Crypto stock prediction using various influence Factors

Results:

Open	High	Low	Close	Adj Close	Volume
244.4551	244.4551	236.0298	237.0184	237.0184	6361000
238.1014	242.0557	237.4556	239.9941	239.9941	5642000
241.1814	241.4397	234.8774	239.7904	239.7904	7026600
235.7467	238.8912	234.1968	235.3443	235.3443	5767400
234.7283	235.081	229.7557	232.4184	232.4184	6054900
232.5475	234.9767	229.4576	230.004	230.004	5006200
232.2842	238.4889	231.6583	238.3746	238.3746	7215600
237.6394	243.3026	234.7283	241.5986	241.5986	6795400
239.4427	240.4362	236.6657	238.012	238.012	5583600
238.5236	246.2732	236.308	245.9056	245.9056	7527100
244.6438	252.7958	244.3011	248.8018	248.8018	8533700
249.378	251.4297	247.391	249.6811	249.6811	5547200
249.1793	250.1431	245.3443	248.3994	248.3994	7298100
247.2419	252.1649	246.5415	251.1118	251.1118	5875100
251.8569	253.6801	251.1167	253.4267	253.4267	7137300
253.2727	253.3522	246.154	247.54	247.54	6711100
243.6304	247.5301	241.7427	243.6652	243.6652	6132500
245.7616	246.7402	239.7258	240.933	240.933	8015800
239.8997	243.3671	238.0865	241.693	241.693	7073900
244.162	245.841	243.1585	243.7099	243.7099	4624200
241.4844	246.8942	239.1943	239.9444	239.9444	4054700
240.1679	240.1679	233.7745	235.548	235.548	6659300

In [4]:
```
data
```

Out[4]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-10-28	6482.660156	6502.279785	6447.910156	6486.390137	6486.390137	3445190000
2018-10-29	6492.350098	6503.600098	6306.990234	6332.629883	6332.629883	4199910000
2018-10-30	6337.040039	6364.990234	6310.140137	6334.270020	6334.270020	3781100000
2018-10-31	6336.990234	6349.160156	6316.879883	6317.609863	6317.609863	4191240000
2018-11-01	6318.140137	6547.140137	6311.830078	6377.779785	6377.779785	3789400000
...	...	...	...	...	...	...
2023-10-24	33077.304688	35150.433594	32880.761719	33901.527344	33901.527344	44934999645
2023-10-25	33916.042969	35133.757813	33709.109375	34502.820313	34502.820313	25254318008
2023-10-26	34504.289063	34832.910156	33762.324219	34156.648438	34156.648438	19427195376
2023-10-27	34156.500000	34238.210938	33416.886719	33909.800781	33909.800781	16418032871
2023-10-28	33907.722656	34155.898438	33875.285156	34095.496094	34095.496094	14318646272

1827 rows × 6 columns

In [1]:

```
import numpy as np
import pandas as pd
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns; sns.set_style("whitegrid")
from plotly import tools
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
```

In [3]:

```
# load data set
data = pd.read_csv(r'C:\Users\Rohit94\Documents\project_2023\Crypto Currency\Crypto Currency\BTC-USD.csv',index
```

In [4]:

```
data
```

Out[4]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-10-28	6482.660156	6502.279785	6447.910156	6486.390137	6486.390137	3445190000
2018-10-29	6492.350098	6503.600098	6306.990234	6332.629883	6332.629883	4199910000
2018-10-30	6337.040039	6364.990234	6310.140137	6334.270020	6334.270020	3781100000
2018-10-31	6336.990234	6349.160156	6316.879883	6317.609863	6317.609863	4191240000
2018-11-01	6318.140137	6547.140137	6311.830078	6377.779785	6377.779785	3789400000
...	...	...	...	...	...	...
2023-10-24	33077.304688	35150.433594	32880.761719	33901.527344	33901.527344	44934999645
2023-10-25	33916.042969	35133.757813	33709.109375	34502.820313	34502.820313	25254318008
2023-10-26	34504.289063	34832.910156	33762.324219	34156.648438	34156.648438	19427195376
2023-10-27	34156.500000	34238.210938	33416.886719	33909.800781	33909.800781	16418032871
2023-10-28	33907.722656	34155.898438	33875.285156	34095.496094	34095.496094	14318646272

1827 rows × 6 columns

In [5]:

```
data.describe()
```

Out[5]:

