

```
In [13]: # Importing pandas, datetime, numpy aliasing
import pandas as pd
import datetime as dt
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: # Converting Date to DateTime dtype.
NFT88_20_1 = pd.read_csv('NIFTY_2008_2020.csv', parse_dates=['Date'])
NFT88_20_1.drop(labels= 'Time',axis=1, inplace= True) #Dropping the time Column.
```

```
In [4]: # checking the columes of the Dataframe.
NFT88_20_1.columns
```

```
Out[4]: Index(['Instrument', 'Date', 'Open', 'High', 'Low', 'Close'], dtype='object')
```

```
In [5]: NFT88_20_1.head(3) # Looking for the top 3 rows in the column.
```

	Instrument	Date	Open	High	Low	Close
0	NIFTY	2008-01-01	6138.60	6154.60	6138.60	6148.90
1	NIFTY	2008-01-01	6149.75	6149.75	6132.80	6132.80
2	NIFTY	2008-01-01	6138.25	6138.25	6127.95	6127.95

```
In [6]: NFT88_20_1.describe() # Looking the whole description of the project like Counts, mean, std(standard deviation) etc.
```

	Open	High	Low	Close
count	1.178891e+06	1.178891e+06	1.178891e+06	1.178891e+06
mean	7.515746e+03	7.517452e+03	7.514029e+03	7.515726e+03
std	2.538014e+03	2.538359e+03	2.537660e+03	2.538002e+03
min	2.255200e+03	2.256100e+03	2.253950e+03	2.254250e+03
25%	5.351950e+03	5.353200e+03	5.350800e+03	5.352000e+03
50%	7.534550e+03	7.536250e+03	7.532700e+03	7.534550e+03
75%	9.802400e+03	9.804300e+03	9.800750e+03	9.802400e+03
max	1.313715e+04	1.314585e+04	1.313275e+04	1.313755e+04

```
In [8]: # Grouping the data using Groupby and Grouper with frequency MS(Months), key 'Date'. Averaging the data according to months.
NFT88_20_1 = NFT88_20_1.groupby(pd.Grouper(freq='MS', key='Date')).mean()
```

```
In [9]: # Looking into the whole data.
NFT88_20_1.info()

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 155 entries, 2008-01-01 to 2020-11-01
Freq: MS
Data columns (total 4 columns):
# Column Non-Null Count Dtype
---  ---
0 Open 155 non-null float64
1 High 155 non-null float64
2 Low 155 non-null float64
3 Close 155 non-null float64
dtypes: float64(4)
memory usage: 6.1 KB
```

```
In [10]: NFT88_20_1.head(3)
```

	Open	High	Low	Close
Date				
2008-01-01	5747.703349	5750.116399	5745.175802	5747.610380
2008-02-01	5186.040276	5187.952390	5184.127095	5186.050358
2008-03-01	4771.493888	4773.366912	4769.534124	4771.424047

```
In [12]: NFT88_20_1.describe()
```

	Open	High	Low	Close
count	155.000000	155.000000	155.000000	155.000000
mean	7430.768255	7432.477989	7429.049268	7430.748744
std	2566.280595	2566.636542	2565.915355	2566.267969
min	2798.794688	2799.803966	2797.809903	2798.827374
25%	5300.528471	5301.752934	5299.315641	5300.502228
50%	7217.889790	7220.010000	7215.661187	7217.777873
75%	9537.618083	9542.278977	9532.920581	9537.509608
max	12573.733720	12577.140386	12570.338654	12573.759834

```
In [30]: def rank_performance (stock_price):
if stock_price <= $390.528471:
    return 'poor'
elif stock_price <= 7217.889790:
    return 'Average'
else:
    return 'Good'
```

```
In [32]: NFT88_20_1['Open'].apply(rank_performance)
```

```
Out[32]: Date
2008-01-01 Average
2008-02-01 poor
2008-03-01 poor
2008-04-01 poor
2008-05-01 poor
.....
2020-07-01 Good
2020-08-01 Good
2020-09-01 Good
2020-10-01 Good
2020-11-01 Good
Freq: MS, Name: Open, Length: 155, dtype: object
```

