

LetsGrowMore

Name : Varsha Reddy

Task : Stock Market Prediction and Forecasting using Stacked LSTM

Level : Beginner

Importing required libraries for Data Manipulation

```
In [3]: import numpy as np
import pandas as pd
```

Importing required libraries for Data Visualization

```
In [5]: import seaborn as sns
import matplotlib.pyplot as plt
import math
```

Importing Dataset

```
In [7]: Dataset = "https://raw.githubusercontent.com/mwitiderrick/stockprice/master/NSE-TATAGLOBAL.csv"

In [8]: df = pd.read_csv(Dataset)
```

```
In [9]: df
```

```
Out[9]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.80	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.30	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.10	123.00	121.05	121.10	121.55	658666	803.56

2035 rows x 8 columns

```
In [10]: df.describe()
```

```
Out[10]:
```

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035.000000	2035.000000	2035.000000	2035.000000	2035.000000	2.035000e+03	2035.000000
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000

```
In [11]: df.head()
```

```
Out[11]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55

```
In [12]: df.tail()
```

```
Out[12]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2030	2010-07-27	117.6	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.1	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.8	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.3	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.1	123.00	121.05	121.10	121.55	658666	803.56

```
In [13]: df.dtypes
```

```
Out[13]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035	2035	2035	2035	2035	2035	2035	2035
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565	
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877	
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000	
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000	
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000	
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000	
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000	

```
In [14]: df.isnull().sum()

Out[14]:
```

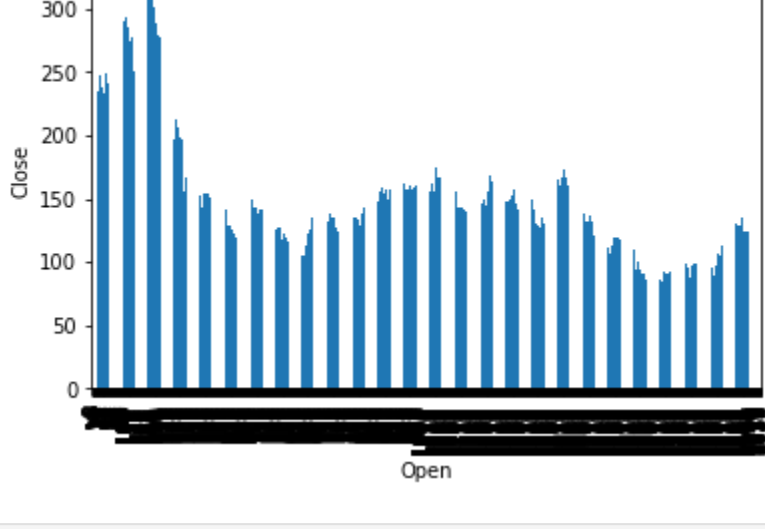
	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035	2035	2035	2035	2035	2035	2035	2035
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565	
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877	
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000	
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000	
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000	
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000	
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000	

```
In [15]: df['Date'].value_counts()

Out[15]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035	2035	2035	2035	2035	2035	2035	2035
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565	
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877	
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000	
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000	
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000	
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000	
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000	

```
In [16]: df['Open'].plot.bar()
plt.xlabel("Open")
plt.ylabel("Close")
plt.show()
```



```
In [17]: plt.figure(figsize=(20,8))
df.plot.hist()
```

```
Out[17]:
```

<AxesSubplot:ylabel='Frequency'>

<Figure size 1440x576 with 0 Axes>

```
In [18]: dfset_set = df.filter(['Close'])
dfset = df.values
train_df_len=math.floor(len(df)*15)
train_df_len
```

```
Out[18]:
```

30525

```
In [19]: dfset
```

```
Out[19]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.80	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.30	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.10	123.00	121.05	121.10	121.55	658666	803.56

```
In [20]: df = df.iloc[:, 0:10]
df
```

```
Out[20]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.80	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.30	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.10	123.00	121.05	121.10	121.55	658666	803.56

2035 rows x 8 columns

```
In [21]: train_set = df.iloc[:,1:2].values
train_set
```

```
Out[21]:
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.80	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.30	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.10	123.00	121.05	121.10	121.55	658666	803.56

```
In [22]: from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
```

```
In [23]: df_train_scale = scaler.fit_transform(train_set)
```

```
In [34]: features_set = []
```

```
In [35]: labels = []
```

```
In [36]: for i in range(60, 586):
    features_set.append(df_train_scale[i - 60:i, 0])
    labels.append(df_train_scale[i, 0])
```

```
In [37]: features_set, labels = np.array(features_set), np.array(labels)
features_set = np.reshape(features_set, (features_set.shape[0], features_set.shape[1], 1))
features_set.shape
```

```
Out[37]:
```

(526, 60, 1)

Importing Tensorflow and Keras libraries

```
In [40]: import tensorflow as tf
```

```
In [41]: from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import LSTM
from keras.layers import Dropout
```

```
In [42]: model = Sequential()
```

```
In [43]: model.compile(optimizer='adam', loss='mean_squared_error')
```

```
In [44]: df_test_complete = pd.read_csv(Dataset)
df_test_processed = df_test_complete.iloc[:,1:2]
df_test_processed
```

```
Out[44]:
```

	Open
0	234.05
1	234.55
2	240.00
3	233.30
4	233.55
...	...
2030	117.60
2031	