

Import modules and libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
import warnings
warnings.filterwarnings("ignore")
import datetime
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import
accuracy_score, classification_report, confusion_matrix, r2_score
from sklearn.tree import DecisionTreeClassifier
```

Load the dataset

```
data=pd.read_csv(r"C:\Users\Admin\Downloads\ADANI PORTS.csv")
data.head()
```

	Date	Symbol	Series	Prev Close	Open	High	Low
Last \							
0	2007-11-27	MUNDRAPORT	EQ	440.00	770.00	1050.00	770.0
959.0							
1	2007-11-28	MUNDRAPORT	EQ	962.90	984.00	990.00	874.0
885.0							
2	2007-11-29	MUNDRAPORT	EQ	893.90	909.00	914.75	841.0
887.0							
3	2007-11-30	MUNDRAPORT	EQ	884.20	890.00	958.00	890.0
929.0							
4	2007-12-03	MUNDRAPORT	EQ	921.55	939.75	995.00	922.0
980.0							

	Close	VWAP	Volume	Turnover	Trades	Deliverable	Volume
\							
0	962.90	984.72	27294366	2.687719e+15	NaN		9859619
1	893.90	941.38	4581338	4.312765e+14	NaN		1453278
2	884.20	888.09	5124121	4.550658e+14	NaN		1069678
3	921.55	929.17	4609762	4.283257e+14	NaN		1260913

```
4  969.30  965.65  2977470  2.875200e+14  NaN  816123
```

```
%Deliverble
0      0.3612
1      0.3172
2      0.2088
3      0.2735
4      0.2741
```

```
data.shape
```

```
(3322, 15)
```

```
data.duplicated().sum()
```

```
0
```

```
data.describe()
```

	Prev Close	Open	High	Low	Last
\count	3322.000000	3322.000000	3322.000000	3322.000000	3322.000000
mean	344.114314	344.763019	351.608007	337.531969	344.239539
std	192.936882	193.619992	198.617808	188.676614	193.187813
min	108.000000	108.000000	110.450000	105.650000	108.000000
25%	164.312500	164.850000	168.000000	161.600000	164.075000
50%	324.700000	325.750000	331.275000	319.850000	325.000000
75%	400.912500	401.000000	407.187500	395.000000	400.912500
max	1307.450000	1310.250000	1324.000000	1270.000000	1308.000000

	Close	VWAP	Volume	Turnover
Trades \				
count	3322.000000	3322.000000	3.322000e+03	3.322000e+03
mean	344.201626	344.853182	2.954564e+06	1.070144e+14
std	193.045886	193.841305	4.104227e+06	2.625564e+14
min	108.000000	108.340000	1.236600e+04	2.415857e+11
25%	164.312500	164.855000	7.493682e+05	1.817650e+13

```

50%      324.700000    325.765000    2.007292e+06    5.836041e+13
3.588150e+04
75%      400.912500    400.607500    3.636883e+06    1.158526e+14
5.336875e+04
max      1307.450000    1302.150000    9.771788e+07    8.160988e+15
1.205984e+06

```

	Deliverable Volume	%Deliverble
count	3.322000e+03	3322.000000
mean	1.207441e+06	0.445899
std	1.398640e+06	0.160496
min	5.383000e+03	0.067000
25%	3.212005e+05	0.332900
50%	8.132775e+05	0.445650
75%	1.605528e+06	0.555850
max	2.241652e+07	0.979800

```
data.isnull().sum()
```

```

Date      0
Symbol    0
Series    0
Prev Close 0
Open      0
High      0
Low       0
Last      0
Close     0
VWAP      0
Volume    0
Turnover  0
Trades    866
Deliverable Volume 0
%Deliverble 0
dtype: int64

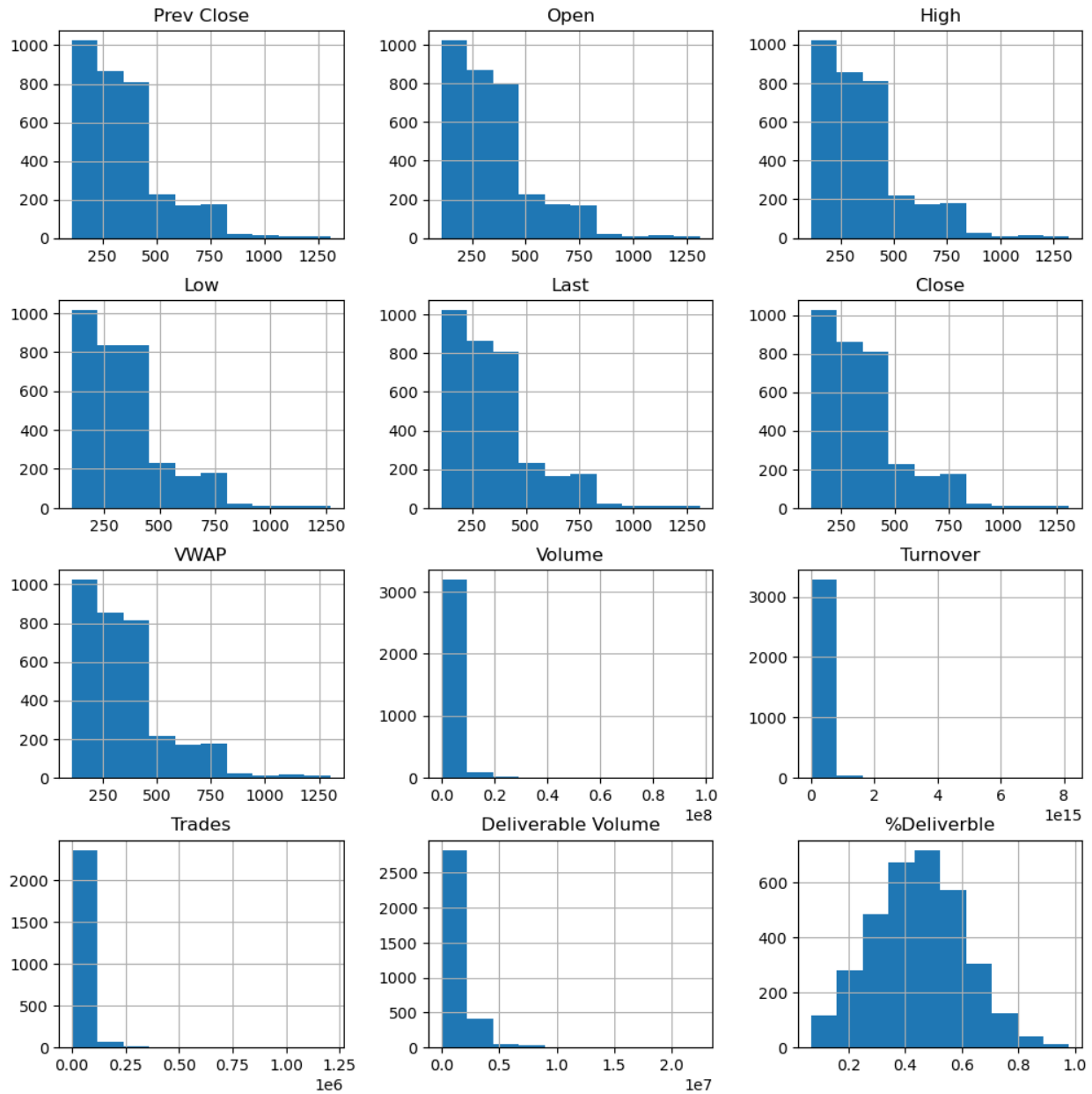
```

visualizing the data

```

data.hist(figsize=(12,12))
plt.show()