	Open	High	Low	Close	Adj Close	Volume
count	1827.000000	1827.000000	1827.000000	1827.000000	1827.000000	1.827000e+03
mean	23353.637436	23890.788310	22773.227257	23366.457316	23366.457316	2.847932e+10
std	16211.256352	16628.789830	15726.714406	16204.022144	16204.022144	1.856998e+10
min	3236.274658	3275.377930	3191.303467	3236.761719	3236.761719	3.445190e+09
25%	9268.348633	9403.769531	9139.787110	9271.754394	9271.754394	1.641349e+10
50%	20208.769531	20647.289063	19801.800781	20231.261719	20231.261719	2.545047e+10
75%	33911.421875	34871.984375	32530.266601	33905.664062	33905.664062	3.615833e+10
max	67549.734375	68789.625000	66382.062500	67566.828125	67566.828125	3.509679e+11

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1827 entries, 2018-10-28 to 2023-10-28
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Open       1827 non-null   float64
 1   High       1827 non-null   float64
 2   Low        1827 non-null   float64
 3   Close      1827 non-null   float64
 4   Adj Close  1827 non-null   float64
 5   Volume     1827 non-null   int64  
dtypes: float64(5), int64(1)
memory usage: 99.9 KB
```

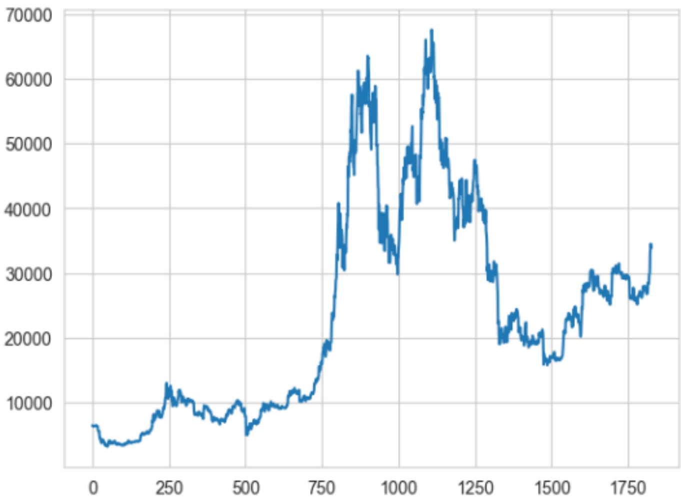
In [7]:
```
# check if data set contains missing values
print(data.isnull().sum())
```

```
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

In [8]:
```
data1=data.reset_index()['Open']
```

In [9]:
```
import matplotlib.pyplot as plt
plt.plot(data1)
```

Out[9]: [<matplotlib.lines.Line2D at 0x1de6edcbf40>]



In [10]:
```
data2=data.reset_index()['Close']
```

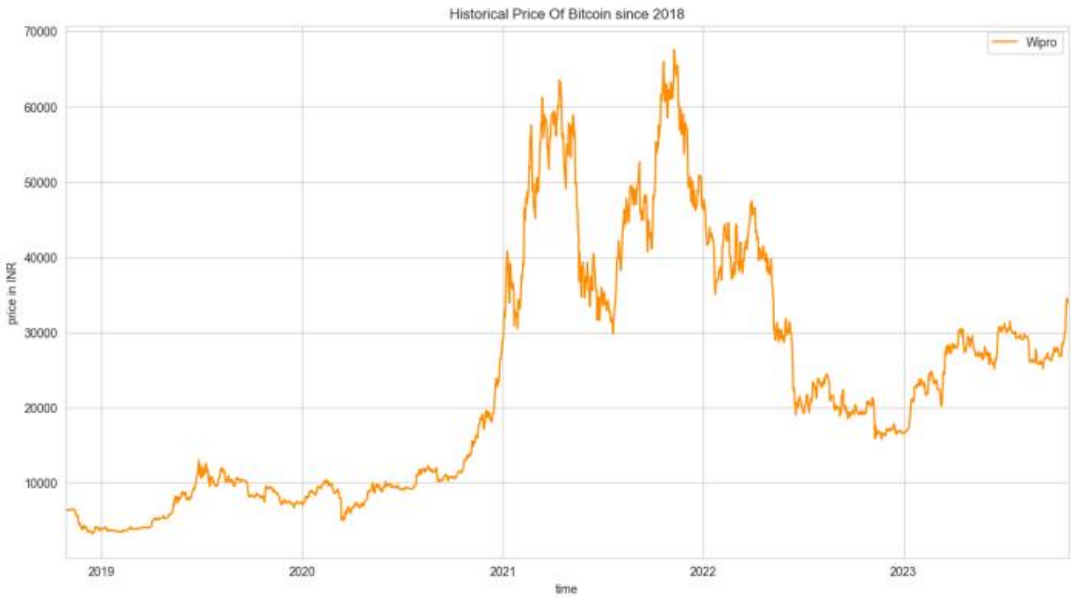
In [11]:

```
import matplotlib.pyplot as plt
plt.plot(data1)
plt.plot(data2)
```

