

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARIMA
import datetime
import itertools
import warnings
from sklearn.metrics import mean_squared_error
import seaborn as sns
import statsmodels
plt.style.use('fivethirtyeight')
%matplotlib inline
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
import pandas.util.testing as tm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
df = pd.read_csv("/content/drive/MyDrive/Ineuron/data_stocks.csv")
df.head()
```

	DATE	SP500	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP
0	1491226200	2363.6101	42.3300	143.6800	129.6300	82.040	102.2300
1	1491226260	2364.1001	42.3600	143.7000	130.3200	82.080	102.1400
2	1491226320	2362.6799	42.3100	143.6901	130.2250	82.030	102.2125
3	1491226380	2364.3101	42.3700	143.6400	130.0729	82.000	102.1400
4	1491226440	2364.8501	42.5378	143.6600	129.8800	82.035	102.0600

5 rows × 502 columns

```
df["DATE"].dtypes
```

```
dtype('int64')
```

```
df['DATE'] = pd.to_datetime(df['DATE'], unit='s')
```

```
df['DATE'].tail()
```

```
41261    2017-08-31 19:56:00
41262    2017-08-31 19:57:00
41263    2017-08-31 19:58:00
41264    2017-08-31 19:59:00
41265    2017-08-31 20:00:00
Name: DATE, dtype: datetime64[ns]
```

```
df.index = df['DATE']
```

```
df.drop('DATE',axis = 1,inplace=True)
```

```
df.tail()
```

	SP500	NASDAQ.AAL	NASDAQ.AAPL	NASDAQ.ADBE	NASDAQ.ADI	NASDAQ.ADP	NASDAQ.A
DATE							
2017-08-31 19:56:00	2472.22	44.72	164.11	155.090	83.67	106.565	11.
2017-08-31 19:57:00	2471.77	44.73	164.12	155.160	83.65	106.590	11.
2017-08-31 19:58:00	2470.03	44.74	164.01	155.065	83.62	106.520	11.
2017-08-31 19:59:00	2471.49	44.71	163.88	154.960	83.58	106.400	11.
2017-08-31 20:00:00	2471.49	44.74	163.98	155.160	83.69	106.470	11.

5 rows × 501 columns

▶ NASDAQ.ADP

[] ↳ 19 cells hidden

▶ NADBAQ.CBOE

[] ↳ 14 cells hidden

▼ NASDAQ.CSCO

```
df_CSCO = df[["NASDAQ.CSCO"]].copy()
df_CSCO.tail()
```

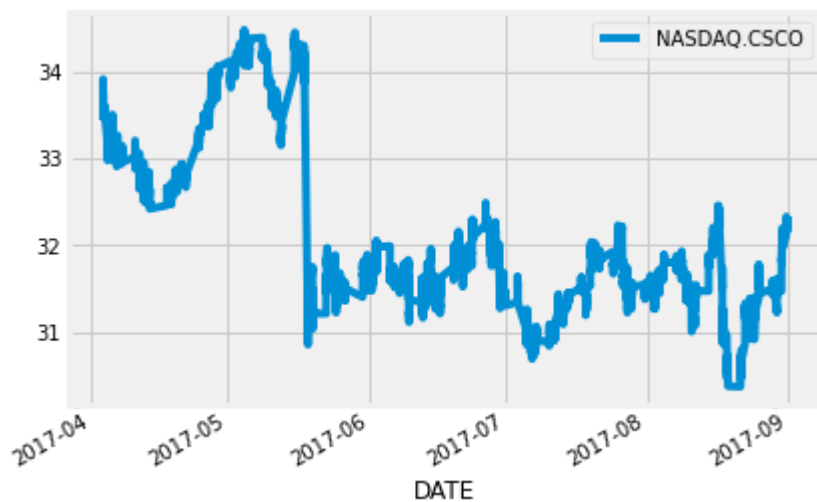
NASDAQ.CSCO	
DATE	
2017-08-31 19:56:00	32.185
2017-08-31 19:57:00	32.200
2017-08-31 19:58:00	32.200
2017-08-31 19:59:00	32.195
2017-08-31 20:00:00	32.225