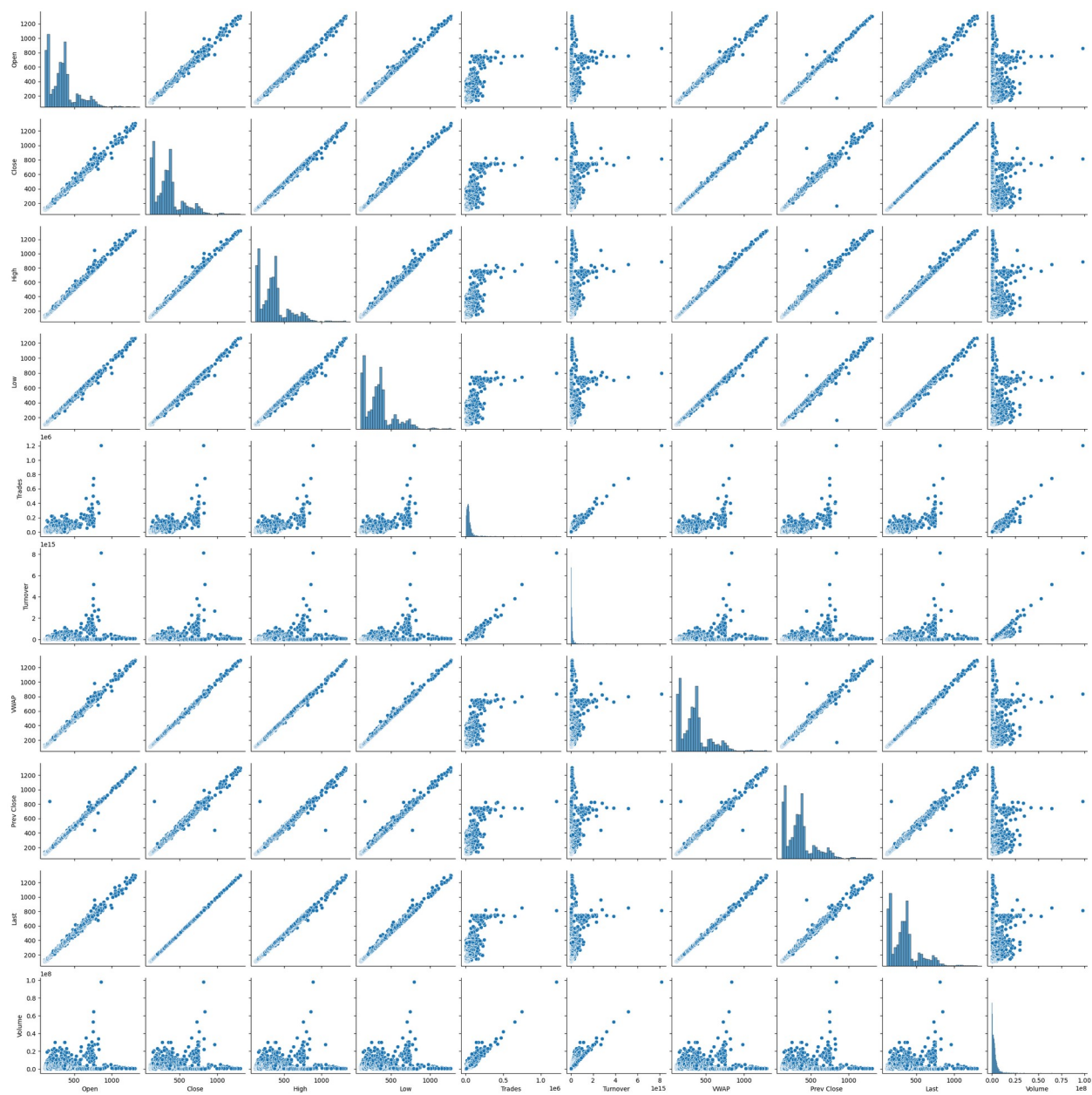
```



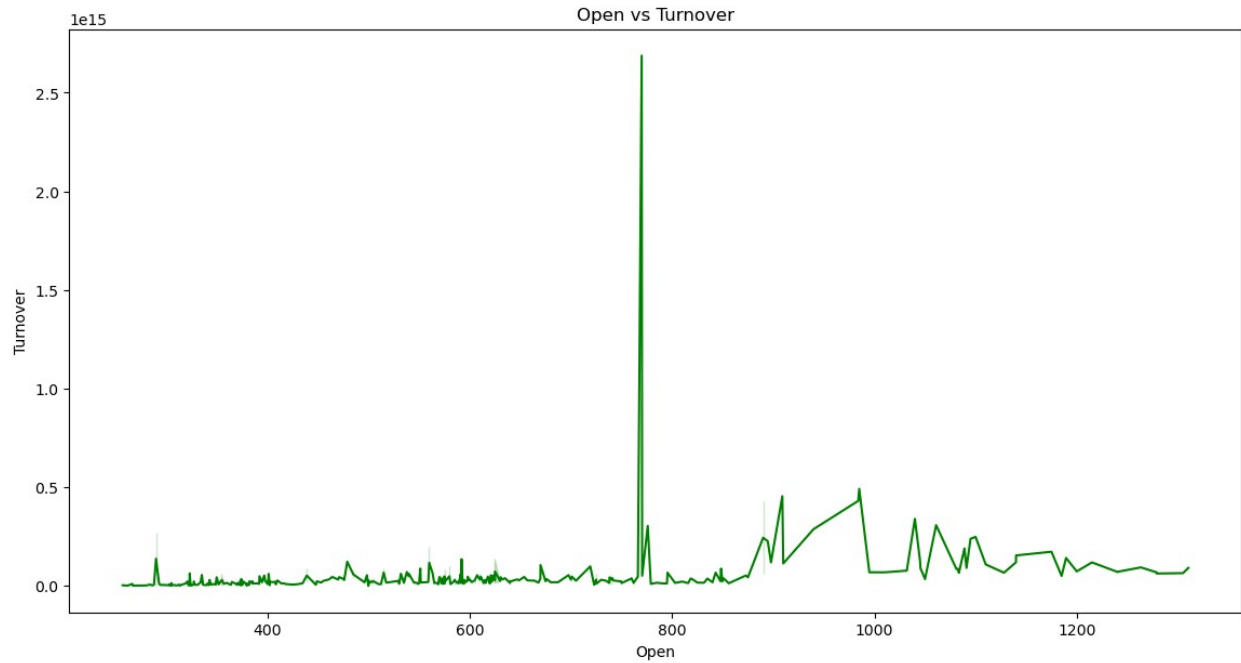
```
mean_col = ['Open', 'Close', 'High', 'Low',
            'Trades', 'Turnover', 'VWAP', 'Prev Close',
            'Last', 'Volume']

sns.pairplot(data[mean_col], palette='Accent')

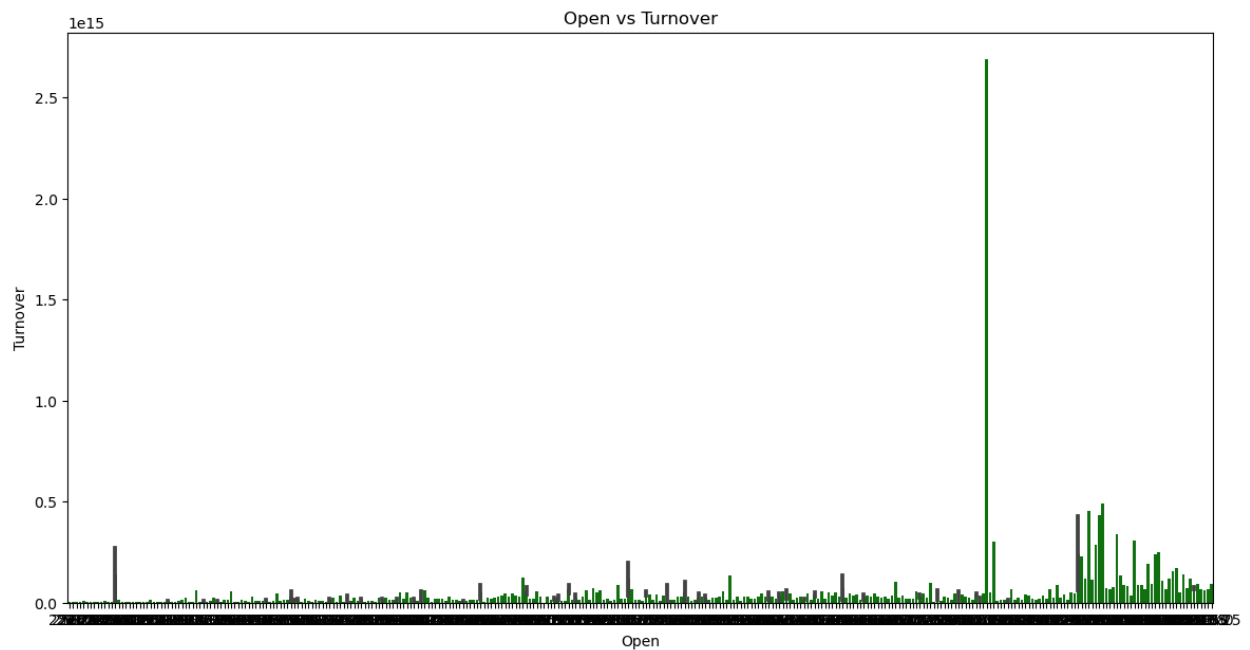
<seaborn.axisgrid.PairGrid at 0x1da93d007c0>
```



```
plt.figure(figsize=(14,7))
sns.lineplot(x = "Open",y = "Turnover",data = data[0:400],
color='green')
plt.title("Open vs Turnover")
plt.xlabel("Open")
plt.ylabel("Turnover")
plt.show()
```

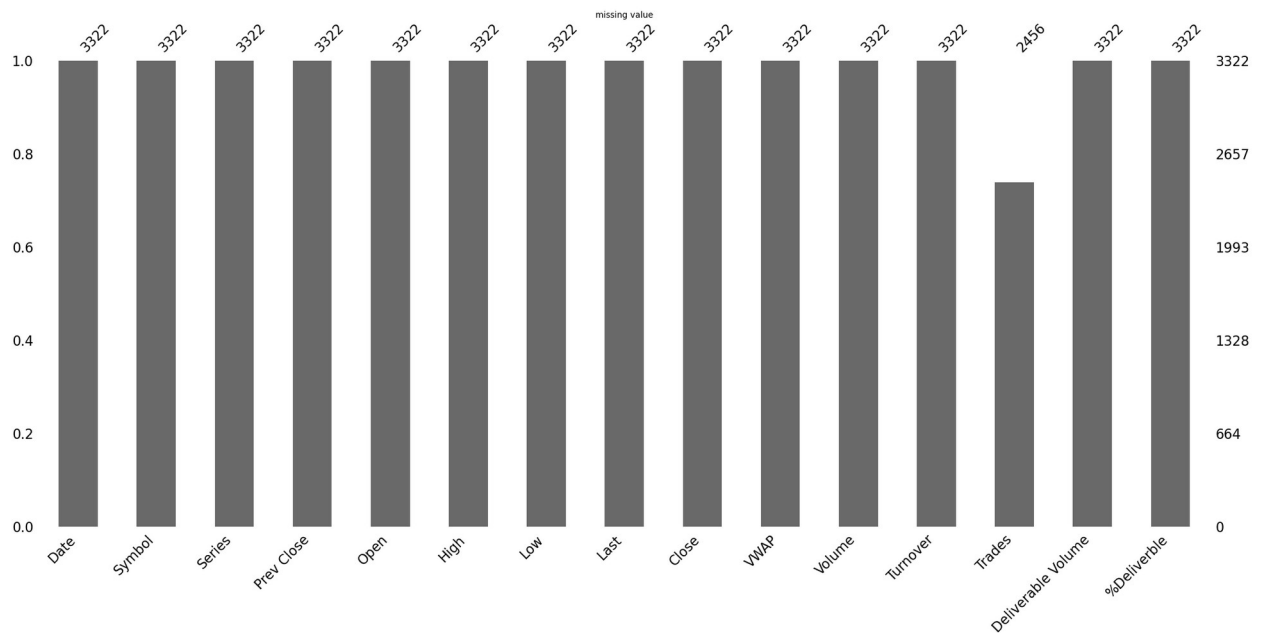


```
plt.figure(figsize=(14,7))
sns.barplot(x = "Open",y = "Turnover",data = data[0:400],
color='green')
plt.title("Open vs Turnover")
plt.xlabel("Open")
plt.ylabel("Turnover")
plt.show()
```



```
import missingno as msno
msno.bar(data)
plt.xlabel('missing value')
plt.ylabel('count')