Out[11]: [<matplotlib.lines.Line2D at 0x1de6ee1bc70>]

In [12]:

```
plt.figure(figsize=(15,8))
(data['Open']).plot(color='darkorange', label='Wipro')

plt.xlabel('time')
plt.ylabel('price in INR')
plt.title('Historical Price Of Bitcoin since 2018')
plt.legend()
plt.show()
```

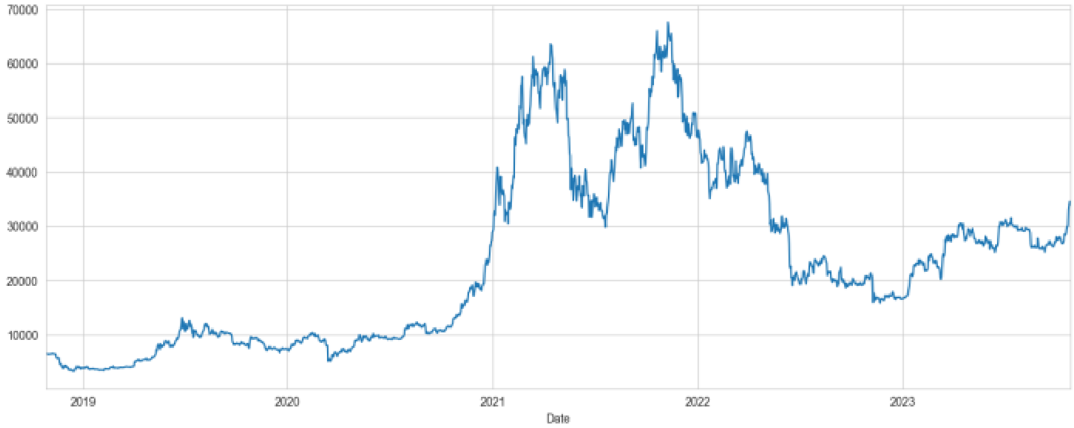


In [13]:

```python
data['Open'].plot(figsize=(16,6))
```

Out[13]:

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



In [14]:

```python
plt.figure(figsize=(15, 12))
plt.subplot(4,2,1)
plt.plot(data[['Open','High','Low','Close']])
plt.ylabel('price in USD')
plt.title('Historical daily average price of Wipro')
plt.legend(['Open','High','Low','Close'])
```

Out[14]:

```
<matplotlib.legend.Legend at 0x1de6f17dd00>
```

In [ ]:

```python

```

In [15]:

```python
data['Open'].plot(figsize=(16,6))
data.rolling(window=30).mean()['Close'].plot()
```

Out[15]:

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

In [18]:
```
data.columns
```
Out[18]:
```
Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```
In [19]:
```
#data = data.dropna().drop(['Symbol','Series','Prev Close','VWAP','Turnover','Turnover','Trades','Deliverable \
```
In [20]:
```
data
```
Out[20]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-10-28	6482.660156	6502.279785	6447.910156	6486.390137	6486.390137	3445190000
2018-10-29	6492.350098	6503.600098	6306.990234	6332.629883	6332.629883	4199910000
2018-10-30	6337.040039	6364.990234	6310.140137	6334.270020	6334.270020	3781100000
2018-10-31	6336.990234	6349.160156	6316.879883	6317.609863	6317.609863	4191240000
2018-11-01	6318.140137	6547.140137	6311.830078	6377.779785	6377.779785	3789400000
...	...	...	...	...	...	...
2023-10-24	33077.304688	35150.433594	32880.761719	33901.527344	33901.527344	44934999645
2023-10-25	33916.042969	35133.757813	33709.109375	34502.820313	34502.820313	25254318008
2023-10-26	34504.289063	34832.910156	33762.324219	34156.648438	34156.648438	19427195376
2023-10-27	34156.500000	34238.210938	33416.886719	33909.800781	33909.800781	16418032871
2023-10-28	33907.722656	34155.898438	33875.285156	34095.496094	34095.496094	14318646272

1827 rows × 6 columns

In [21]:
```
import datetime as dt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.callbacks import EarlyStopping
```
In [22]:
```
sc = MinMaxScaler()
train_set=sc.fit_transform(data['Close'][:2456].values.reshape(-1,1))
```
In [23]:
```
past_days = 30
```
In [24]:
```
# preparing independent and dependent features
def prepare_data(timeseries_data, n_features):
        X, y =[],[]
        for i in range(len(timeseries_data)):
                # find the end of this pattern
                end_ix = i + n_features
                # check if we are beyond the sequence
                if end_ix > len(timeseries_data)-1:
                        break
                # gather input and output parts of the pattern
                seq_x, seq_y = timeseries_data[i:end_ix], timeseries_data[end_ix]
                X.append(seq_x)
                y.append(seq_y)
        return np.array(X), np.array(y)
```
In [25]:
```
# define input sequence
timeseries_data = data['Close'][:6100].tolist()
# choose a number of time steps
n_steps = past_days
# split into samples
X, y = prepare_data(timeseries_data, n_steps)
```
In [26]:
```
X.shape
```
Out[26]:
```
(1797, 30)
```
In [27]:
```
# reshape from [samples, timesteps] into [samples, timesteps, features]
n_features = 1
X = X.reshape((X.shape[0], X.shape[1], n_features))
```
In [28]:
```
# define model
callback = [EarlyStopping(monitor='loss', mode='auto',)]
model = Sequential()
model.add(LSTM(64, activation='relu', return_sequences=True, input_shape=(n_steps, n_features)))
model.add(LSTM(64, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse',)
# fit model
model.fit(X, y, epochs=150, verbose=1)
```