```
df_CSCO.count()
```

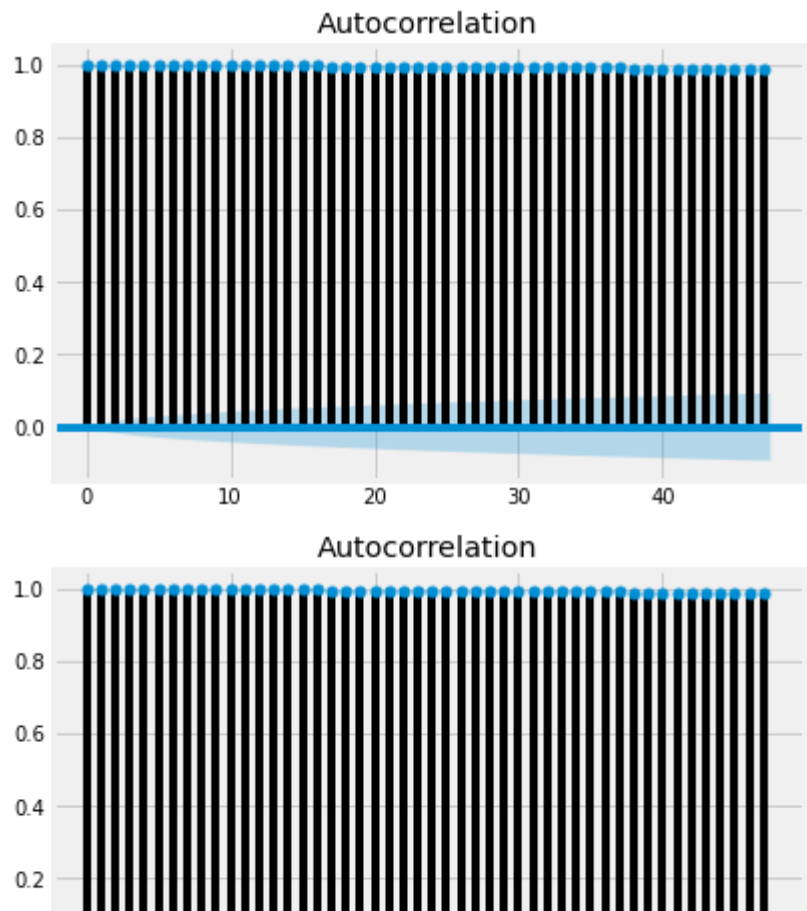
```
NASDAQ.CSCO    41266
dtype: int64
```

```
df_CSCO.plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f98413a5978>



```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(df_CSCO)
```



```
##Converting series to stationary
df_CSCO.shift(1)
```

NASDAQ.CSCO	
DATE	
2017-04-03 13:30:00	NaN
2017-04-03 13:31:00	33.7400
2017-04-03 13:32:00	33.8800
2017-04-03 13:33:00	33.9000
2017-04-03 13:34:00	33.8499
...	...
2017-08-31 19:56:00	32.1700
2017-08-31 19:57:00	32.1850
2017-08-31 19:58:00	32.2000
2017-08-31 19:59:00	32.2000
2017-08-31 20:00:00	32.1950

41266 rows × 1 columns

```

X4 = df_CSC0.values
train4 = X4[0:28886] # 27 data as train data
test4 = X4[28886:] # 9 data as test data
print(train4.size)
print(test4.size)
predictions4 = []

28886
12380

p4=d4=q4=range(0,2)
pdq4=list(itertools.product(p4,d4,q4))

warnings.filterwarnings('ignore')
for param in pdq4:
    try:
        model_arima4 = ARIMA(train4, order=param)
        model_arima_fit4 = model_arima4.fit()
        print(param,model_arima_fit4.aic)
    except:
        continue

(0, 0, 0) 85353.81423993816
(0, 0, 1) 46052.34616741124
(0, 1, 0) -135890.52761203377
(0, 1, 1) -135922.78479698166
(1, 0, 0) -135888.85266493712
(1, 0, 1) -135921.37291295695
(1, 1, 0) -135920.50823736849
(1, 1, 1) -136015.15611626126

from statsmodels.tsa.arima_model import ARIMA
model_arima4 = ARIMA(train4, order=(4,1,4))
model_arima_fit4 = model_arima4.fit()

predictions4 = model_arima_fit4.forecast(steps=12380)[0]
predictions4

array([31.87458762, 31.87343597, 31.87461045, ..., 31.07262991,
       31.07256513, 31.07250035])

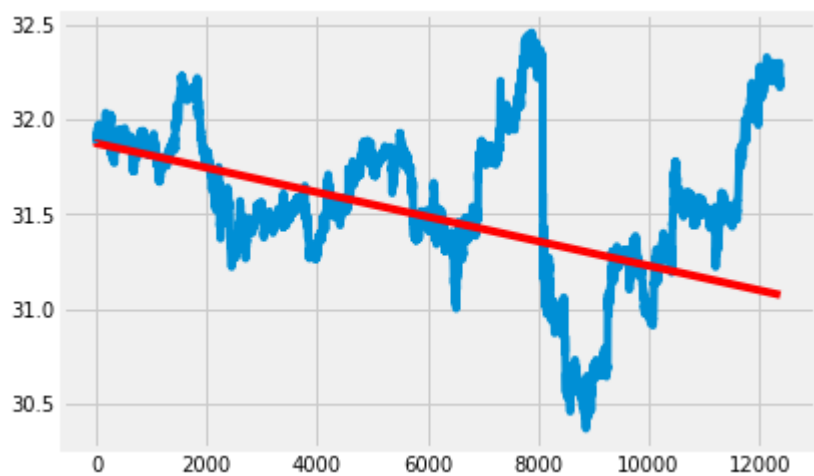
res4 = round(mean_squared_error(test4,predictions4))
res4

0

plt.plot(test4)
plt.plot(predictions4, color='red')

```

```
[<matplotlib.lines.Line2D at 0x7f98410d01d0>]
```



► NASDAQ.EBAY

[] ↳ 12 cells hidden