Text(0, 0.5, 'count')
```



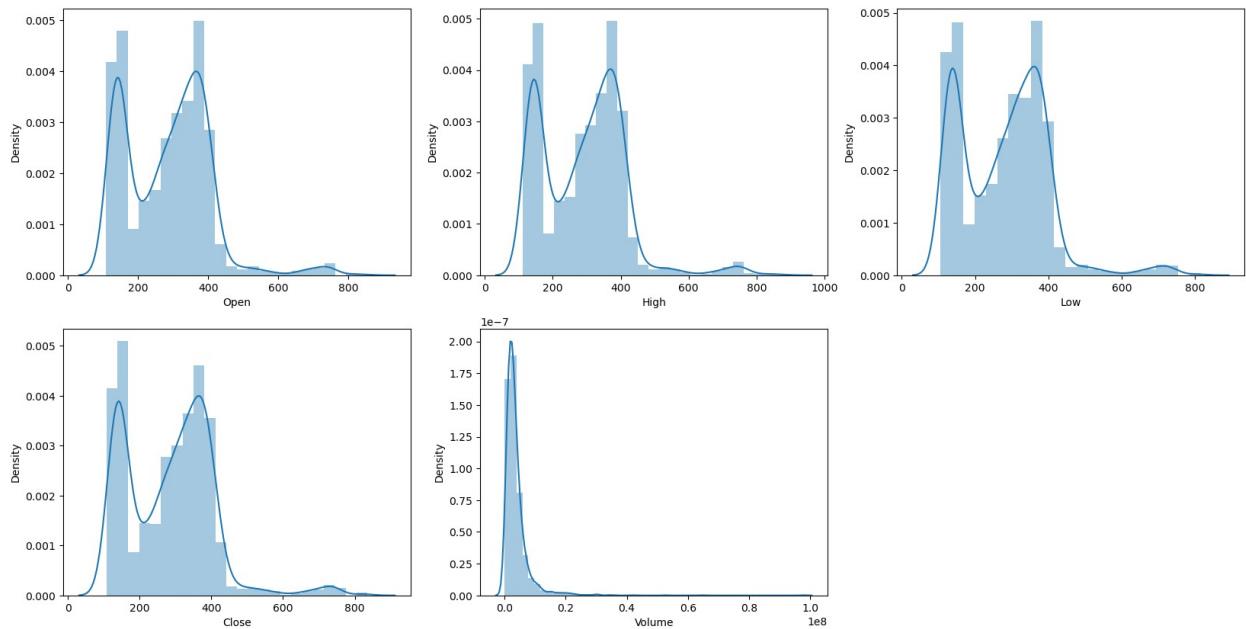
```
data.dropna(inplace=True)
data.isnull().sum()
```

```
Date      0
Symbol    0
Series    0
Prev Close 0
Open      0
High      0
Low       0
Last      0
Close     0
VWAP      0
Volume    0
Turnover  0
Trades    0
Deliverable Volume 0
%Deliverble 0
dtype: int64
```

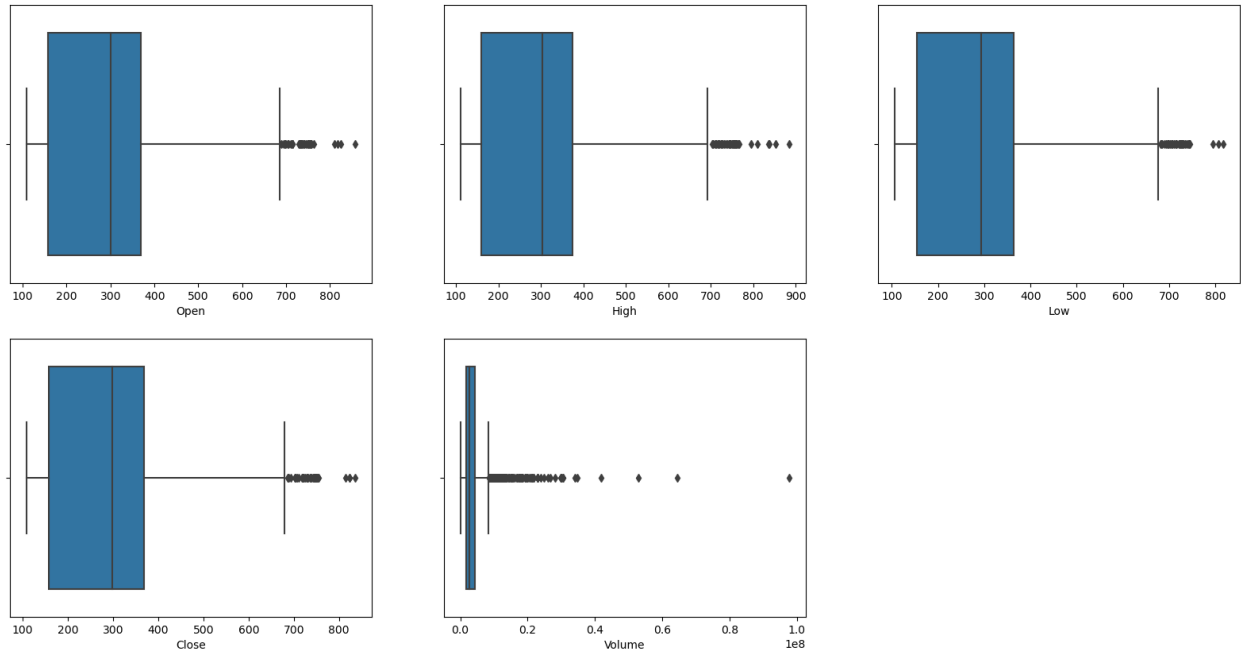
```
features = ['Open', 'High', 'Low', 'Close', 'Volume']
```

```
plt.subplots(figsize=(20,10))
```

```
for i, col in enumerate(features):  
    plt.subplot(2,3,i+1)  
    sns.distplot(data[col])  
plt.show()
```



```
plt.subplots(figsize=(20,10))  
for i, col in enumerate(features):  
    plt.subplot(2,3,i+1)  
    sns.boxplot(data[col])  
plt.show()
```

```
data['Month']=pd.DatetimeIndex(data['Date']).month
data['Year']=pd.DatetimeIndex(data['Date']).year
data
```

	Date	Symbol	Series	Prev	Close	Open	High
Low \							
866	2011-06-01	MUNDRAPORT	EQ	161.45	162.10	165.70	
161.25							
867	2011-06-02	MUNDRAPORT	EQ	164.00	164.00	165.15	
160.15							
868	2011-06-03	MUNDRAPORT	EQ	161.25	161.50	162.80	
159.20							
869	2011-06-06	MUNDRAPORT	EQ	161.05	160.50	161.10	
159.05							
870	2011-06-07	MUNDRAPORT	EQ	159.85	159.85	162.75	
156.35							
...
.							
3317	2021-04-26	ADANI PORTS	EQ	725.35	733.00	739.65	
728.90							
3318	2021-04-27	ADANI PORTS	EQ	730.75	735.00	757.50	
727.35							
3319	2021-04-28	ADANI PORTS	EQ	749.15	755.00	760.00	
741.10							
3320	2021-04-29	ADANI PORTS	EQ	746.25	753.20	765.85	
743.40							
3321	2021-04-30	ADANI PORTS	EQ	746.75	739.00	759.45	
724.50							

	Last	Close	VWAP	Volume	Turnover	Trades	\
--	------	-------	------	--------	----------	--------	---

866	163.50	164.00	164.08	2574106	4.223703e+13	19171.0
867	161.15	161.25	162.17	1699298	2.755678e+13	16176.0
868	161.00	161.05	161.02	1185817	1.909361e+13	14810.0
869	160.00	159.85	160.09	546378	8.746905e+12	7071.0
870	157.00	157.25	158.52	2193466	3.477027e+13	17865.0
...
3317	729.20	730.75	733.25	9390549	6.885658e+14	116457.0
3318	748.60	749.15	747.67	20573107	1.538191e+15	236896.0
3319	743.40	746.25	751.02	11156977	8.379106e+14	130847.0
3320	746.40	746.75	753.06	13851910	1.043139e+15	153293.0
3321	726.40	730.05	743.35	12600934	9.366911e+14	132141.0

	Deliverable Volume	%Deliverble	Month	Year
866	1271255	0.4939	6	2011
867	791462	0.4658	6	2011
868	722154	0.6090	6	2011
869	386144	0.7067	6	2011
870	1425849	0.6500	6	2011
...
3317	838079	0.0892	4	2021
3318	1779639	0.0865	4	2021
3319	1342353	0.1203	4	2021
3320	1304895	0.0942	4	2021
3321	3514692	0.2789	4	2021

[2456 rows x 17 columns]

```
data['is_quarter_end']=np.where(data['Month']%3==0,1,0)
data
```

	Date	Symbol	Series	Prev Close	Open	High
Low \						
866	2011-06-01	MUNDRAPORT	EQ	161.45	162.10	165.70
161.25						
867	2011-06-02	MUNDRAPORT	EQ	164.00	164.00	165.15
160.15						
868	2011-06-03	MUNDRAPORT	EQ	161.25	161.50	162.80
159.20						
869	2011-06-06	MUNDRAPORT	EQ	161.05	160.50	161.10
159.05						
870	2011-06-07	MUNDRAPORT	EQ	159.85	159.85	162.75
156.35						
...
.						
3317	2021-04-26	ADANI PORTS	EQ	725.35	733.00	739.65
728.90						
3318	2021-04-27	ADANI PORTS	EQ	730.75	735.00	757.50
727.35						
3319	2021-04-28	ADANI PORTS	EQ	749.15	755.00	760.00
741.10						

3320	2021-04-29	ADANI PORTS	EQ	746.25	753.20	765.85
743.40						
3321	2021-04-30	ADANI PORTS	EQ	746.75	739.00	759.45
724.50						

	Last	Close	VWAP	Volume	Turnover	Trades \
866	163.50	164.00	164.08	2574106	4.223703e+13	19171.0
867	161.15	161.25	162.17	1699298	2.755678e+13	16176.0
868	161.00	161.05	161.02	1185817	1.909361e+13	14810.0
869	160.00	159.85	160.09	546378	8.746905e+12	7071.0
870	157.00	157.25	158.52	2193466	3.477027e+13	17865.0
...
3317	729.20	730.75	733.25	9390549	6.885658e+14	116457.0
3318	748.60	749.15	747.67	20573107	1.538191e+15	236896.0
3319	743.40	746.25	751.02	11156977	8.379106e+14	130847.0
3320	746.40	746.75	753.06	13851910	1.043139e+15	153293.0
3321	726.40	730.05	743.35	12600934	9.366911e+14	132141.0

	Deliverable Volume	%Deliverble	Month	Year	is_quarter_end
866	1271255	0.4939	6	2011	1
867	791462	0.4658	6	2011	1
868	722154	0.6090	6	2011	1
869	386144	0.7067	6	2011	1
870	1425849	0.6500	6	2011	1
...
3317	838079	0.0892	4	2021	0
3318	1779639	0.0865	4	2021	0
3319	1342353	0.1203	4	2021	0
3320	1304895	0.0942	4	2021	0
3321	3514692	0.2789	4	2021	0

[2456 rows x 18 columns]

```
stockprice_oneyear=data[(data['Date']>'2007-03-31')&(data['Date']<'2012-04-01')]
stockprice_oneyear
```

	Date	Symbol	Series	Prev Close	Open	High
Low \						
866	2011-06-01	MUNDRAPORT	EQ	161.45	162.10	165.70
161.25						
867	2011-06-02	MUNDRAPORT	EQ	164.00	164.00	165.15
160.15						
868	2011-06-03	MUNDRAPORT	EQ	161.25	161.50	162.80
159.20						
869	2011-06-06	MUNDRAPORT	EQ	161.05	160.50	161.10
159.05						
870	2011-06-07	MUNDRAPORT	EQ	159.85	159.85	162.75
156.35						
...