```
Epoch 1/150
57/57 [==============================] - 2s 10ms/step - loss: 281734240.0000
Epoch 2/150
57/57 [==============================] - 1s 10ms/step - loss: 244775920.0000
Epoch 3/150
57/57 [==============================] - 1s 11ms/step - loss: 356642752.0000
Epoch 4/150
57/57 [==============================] - 1s 10ms/step - loss: 590591872.0000
Epoch 5/150
57/57 [==============================] - 1s 10ms/step - loss: 487533056.0000
Epoch 6/150
57/57 [==============================] - 1s 10ms/step - loss: 459122176.0000
Epoch 7/150
57/57 [==============================] - 1s 11ms/step - loss: 224523456.0000
Epoch 8/150
57/57 [==============================] - 1s 12ms/step - loss: 149752816.0000
Epoch 9/150
57/57 [==============================] - 1s 12ms/step - loss: 139113520.0000
Epoch 10/150
57/57 [==============================] - 1s 12ms/step - loss: 130821656.0000
Epoch 11/150
57/57 [==============================] - 1s 12ms/step - loss: 35632328.0000
Epoch 12/150
57/57 [==============================] - 1s 12ms/step - loss: 16078325.0000
Epoch 13/150
57/57 [==============================] - 1s 11ms/step - loss: 16479484.0000
Epoch 14/150
57/57 [==============================] - 1s 11ms/step - loss: 13477521.0000
Epoch 15/150
57/57 [==============================] - 1s 12ms/step - loss: 11695184.0000
Epoch 16/150
57/57 [==============================] - 1s 12ms/step - loss: 11409919.0000
Epoch 17/150
57/57 [==============================] - 1s 11ms/step - loss: 12049713.0000
Epoch 18/150
57/57 [==============================] - 1s 12ms/step - loss: 13327271.0000
Epoch 19/150
57/57 [==============================] - 1s 12ms/step - loss: 21315224.0000
Epoch 20/150
57/57 [==============================] - 1s 12ms/step - loss: 19061474.0000
Epoch 21/150
57/57 [==============================] - 1s 12ms/step - loss: 109726272.0000
Epoch 22/150
57/57 [==============================] - 1s 12ms/step - loss: 50574500.0000
Epoch 23/150
57/57 [==============================] - 1s 12ms/step - loss: 44558024.0000
Epoch 24/150
57/57 [==============================] - 1s 12ms/step - loss: 31844762.0000
Epoch 25/150
57/57 [==============================] - 1s 12ms/step - loss: 29395076.0000
Epoch 26/150
57/57 [==============================] - 1s 12ms/step - loss: 27584408.0000
Epoch 27/150
```

