differencing
differenced_data = difference_series(data, lag=1)
```

As we can see that the series is Stationary

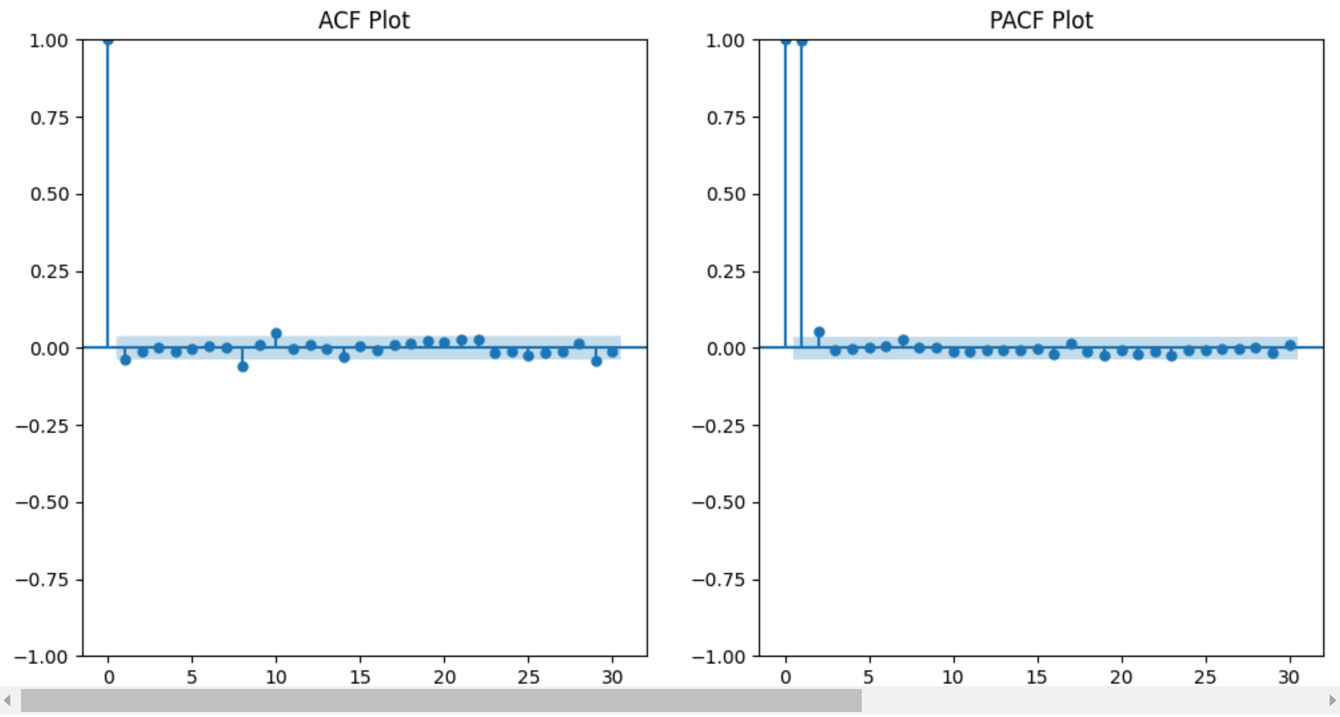
```
from statsmodels.tsa.stattools import adfuller

# Perform the Augmented Dickey-Fuller test
result = adfuller(differenced_data)
print("ADF Statistic:", result[0])
print("p-value:", result[1])
print("Critical Values:", result[4])

# Interpretation:
# - If p-value > 0.05, the series is not stationary.
# - Differencing might be needed if the series is non-stationary.
```

```
ADF Statistic: -54.29578560867521
p-value: 0.0
Critical Values: {'1%': -3.4327769688071754, '5%': -2.8626122563578624, '10%': -2.5673407977484697}
```

Model Building

ploting PACF and ACF plot for ARIMA model

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Plot ACF and PACF to identify lags
plt.figure(figsize=(12, 6))
plt.subplot(121)
plot_acf(differenced_data, ax=plt.gca(), lags=30)
plt.title("ACF Plot")

plt.subplot(122)
plot_pacf(data, ax=plt.gca(), lags=30)
plt.title("PACF Plot")

plt.show()
```



ARIMA

```
from statsmodels.tsa.arima.model import ARIMA
# Define ARIMA model
arima_order = (2, 1, 2)
arima_model = ARIMA(differenced_data[0:2571], order=arima_order)
arima_result = arima_model.fit()
aic_values_arima = arima_result.aic
bic_values_arima = arima_result.bic


# Summary of the model
print(arima_result.summary())

# Forecast next 11 outcomes to match the length of the actual data
arima_forecast = arima_result.forecast(steps=128)  # Changed from 12 to 11
print("ARIMA Forecast:", arima_forecast)

from sklearn.metrics import root_mean_squared_error
# Calculate RMSE using data[2571:] and the adjusted arima_forecast
rmse = root_mean_squared_error(data[2571:], arima_forecast)
print("RMSE:", rmse)
print(aic_values_arima)
print(bic_values_arima)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Price   No. Observations:                 2571
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -9644.458
Date:                Fri, 10 Jan 2025   AIC                          19298.917
Time:                        05:13:12   BIC                          19328.175
Sample:                             0   HQIC                         19309.524
                               - 2571
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.0109      0.054    -18.770      0.000      -1.116      -0.905
ar.L2         -0.0450      0.013     -3.339      0.001      -0.071      -0.019
ma.L1         -0.0281      0.053     -0.528      0.597      -0.133       0.076
ma.L2         -0.9668      0.053    -18.098      0.000      -1.072      -0.862
sigma2       106.2076      1.514     70.152      0.000     103.240     109.175
```

```
===============================================================================
Ljung-Box (L1) (Q):                   0.28   Jarque-Bera (JB):              5001.04
Prob(Q):                              0.60   Prob(JB):                         0.00
Heteroskedasticity (H):               0.21   Skew:                            -0.21
Prob(H) (two-sided):                  0.00   Kurtosis:                         9.82
===============================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
ARIMA Forecast: 2571   -0.112229
2572   -0.231707
2573    0.055389
2574   -0.229455
2575    0.045563
          ...
2694   -0.091207
2695   -0.087751
2696   -0.091083
2697   -0.087870
2698   -0.090968
Name: predicted_mean, Length: 128, dtype: float64
RMSE: 295.28185214260355
19298.916746687442
19328.175052576888
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In
  return get_prediction_index(
```