```

.
1070 2012-03-26 ADANI PORTS      EQ      123.45  123.00  123.70
119.05
1071 2012-03-27 ADANI PORTS      EQ      120.55  121.55  123.00
118.25
1072 2012-03-28 ADANI PORTS      EQ      121.50  121.50  121.50
117.00
1073 2012-03-29 ADANI PORTS      EQ      118.25  118.00  125.50
116.10
1074 2012-03-30 ADANI PORTS      EQ      124.20  124.50  130.00
124.10

```

```

      Last   Close   VWAP   Volume   Turnover   Trades \
866    163.50  164.00  164.08  2574106  4.223703e+13  19171.0
867    161.15  161.25  162.17  1699298  2.755678e+13  16176.0
868    161.00  161.05  161.02  1185817  1.909361e+13  14810.0
869    160.00  159.85  160.09   546378  8.746905e+12   7071.0
870    157.00  157.25  158.52  2193466  3.477027e+13  17865.0
...      ...      ...      ...      ...      ...      ...
1070    119.75  120.55  121.00  1124699  1.360846e+13  14021.0
1071    121.65  121.50  120.66  1098462  1.325451e+13  14310.0
1072    118.55  118.25  118.44  1756343  2.080279e+13  26906.0
1073    124.15  124.20  120.57  2869127  3.459164e+13  29996.0
1074    128.90  129.50  127.74  1405080  1.794833e+13  24475.0

```

```

      Deliverable Volume  %Deliverble  Month  Year  is_quarter_end
866              1271255         0.4939      6  2011             1
867              791462         0.4658      6  2011             1
868              722154         0.6090      6  2011             1
869              386144         0.7067      6  2011             1
870             1425849         0.6500      6  2011             1
...              ...          ...      ...      ...      ...
1070             602271         0.5355      3  2012             1
1071             487096         0.4434      3  2012             1
1072             1083219        0.6167      3  2012             1
1073             1933482        0.6739      3  2012             1
1074             872331         0.6208      3  2012             1

```

```
[209 rows x 18 columns]
```

```
stockprice_oneyear.isnull().sum()
```

```

Date          0
Symbol        0
Series        0
Prev Close    0
Open          0
High          0
Low           0
Last          0

```

```

Close          0
VWAP           0
Volume         0
Turnover       0
Trades         0
Deliverable Volume  0
%Deliverble    0
Month          0
Year           0
is_quarter_end 0
dtype: int64

```

```
stockprice_oneyear.describe()
```

	Prev Close	Open	High	Low	Last
Close \					
count	209.000000	209.000000	209.000000	209.000000	209.000000
mean	144.543301	144.831100	147.739474	141.408134	144.327990
std	12.829445	12.942136	12.883065	13.079199	12.858106
min	114.700000	117.500000	120.900000	111.000000	115.000000
25%	133.450000	133.450000	137.300000	130.400000	132.750000
50%	147.100000	147.500000	150.550000	144.000000	146.750000
75%	155.150000	155.400000	158.200000	153.050000	155.250000
max	165.000000	165.050000	170.450000	163.550000	165.400000

	VWAP	Volume	Turnover	Trades \
count	209.00000	2.090000e+02	2.090000e+02	209.000000
mean	144.53512	1.434648e+06	2.039672e+13	13896.727273
std	12.99207	1.161678e+06	1.586701e+13	7357.804577
min	115.90000	2.209200e+04	3.235358e+11	417.000000
25%	133.46000	7.173930e+05	1.036262e+13	8995.000000
50%	146.92000	1.139611e+06	1.649115e+13	12293.000000
75%	155.67000	1.691856e+06	2.497078e+13	17565.000000
max	166.37000	9.280860e+06	1.286813e+14	46546.000000

	Deliverable Volume	%Deliverble	Month	Year \
count	2.090000e+02	209.000000	209.000000	209.000000
mean	8.300263e+05	0.531254	6.822967	2011.306220
std	8.834768e+05	0.136320	3.656166	0.462029
min	9.831000e+03	0.186900	1.000000	2011.000000
25%	3.301590e+05	0.427500	3.000000	2011.000000
50%	6.248810e+05	0.537400	7.000000	2011.000000

75%	1.017352e+06	0.613800	10.000000	2012.000000
max	8.597655e+06	0.943300	12.000000	2012.000000

```

is_quarter_end
count      209.000000
mean        0.411483
std         0.493284
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000

```

```
stockprice_oneyear.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 209 entries, 866 to 1074
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	Date	209 non-null	object
1	Symbol	209 non-null	object
2	Series	209 non-null	object
3	Prev Close	209 non-null	float64
4	Open	209 non-null	float64
5	High	209 non-null	float64
6	Low	209 non-null	float64
7	Last	209 non-null	float64
8	Close	209 non-null	float64
9	VWAP	209 non-null	float64
10	Volume	209 non-null	int64
11	Turnover	209 non-null	float64
12	Trades	209 non-null	float64
13	Deliverable Volume	209 non-null	int64
14	%Deliverble	209 non-null	float64
15	Month	209 non-null	int64
16	Year	209 non-null	int64
17	is_quarter_end	209 non-null	int32

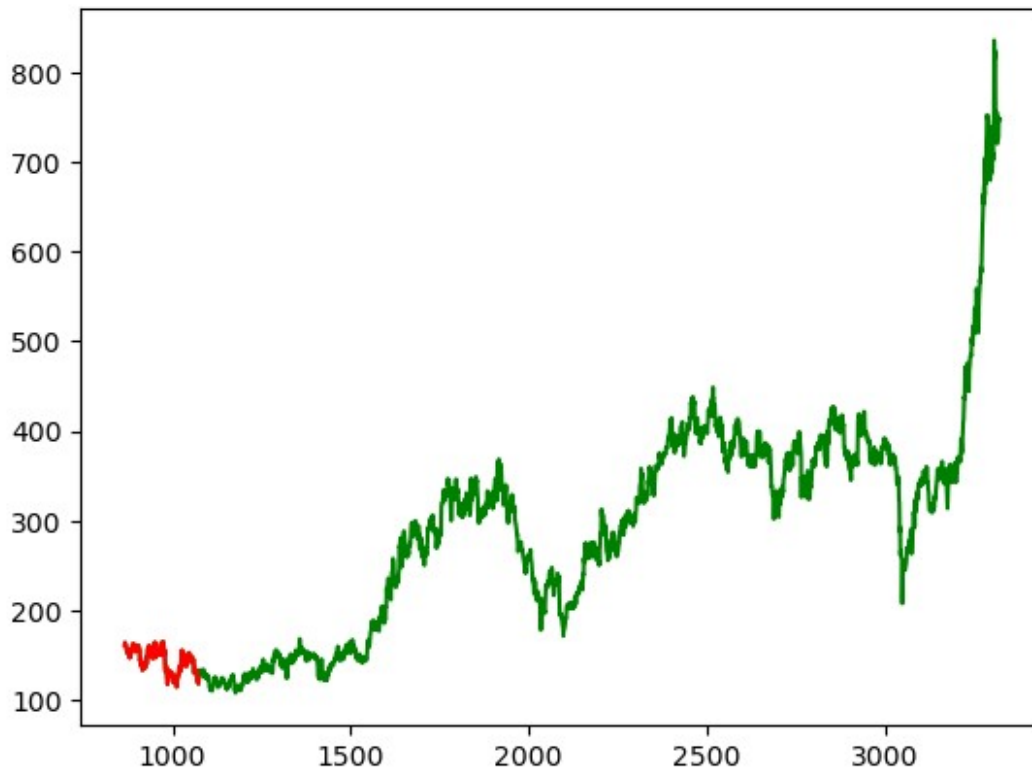
```
dtypes: float64(10), int32(1), int64(4), object(3)
```

```
memory usage: 30.2+ KB
```

```
df=data['Prev Close'].plot(c='green')
```

```
stockprice_oneyear['Prev Close'].plot(ax=df,c='red')
```

```
<AxesSubplot:>
```



```
plt.figure(figsize=(7,5))

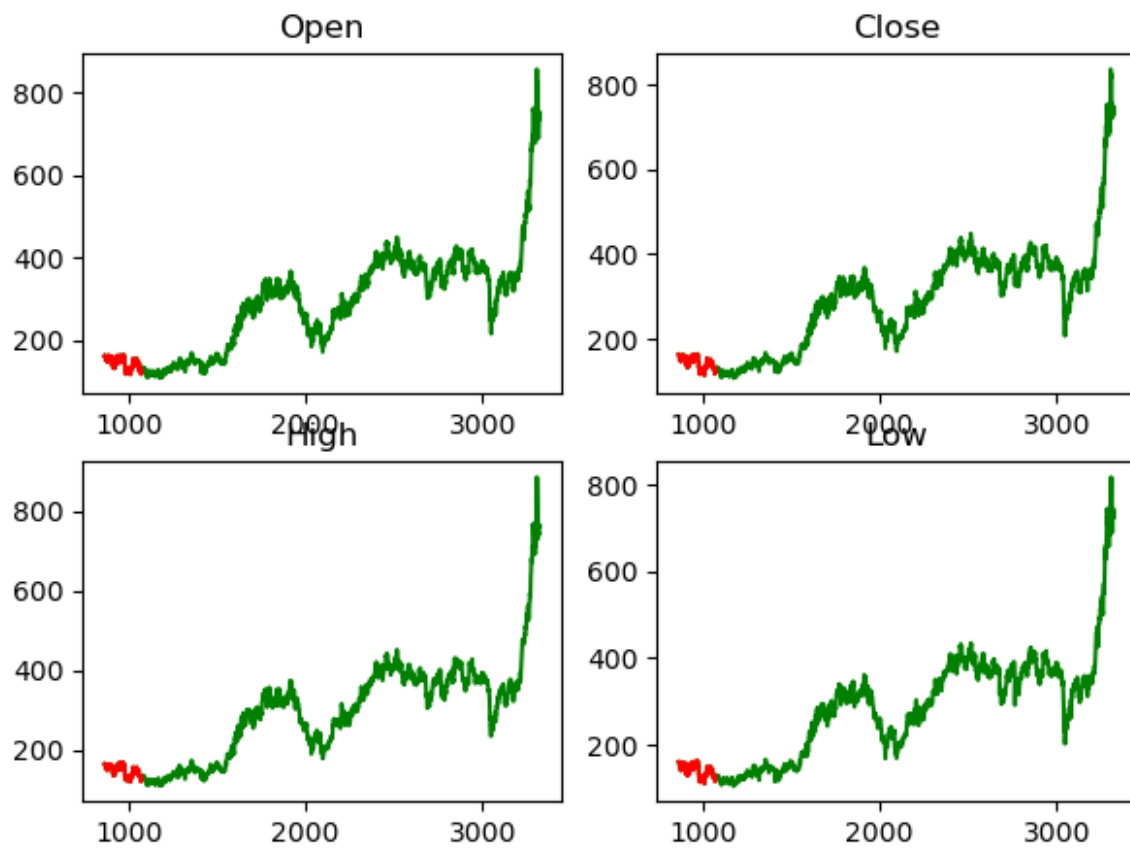
plt.subplot(221)
plt.title('Open')
df=data['Open'].plot(c='green')
stockprice_oneyear['Open'].plot(ax=df,c='red')

plt.subplot(222)
plt.title('Close')
df=data['Close'].plot(c='green')
stockprice_oneyear['Close'].plot(ax=df,c='red')

plt.subplot(223)
plt.title('High')
df=data['High'].plot(c='green')
stockprice_oneyear['High'].plot(ax=df,c='red')

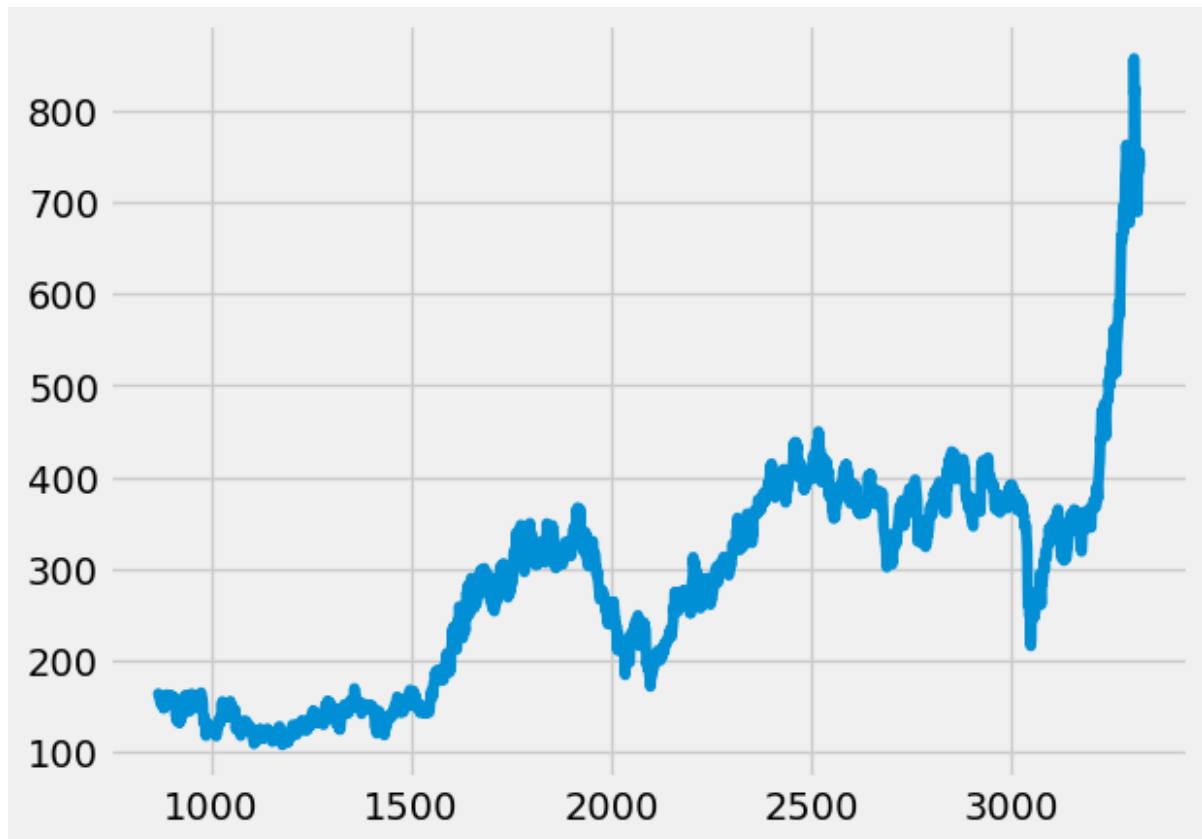
plt.subplot(224)
plt.title('Low')
df=data['Low'].plot(c='green')
stockprice_oneyear['Low'].plot(ax=df,c='red')
```

```
plt.show()
```

