

```
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors,
870     else:
871         # Binary mode
--> 872         handle = open(handle, ioargs.mode)
873         handles.append(handle)
874
FileNotFoundError: [Errno 2] No such file or directory: 'SBIN.NS'
```

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import yfinance as yf

start = '2012-01-01'
end = '2022-12-21'
stock = 'GOOG'

data = yf.download(stock,start,end)
```

```
[*****100%*****] 1 of 1 completed
```

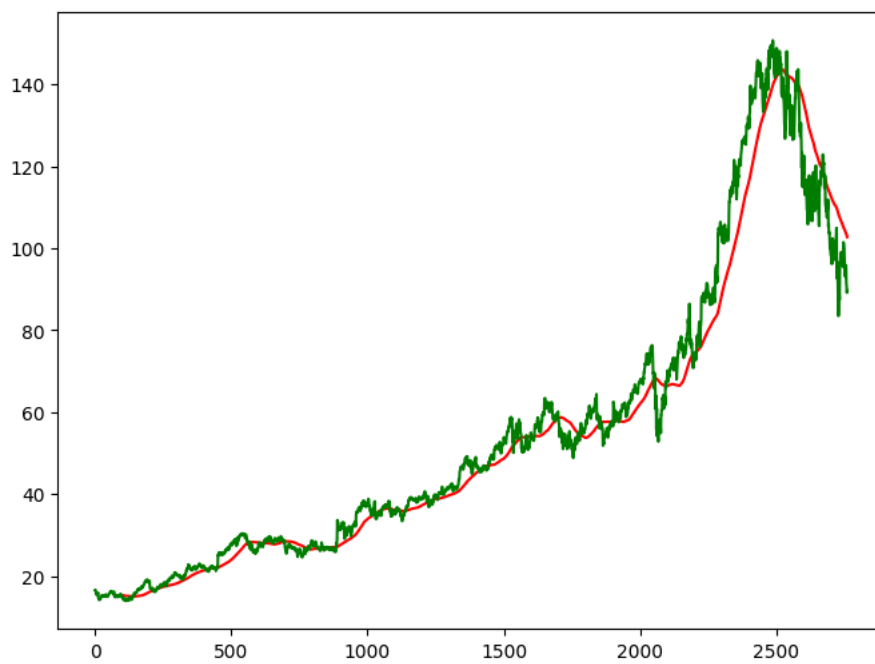
```
In [ ]: data.reset_index(inplace=True)
data
```

```
Out [6]:
```

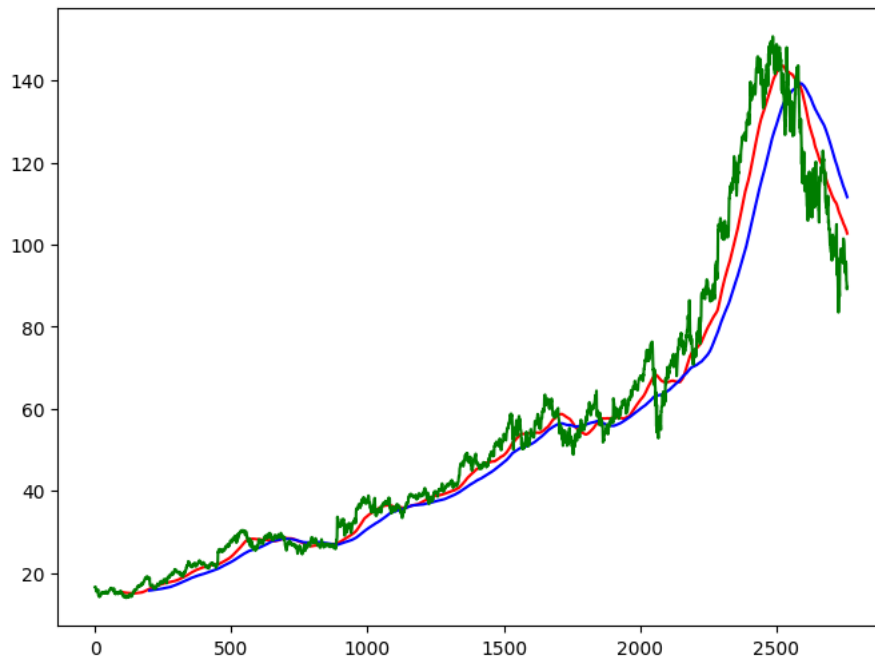
	Date	Open	High	Low	Close	Adj Close	Volume
0	2012-01-03	16.262545	16.641375	16.248346	16.573130	16.554291	147611217
1	2012-01-04	16.563665	16.693678	16.453827	16.644611	16.625692	114989399
2	2012-01-05	16.491436	16.537264	16.344486	16.413727	16.395069	131808205
3	2012-01-06	16.417213	16.438385	16.184088	16.189817	16.171415	108119746
4	2012-01-09	16.102144	16.114599	15.472754	15.503389	15.485767	233776981
...	...	...	...	...	...	...	...
2756	2022-12-14	95.540001	97.220001	93.940002	95.309998	95.201660	26452900
2757	2022-12-15	93.540001	94.029999	90.430000	91.199997	91.096336	28298800
2758	2022-12-16	91.199997	91.750000	90.010002	90.860001	90.756721	48485500
2759	2022-12-19	90.879997	91.199997	88.925003	89.150002	89.048668	23020500
2760	2022-12-20	88.730003	89.779999	88.040001	89.629997	89.528122	21976800

2761 rows × 7 columns