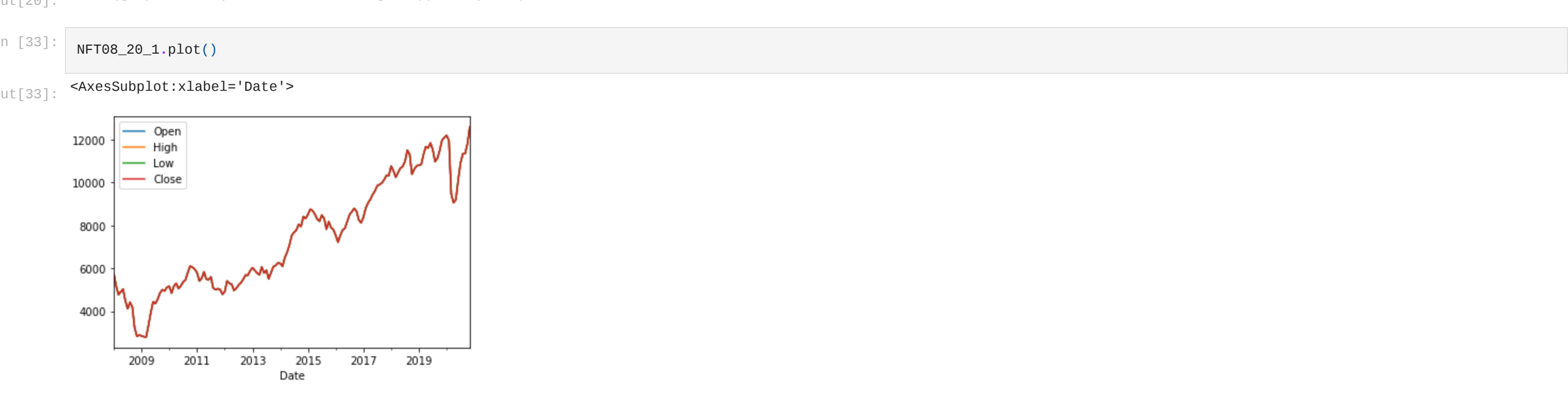
```
In [15]: plt.style.available
```

```
Out[15]: ['Solarize_Light2',
'classic_test_patch',
'_mpl-gallery',
'_mpl-gallery-nogrid',
'bmh',
'classic',
'dark_background',
'fast',
'fivethirtyeight',
'ggplot',
'grayscale',
'seaborn',
'seaborn-bright',
'seaborn-colorblind',
'seaborn-dark',
'seaborn-dark-palette',
'seaborn-darkgrid',
'seaborn-deep',
'seaborn-muted',
'seaborn-notebook',
'seaborn-paper',
'seaborn-pastel',
'seaborn-poster',
'seaborn-talk',
'seaborn-ticks',
'seaborn-white',
'seaborn-whitegrid',
'tableau-colorblind9']
```

```
In [20]: NFT88_20_1.columns
```

```
Out[20]: Index(['Open', 'High', 'Low', 'Close'], dtype='object')
```

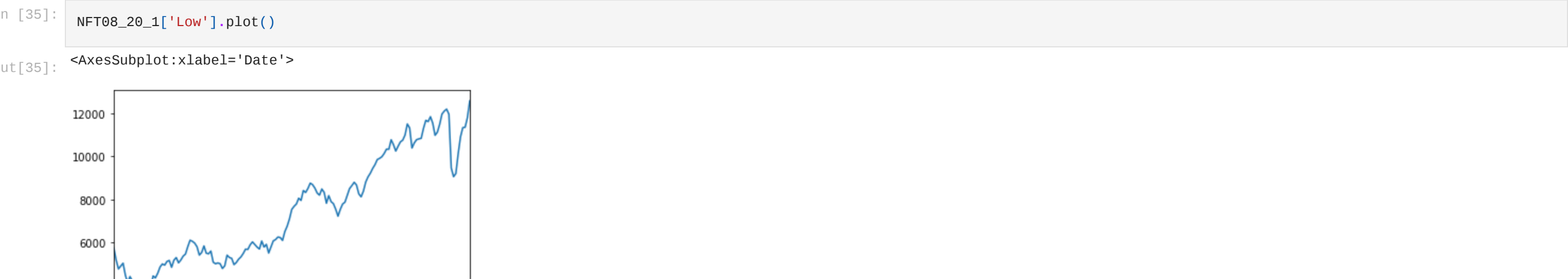
```
In [33]: NFT88_20_1.plot()
```



```
In [34]: NFT88_20_1['High'].plot()
```



```
In [35]: NFT88_20_1['Low'].plot()
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```