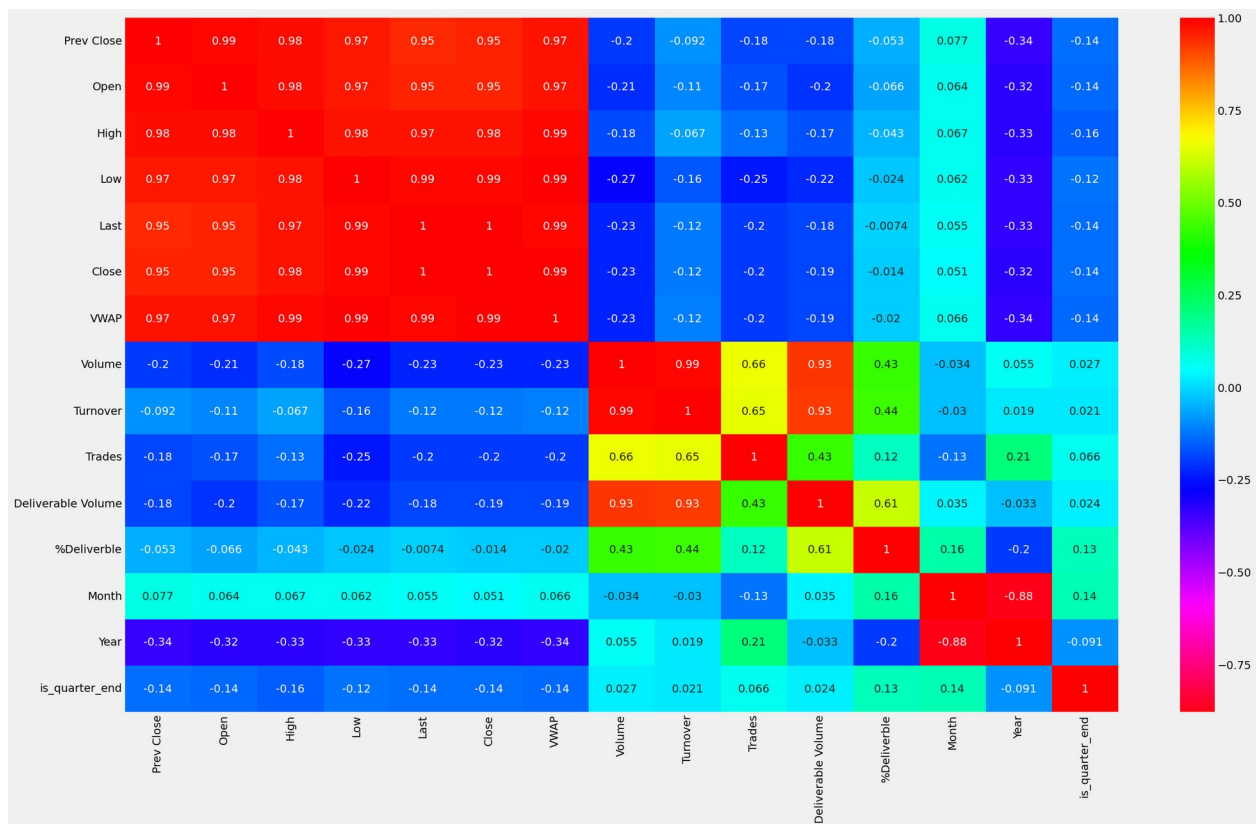


```
plt.style.use('fivethirtyeight')
plt.plot(data['Open'])

[<matplotlib.lines.Line2D at 0x1da9cd9ce20>]
```

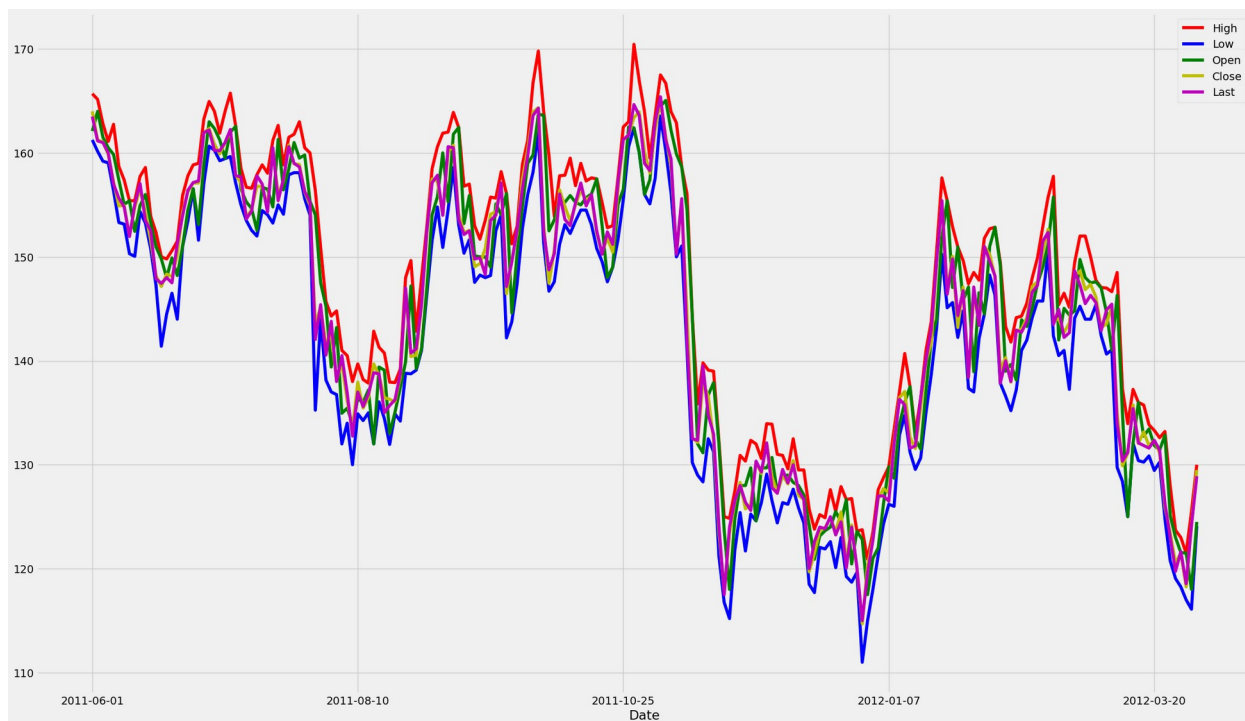



```
plt.figure(figsize=(25,15))
sns.heatmap(stockprice_oneyear.corr(), cmap="hsv_r", annot=True)
plt.show()
```