```
In [ ]: moving_avg_100_days = data.Close.rolling(100).mean()
plt.figure(figsize=(8,6))
plt.plot(moving_avg_100_days,'r')
plt.plot(data.Close,'g')
plt.show()
```



```
In [ ]: moving_avg_200_days = data.Close.rolling(200).mean()
plt.figure(figsize=(8,6))
plt.plot(moving_avg_100_days,'r')
plt.plot(moving_avg_200_days,'b')
plt.plot(data.Close,'g')
plt.show()
```



```
In [ ]: data.dropna(inplace=True)
data_train = pd.DataFrame(data.Close[0: int(len(data)*0.80)])
data_test = pd.DataFrame(data.Close[int(len(data)*0.80): len(data)])
```

```
In [ ]: data_train.shape[0]
```

Out [91]: 2208

```
In [ ]: data_test.shape[0]
```

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

```
In [ ]: data_train_scale = scalar.fit_transform(data_train)
```

```
In [ ]: x = []
        y = []

        for i in range(100 , data_train_scale.shape[0]):
            x.append(data_train_scale[i-100:i])
            y.append(data_train_scale[i,0])
```

```
In [ ]: # x, y = np.array(x) , np.array(y)
```

```
In [ ]: # from keras.layers import Dense,Dropout,LSTM
        # from keras.models import Sequential
```

```
In [ ]: x, y = np.array(x) , np.array(y)
        from keras.layers import Dense, Dropout, LSTM
        from keras.models import Sequential

        model = Sequential()
        model.add(LSTM(units = 50 , activation = 'relu' , return_sequences = True , input_shape = ((x.shape[1],1))))
        model.add(Dropout(0.2))

        model.add(LSTM(units = 60 , activation = 'relu' , return_sequences = True))
        model.add(Dropout(0.3))

        model.add(LSTM(units = 80, activation = 'relu' , return_sequences = True))
        model.add(Dropout(0.4))

        model.add(LSTM(units = 120 , activation= 'relu'))
        model.add(Dropout(0.5))

        model.add(Dense(units=1))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an `input_shape`/`input_dim` argument
to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)
```

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

```
In [ ]: model.fit(x,y, epochs= 50 , batch_size= 32 , verbose= 1)
```

```
Epoch 1/50
[1m66/66][0m [32m[0m [37m[0m [1m25s[0m 268ms/step - loss: 0.0610
Epoch 2/50
[1m66/66][0m [32m[0m [37m[0m [1m22s[0m 297ms/step - loss: 0.0066
Epoch 3/50
[1m66/66][0m [32m[0m [37m[0m [1m18s[0m 257ms/step - loss: 0.0056
Epoch 4/50
[1m66/66][0m [32m[0m [37m[0m [1m22s[0m 287ms/step - loss: 0.0054
Epoch 5/50
[1m66/66][0m [32m[0m [37m[0m [1m17s[0m 260ms/step - loss: 0.0045
Epoch 6/50
[1m66/66][0m [32m[0m [37m[0m [1m17s[0m 257ms/step - loss: 0.0056
Epoch 7/50
[1m66/66][0m [32m[0m [37m[0m [1m21s[0m 264ms/step - loss: 0.0051
Epoch 8/50
[1m66/66][0m [32m[0m [37m[0m [1m20s[0m 262ms/step - loss: 0.0039
Epoch 9/50
[1m66/66][0m [32m[0m [37m[0m [1m20s[0m 262ms/step - loss: 0.0041
Epoch 10/50
[1m66/66][0m [32m[0m [37m[0m [1m21s[0m 270ms/step - loss: 0.0036
Epoch 11/50
[1m66/66][0m [32m[0m [37m[0m [1m18s[0m 277ms/step - loss: 0.0039
Epoch 12/50
[1m66/66][0m [32m[0m [37m[0m [1m19s[0m 262ms/step - loss: 0.0036
Epoch 13/50
[1m66/66][0m [32m[0m [37m[0m [1m21s[0m 273ms/step - loss: 0.0033
Epoch 14/50
```

Out [98]:

```
model.summary()
```

```
Model: "sequential_8"
```

Layer (type)	Output Shape	Param #
lstm_19 (LSTM)	(None, 100, 50)	10,400
dropout_16 (Dropout)	(None, 100, 50)	0
lstm_20 (LSTM)	(None, 100, 60)	26,640
dropout_17 (Dropout)	(None, 100, 60)	0
lstm_21 (LSTM)	(None, 100, 80)	45,120
dropout_18 (Dropout)	(None, 100, 80)	0
lstm_22 (LSTM)	(None, 120)	96,480