AR

```
# Define ARIMA model (p=1, d=0, q=1)
arima_order = (2, 0, 0)
AR_model = ARIMA(data[0:2571], order=arima_order)
AR_result = AR_model.fit()
aic_values_ar= AR_result.aic
bic_values_ar= AR_result.bic

# Summary of the model
print(AR_result.summary())

# Forecast next 11 outcomes to match the length of the actual data
AR_forecast = AR_result.forecast(steps=128)  # Changed from 12 to 11
print("AR Forecast:", AR_forecast)


from sklearn.metrics import root_mean_squared_error
rmse = root_mean_squared_error(data[2571:], AR_forecast) # data[2571:] has 11 samples
print("RMSE:", rmse)
print(aic_values_ar)
print(bic_values_ar)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Price   No. Observations:                 2571
Model:                 ARIMA(2, 0, 0)   Log Likelihood               -9654.442
Date:                Fri, 10 Jan 2025   AIC                          19316.884
Time:                        05:13:57   BIC                          19340.292
Sample:                             0   HQIC                         19325.370
                               - 2571
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        562.4027   1375.011      0.409      0.683   -2132.569    3257.374
ar.L1          0.9632      0.013     73.077      0.000       0.937       0.989
ar.L2          0.0367      0.013      2.793      0.005       0.011       0.062
sigma2       106.6007      1.452     73.402      0.000     103.754     109.447
===================================================================================
Ljung-Box (L1) (Q):                   0.11   Jarque-Bera (JB):              5389.34
Prob(Q):                              0.74   Prob(JB):                         0.00
Heteroskedasticity (H):               0.21   Skew:                            -0.46
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.03
===================================================================================
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
AR Forecast: 2571    312.747830
2572    312.771648
2573    312.796711
2574    312.821725
2575    312.846739
           ...
2694    315.805523
2695    315.830237
2696    315.854949
2697    315.879659
2698    315.904366
Name: predicted_mean, Length: 128, dtype: float64
RMSE: 26.757807171272823
19316.884152419436
19340.2923532485
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In
  return get_prediction_index(
```

MA

```
# Define ARIMA model (p=1, d=0, q=1)
arima_order = (0, 0, 2)
MA_model = ARIMA(data[0:2571], order=arima_order)
MA_result = MA_model.fit()
aic_values_ma= MA_result.aic
bic_values_ma= MA_result.bic

# Summary of the model
print(MA_result.summary())

# Forecast next 11 outcomes instead of 12 to match data[2571:]
MA_forecast = MA_result.forecast(steps=128)
print("MA Forecast:", MA_forecast)


from sklearn.metrics import root_mean_squared_error
rmse = root_mean_squared_error(data[2571:], MA_forecast)
print("RMSE:", rmse)
print(aic_values_ma)
print(bic_values_ma)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Price   No. Observations:                 2571
Model:                 ARIMA(0, 0, 2)   Log Likelihood              -15318.812
Date:                Fri, 10 Jan 2025   AIC                          30645.625
Time:                        05:15:27   BIC                          30669.033
Sample:                             0   HQIC                         30654.111
                               - 2571
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        562.1199     10.366     54.227      0.000     541.803     582.437
ma.L1          1.7650      0.006    274.312      0.000       1.752       1.778
ma.L2          0.9062      0.006    140.445      0.000       0.894       0.919
sigma2      8748.4477    237.889     36.775      0.000    8282.193    9214.702
===================================================================================
Ljung-Box (L1) (Q):                1531.31   Jarque-Bera (JB):              2186.03
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.22   Skew:                             1.68
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.02
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
MA Forecast: 2571    356.914759
2572    484.819635
```

```
2573    562.119869
2574    562.119869
2575    562.119869
           ...
2694    562.119869
2695    562.119869
2696    562.119869
2697    562.119869
2698    562.119869
Name: predicted_mean, Length: 128, dtype: float64
RMSE: 266.69342244804534
30645.62493980009
30669.033140629155
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Predic
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In th
  return get_prediction_index(
```

```
Final_metrics_data=pd.DataFrame({'Model':['AR','MA','ARIMA'],
                                 'AIC':[aic_values_ar,aic_values_ma,aic_values_arima],
                                 'BIC':[bic_values_ar,bic_values_ma,bic_values_arima],
                                 'RMSE':[root_mean_squared_error(data[2571:], AR_forecast),
                                         root_mean_squared_error(data[2571:], MA_forecast),
                                         root_mean_squared_error(data[2571:], arima_forecast)]})
Final_metrics_data
```

	Model	AIC	BIC	RMSE
0	AR	19316.884152	19340.292353	26.757807
1	MA	30645.624940	30669.033141	266.693422