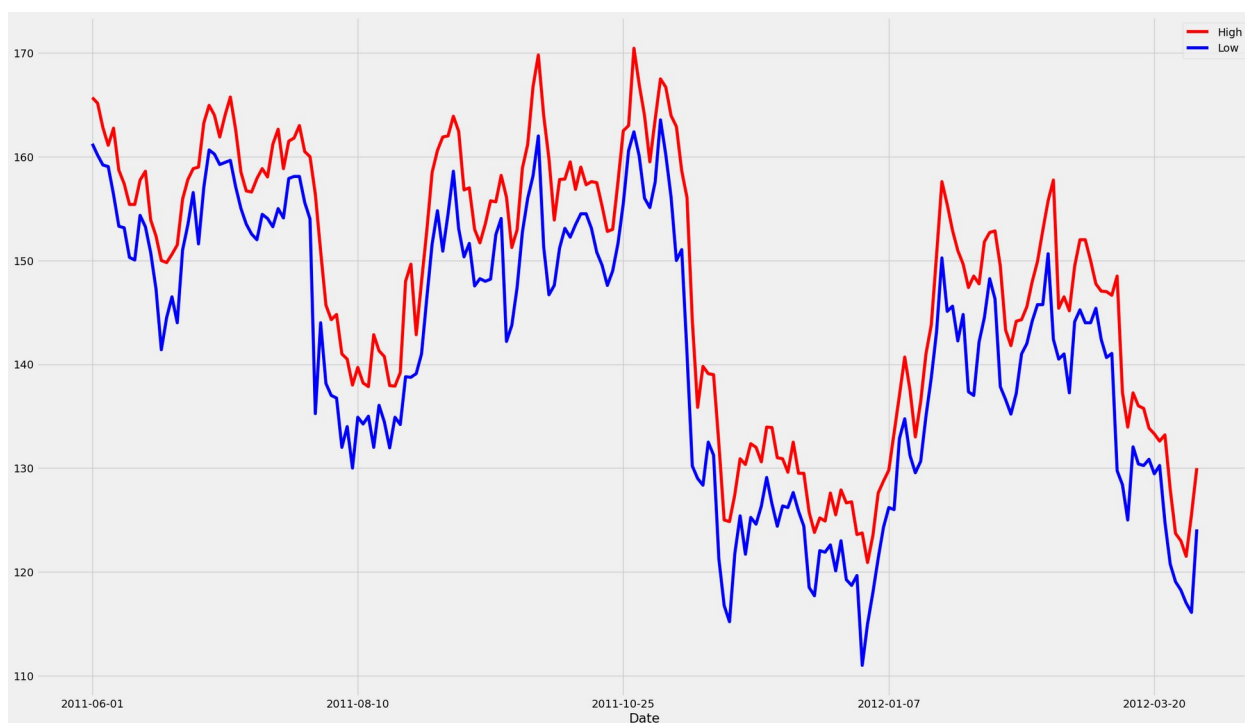
```
stockprice_oneyear.plot(x='Date',y=['High','Low','Open','Close','Last'],color=['r','b','g','y','m'],figsize=[25,15])
```

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



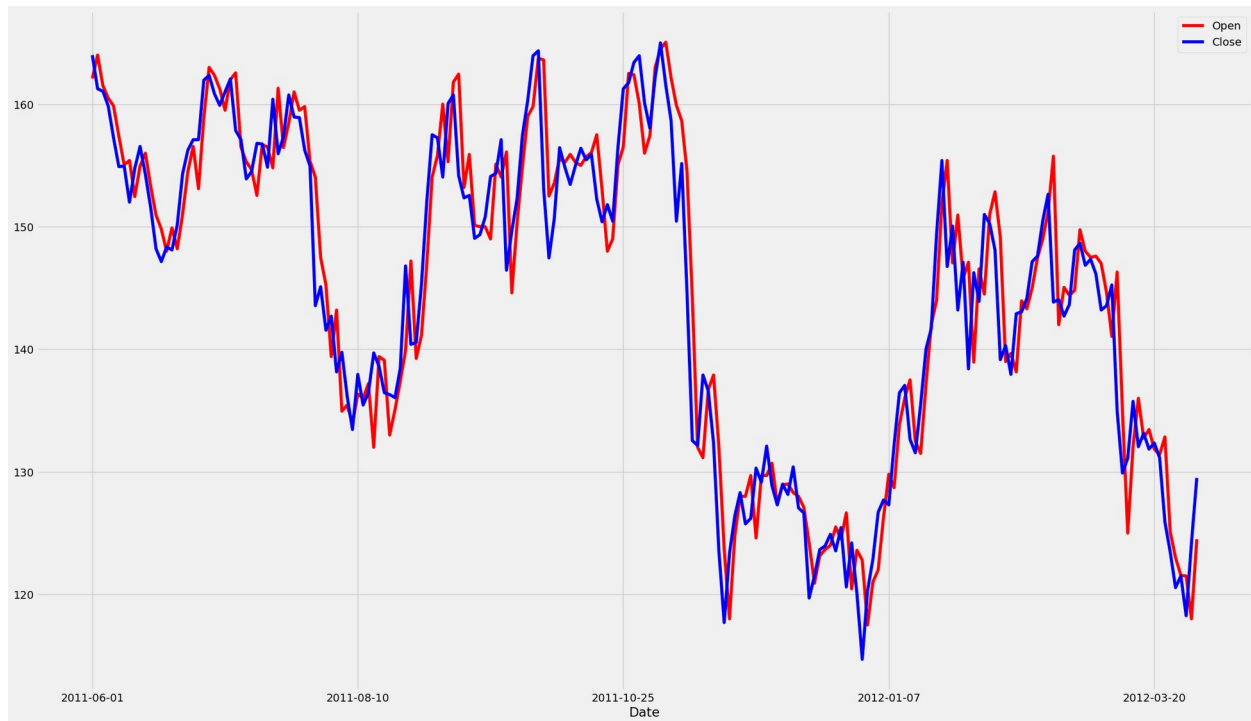
```
stockprice_oneyear.plot(x='Date',y=['High','Low'],color=['r','b'],figs
ize=[25,15])
```

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

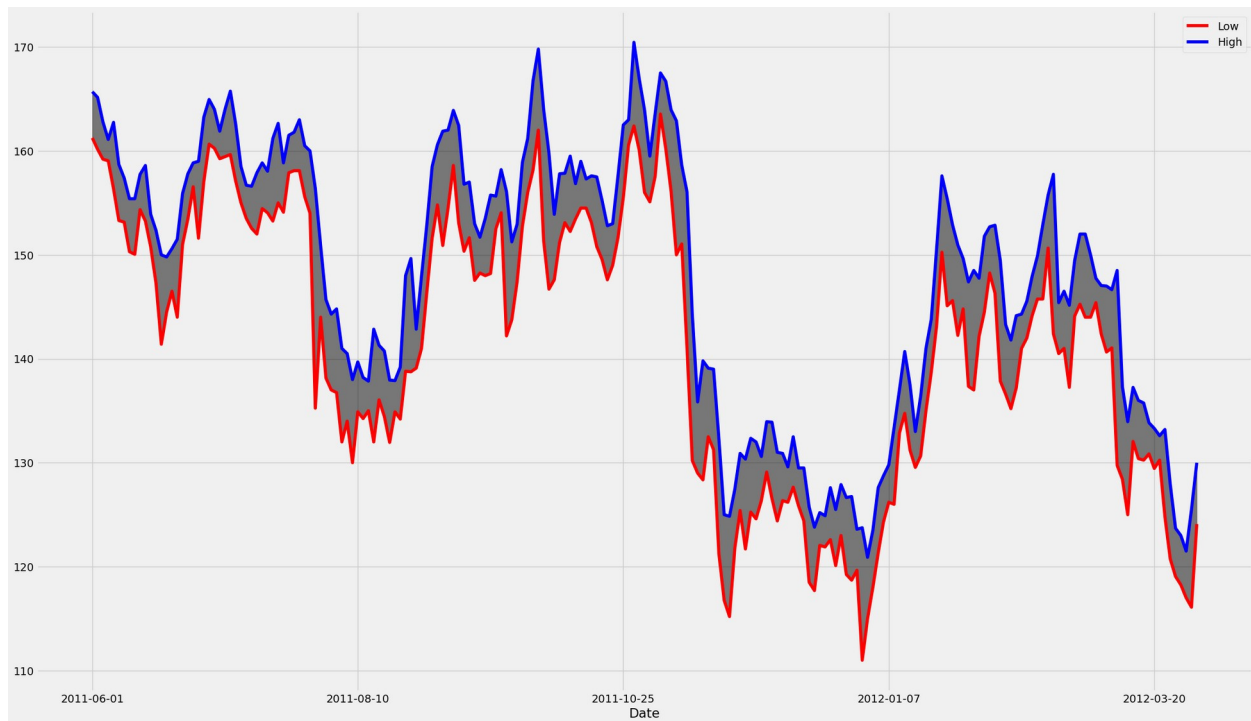


```
stockprice_oneyear.plot(x='Date',y=['Open','Close'],color=['r','b'],figsize=[25,15])
```

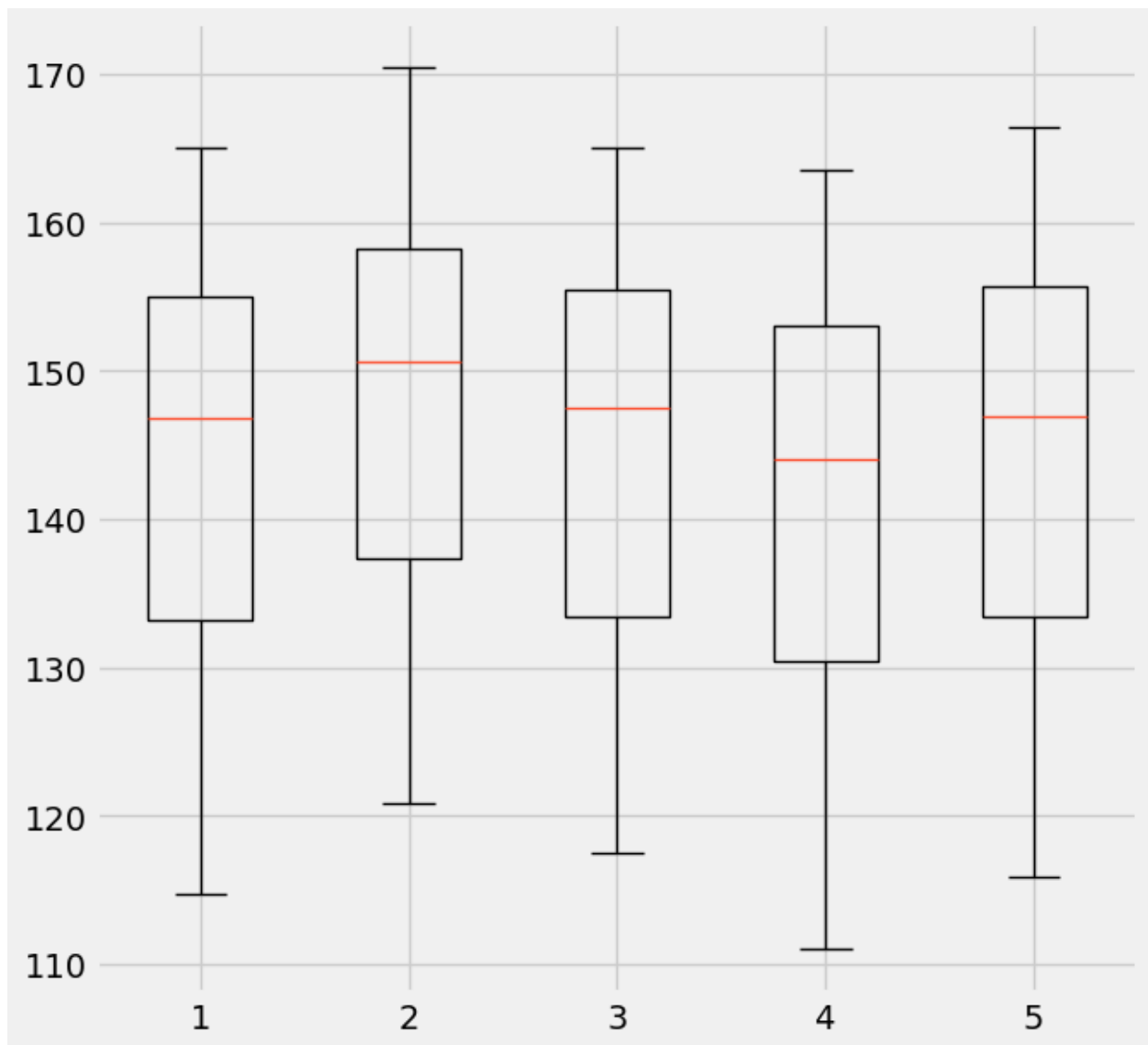
```
<AxesSubplot:xlabel='Date'>
```



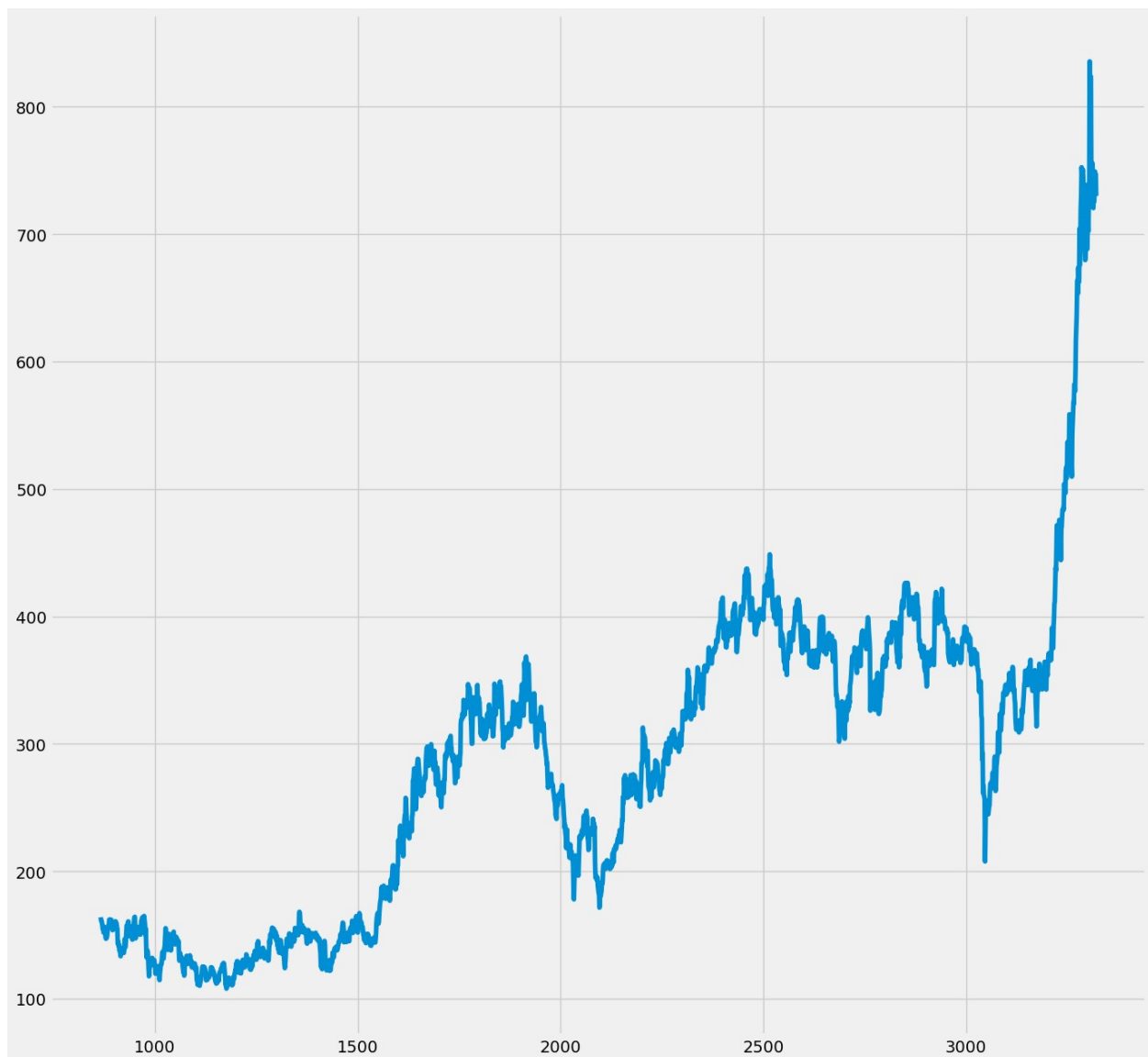
```
stockprice_oneyear.plot(x='Date',y=['Low','High'],color=['r','b'],figsize=[25,15])
plt.fill_between(stockprice_oneyear['Date'],stockprice_oneyear['Low'],
stockprice_oneyear['High'],color='k',alpha=0.5)
plt.show()
```



```
plt.figure(figsize=(7,7))
plt.boxplot(stockprice_oneyear[['Close', 'High', 'Open', 'Low', 'VWAP']])
plt.show()
```



```
#data.set_index('Date',drop=False,inplace=True)
data.Close.plot(figsize=(15,15))
<AxesSubplot:>
```



```
data['Tomorrow']=data['Close'].shift(-1)
data
```

	Date	Symbol	Series	Prev Close	Open	High
Low \						
866	2011-06-01	MUNDRAPORT	EQ	161.45	162.10	165.70
161.25						
867	2011-06-02	MUNDRAPORT	EQ	164.00	164.00	165.15
160.15						
868	2011-06-03	MUNDRAPORT	EQ	161.25	161.50	162.80
159.20						
869	2011-06-06	MUNDRAPORT	EQ	161.05	160.50	161.10
159.05						
870	2011-06-07	MUNDRAPORT	EQ	159.85	159.85	162.75
156.35						

...
3317	2021-04-26	ADANI	EQ	725.35	733.00	739.65	
728.90							
3318	2021-04-27	ADANI	EQ	730.75	735.00	757.50	
727.35							
3319	2021-04-28	ADANI	EQ	749.15	755.00	760.00	
741.10							
3320	2021-04-29	ADANI	EQ	746.25	753.20	765.85	
743.40							
3321	2021-04-30	ADANI	EQ	746.75	739.00	759.45	
724.50							

	Last	Close	VWAP	Volume	Turnover	Trades	\
866	163.50	164.00	164.08	2574106	4.223703e+13	19171.0	
867	161.15	161.25	162.17	1699298	2.755678e+13	16176.0	
868	161.00	161.05	161.02	1185817	1.909361e+13	14810.0	
869	160.00	159.85	160.09	546378	8.746905e+12	7071.0	
870	157.00	157.25	158.52	2193466	3.477027e+13	17865.0	
...	
3317	729.20	730.75	733.25	9390549	6.885658e+14	116457.0	
3318	748.60	749.15	747.67	20573107	1.538191e+15	236896.0	
3319	743.40	746.25	751.02	11156977	8.379106e+14	130847.0	
3320	746.40	746.75	753.06	13851910	1.043139e+15	153293.0	
3321	726.40	730.05	743.35	12600934	9.366911e+14	132141.0	

	Deliverable Volume	%Deliverble	Month	Year	is_quarter_end
Tomorrow					
866	1271255	0.4939	6	2011	1
161.25					
867	791462	0.4658	6	2011	1
161.05					
868	722154	0.6090	6	2011	1
159.85					
869	386144	0.7067	6	2011	1
157.25					
870	1425849	0.6500	6	2011	1
154.90					
...
...					
3317	838079	0.0892	4	2021	0
749.15					
3318	1779639	0.0865	4	2021	0
746.25					
3319	1342353	0.1203	4	2021	0
746.75					
3320	1304895	0.0942	4	2021	0
730.05					
3321	3514692	0.2789	4	2021	0

NaN

[2456 rows x 19 columns]

```
data['open-close'] = data['Open'] - data['Close']
data['low-high'] = data['Low'] - data['High']
#data['target'] = np.where(data['Close'].shift(-1) > data['Close'], 1,
0)
data['Target']=(data['Tomorrow']>data['Close']).astype(int)
data
```

	Date	Symbol	Series	Prev Close	Open	High
Low \						
866	2011-06-01	MUNDRAPORT	EQ	161.45	162.10	165.70
161.25						
867	2011-06-02	MUNDRAPORT	EQ	164.00	164.00	165.15
160.15						
868	2011-06-03	MUNDRAPORT	EQ	161.25	161.50	162.80
159.20						
869	2011-06-06	MUNDRAPORT	EQ	161.05	160.50	161.10
159.05						
870	2011-06-07	MUNDRAPORT	EQ	159.85	159.85	162.75
156.35						
...
.						
3317	2021-04-26	ADANI PORTS	EQ	725.35	733.00	739.65
728.90						
3318	2021-04-27	ADANI PORTS	EQ	730.75	735.00	757.50
727.35						
3319	2021-04-28	ADANI PORTS	EQ	749.15	755.00	760.00
741.10						
3320	2021-04-29	ADANI PORTS	EQ	746.25	753.20	765.85
743.40						
3321	2021-04-30	ADANI PORTS	EQ	746.75	739.00	759.45
724.50						
	Last	Close	VWAP	...	Trades	Deliverable Volume
%Deliverble \						
866	163.50	164.00	164.08	...	19171.0	1271255
0.4939						
867	161.15	161.25	162.17	...	16176.0	791462
0.4658						
868	161.00	161.05	161.02	...	14810.0	722154
0.6090						
869	160.00	159.85	160.09	...	7071.0	386144
0.7067						
870	157.00	157.25	158.52	...	17865.0	1425849
0.6500						
...
...						

```

3317  729.20  730.75  733.25  ...  116457.0      838079
0.0892
3318  748.60  749.15  747.67  ...  236896.0      1779639
0.0865
3319  743.40  746.25  751.02  ...  130847.0      1342353
0.1203
3320  746.40  746.75  753.06  ...  153293.0      1304895
0.0942
3321  726.40  730.05  743.35  ...  132141.0      3514692
0.2789

```

```

      Month  Year  is_quarter_end  Tomorrow  open-close  low-high
Target
866         6  2011              1    161.25      -1.90     -4.45
0
867         6  2011              1    161.05       2.75     -5.00
0
868         6  2011              1    159.85       0.45     -3.60
0
869         6  2011              1    157.25       0.65     -2.05
0
870         6  2011              1    154.90       2.60     -6.40
0
...         ...  ...              ...      ...      ...      ...
...
3317         4  2021              0    749.15       2.25    -10.75
1
3318         4  2021              0    746.25     -14.15    -30.15
0
3319         4  2021              0    746.75       8.75    -18.90
1
3320         4  2021              0    730.05       6.45    -22.45
0
3321         4  2021              0         NaN       8.95    -34.95
0

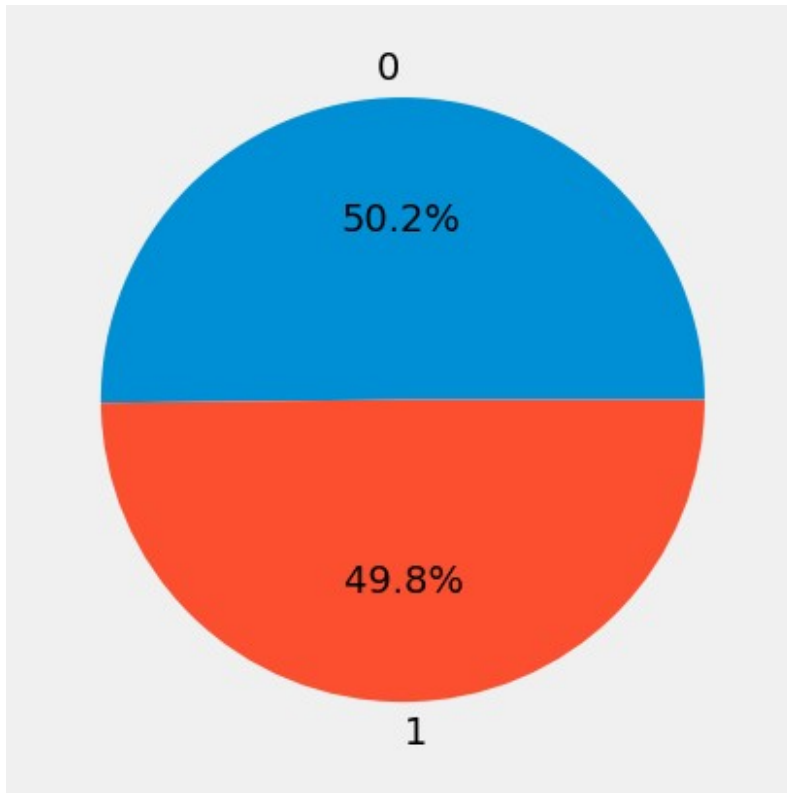
```

```
[2456 rows x 22 columns]
```

```

plt.pie(data['Target'].value_counts().values, labels=[0, 1],
autopct='%1.1f%%')
plt.show()

```



```
sns.heatmap(data.corr() > 0.9, annot=True, cbar=False)  
plt.show()
```

Prev Close	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
High	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Last	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
VWAP	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Turnover	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Deliverable Volume	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Month	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
is_quarter_end	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
open-close	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Target	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverable	Month	Year	is_quarter_end	Tomorrow	open-close	low-high	Target	

```
X = data[['open-close', 'low-high']]
Y = data['Target']
```

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
X_train=ss.fit_transform(X_train) # xtrain = training input samples
X_test=ss.transform(X_test) # xtest - testing input samples

clfr=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)

clfr.fit(X_train,Y_train)

RandomForestClassifier(criterion='entropy', n_estimators=10,
random_state=0)
```

```

clfr1=RandomForestClassifier(n_estimators=10,criterion='gini',random_s
tate=0)

clfr1.fit(X_train,Y_train)

RandomForestClassifier(n_estimators=10, random_state=0)

from sklearn.metrics import
confusion_matrix,classification_report,accuracy_score

ypre=clfr.predict(X_test)# entropy ypre calculation
yprel=clfr1.predict(X_test)# gini ypre calculation

print('entropy Accuracy Score:')
accuracy_score(Y_test,ypre)*100

entropy Accuracy Score:
50.81300813008131

print('gini Accuracy Score:')
accuracy_score(Y_test,yprel)*100

gini Accuracy Score:
50.0

print('entropy - confusion matrix\n-----\n')
print(confusion_matrix(Y_test,ypre))
print('gini - confusion matrix\n-----\n')
print(confusion_matrix(Y_test,yprel))

entropy - confusion matrix
-----

[[139  94]
 [148 111]]
gini - confusion matrix
-----

[[145  88]
 [158 101]]

print('entropy result\n-----')
print(classification_report(Y_test,ypre))
print('gini index result\n-----')
print(classification_report(Y_test,yprel))

entropy result
-----

```

	precision	recall	f1-score	support
0	0.48	0.60	0.53	233
1	0.54	0.43	0.48	259
accuracy			0.51	492
macro avg	0.51	0.51	0.51	492
weighted avg	0.51	0.51	0.51	492

gini index result

	precision	recall	f1-score	support
0	0.48	0.62	0.54	233
1	0.53	0.39	0.45	259
accuracy			0.50	492
macro avg	0.51	0.51	0.50	492
weighted avg	0.51	0.50	0.49	492

```
data = pd.DataFrame({'Actual': Y_test, 'Predicted': ypre})
data.head()
```

	Actual	Predicted
2069	1	0
1186	0	0
3084	0	0
1744	0	1
2077	1	1

```
data = pd.DataFrame({'Actual': Y_test, 'Predicted': ypre1})
data.head()
```

	Actual	Predicted
2069	1	0
1186	0	0
3084	0	0
1744	0	0
2077	1	1

```
log_reg = LogisticRegression()
log_reg.fit(X_train,Y_train)
```

```
LogisticRegression()
```

```
ypre2 = log_reg.predict(X_test)
```

```
print(confusion_matrix(Y_test,ypre2))
```

```
[[155  78]
 [160  99]]
```

```
print(classification_report(Y_test,ypre2))
```

	precision	recall	f1-score	support
0	0.49	0.67	0.57	233
1	0.56	0.38	0.45	259
accuracy			0.52	492
macro avg	0.53	0.52	0.51	492
weighted avg	0.53	0.52	0.51	492

```
print(accuracy_score(Y_test,ypre2))
```

```
0.516260162601626
```

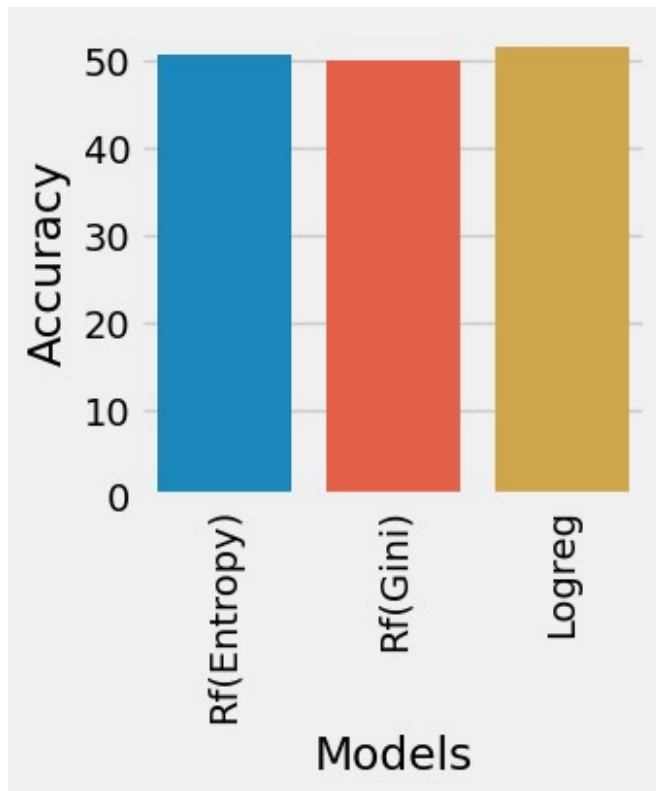
```
data = pd.DataFrame({'Actual': Y_test, 'Predicted': ypre2})  
data.head()
```

	Actual	Predicted
2069	1	1
1186	0	0
3084	0	0
1744	0	0
2077	1	0

```
data=pd.DataFrame({'Models': ['Rf(Entropy)', 'Rf(Gini)', 'Logreg'],  
                   'Accuracy':  
[accuracy_score(Y_test,ypre)*100,accuracy_score(Y_test,ypre1)*100,  
  accuracy_score(Y_test,ypre2)*100]})  
data
```

	Models	Accuracy
0	Rf(Entropy)	50.813008
1	Rf(Gini)	50.000000
2	Logreg	51.626016

```
plt.figure(figsize=(3,3))  
sns.barplot(data['Models'],data['Accuracy'])  
plt.xticks(rotation=90)  
plt.show()
```



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

```
data_train=data[data.Date<'2018']
```

```
-----  
-----  
AttributeError                                Traceback (most recent call  
last)
```

```
~\AppData\Local\Temp\ipykernel_12048\2009902171.py in <module>
```

```
----> 1 data_train=data[data.Date<'2018']
```

```
~\anaconda3\lib\site-packages\pandas\core\generic.py in
```

```
__getattr__(self, name)
```

```
5573         ):
```

```
5574             return self[name]
```

```
-> 5575         return object.__getattribute__(self, name)
```

```
5576
```

```
5577     def __setattr__(self, name: str, value) -> None:
```

```
AttributeError: 'DataFrame' object has no attribute 'Date'
```

```
import tensorflow
```

```
from sklearn.preprocessing import MinMaxScaler
```