```
Epoch 28/150
57/57 [==============================] - 1s 12ms/step - loss: 60519560.0000
Epoch 29/150
57/57 [==============================] - 1s 13ms/step - loss: 31023124.0000
Epoch 30/150
57/57 [==============================] - 1s 14ms/step - loss: 99096880.0000
Epoch 31/150
57/57 [==============================] - 1s 17ms/step - loss: 80313672.0000
Epoch 32/150
57/57 [==============================] - 1s 14ms/step - loss: 31691872.0000
Epoch 33/150
57/57 [==============================] - 1s 14ms/step - loss: 11691262.0000
Epoch 34/150
57/57 [==============================] - 1s 15ms/step - loss: 10333938.0000
Epoch 35/150
57/57 [==============================] - 1s 17ms/step - loss: 11362641.0000
Epoch 36/150
57/57 [==============================] - 1s 18ms/step - loss: 11008741.0000
Epoch 37/150
57/57 [==============================] - 1s 21ms/step - loss: 10275741.0000
Epoch 38/150
57/57 [==============================] - 1s 19ms/step - loss: 9435084.0000
Epoch 39/150
57/57 [==============================] - 1s 17ms/step - loss: 8880603.0000
Epoch 40/150
57/57 [==============================] - 1s 18ms/step - loss: 8722080.0000
Epoch 41/150
57/57 [==============================] - 1s 19ms/step - loss: 8788672.0000
Epoch 42/150
57/57 [==============================] - 1s 15ms/step - loss: 8670982.0000
Epoch 43/150
57/57 [==============================] - 1s 15ms/step - loss: 8170998.5000
Epoch 44/150
57/57 [==============================] - 1s 16ms/step - loss: 8164780.5000
Epoch 45/150
57/57 [==============================] - 1s 17ms/step - loss: 7199422.5000
Epoch 46/150
57/57 [==============================] - 1s 18ms/step - loss: 8806760.0000
Epoch 47/150
57/57 [==============================] - 1s 21ms/step - loss: 8419513.0000
Epoch 48/150
57/57 [==============================] - 1s 17ms/step - loss: 8959826.0000
Epoch 49/150
57/57 [==============================] - 1s 18ms/step - loss: 7433759.5000
Epoch 50/150
57/57 [==============================] - 1s 15ms/step - loss: 6248579.5000
Epoch 51/150
57/57 [==============================] - 1s 16ms/step - loss: 5601395.5000
Epoch 52/150
57/57 [==============================] - 1s 13ms/step - loss: 4739147.0000
Epoch 53/150
57/57 [==============================] - 1s 12ms/step - loss: 4277506.5000
```

```
Epoch 54/150
57/57 [==============================] - 1s 12ms/step - loss: 4290389.5000
Epoch 55/150
57/57 [==============================] - 1s 12ms/step - loss: 4538426.5000
Epoch 56/150
57/57 [==============================] - 1s 12ms/step - loss: 4374251.5000
Epoch 57/150
57/57 [==============================] - 1s 12ms/step - loss: 4369737.0000
Epoch 58/150
57/57 [==============================] - 1s 12ms/step - loss: 4268196.0000
Epoch 59/150
57/57 [==============================] - 1s 12ms/step - loss: 3766785.5000
Epoch 60/150
57/57 [==============================] - 1s 12ms/step - loss: 3824224.5000
Epoch 61/150
57/57 [==============================] - 1s 12ms/step - loss: 3672877.7500
Epoch 62/150
57/57 [==============================] - 1s 12ms/step - loss: 3727877.2500
Epoch 63/150
57/57 [==============================] - 1s 11ms/step - loss: 3634162.0000
Epoch 64/150
57/57 [==============================] - 1s 12ms/step - loss: 4292644.5000
Epoch 65/150
57/57 [==============================] - 1s 11ms/step - loss: 3801562.5000
Epoch 66/150
57/57 [==============================] - 1s 11ms/step - loss: 3500722.7500
Epoch 67/150
57/57 [==============================] - 1s 11ms/step - loss: 3326307.5000
Epoch 68/150
57/57 [==============================] - 1s 12ms/step - loss: 3363223.7500
Epoch 69/150
57/57 [==============================] - 1s 11ms/step - loss: 3418229.5000
Epoch 70/150
57/57 [==============================] - 1s 12ms/step - loss: 3150019.2500
Epoch 71/150
57/57 [==============================] - 1s 12ms/step - loss: 3098448.5000
Epoch 72/150
57/57 [==============================] - 1s 11ms/step - loss: 2968943.0000
Epoch 73/150
57/57 [==============================] - 1s 12ms/step - loss: 3218390.5000
Epoch 74/150
57/57 [==============================] - 1s 12ms/step - loss: 2861950.2500
Epoch 75/150
57/57 [==============================] - 1s 11ms/step - loss: 2818465.5000
Epoch 76/150
57/57 [==============================] - 1s 12ms/step - loss: 2904118.5000
Epoch 77/150
57/57 [==============================] - 1s 13ms/step - loss: 2823362.5000
Epoch 78/150
57/57 [==============================] - 1s 13ms/step - loss: 2738741.2500
Epoch 79/150
57/57 [==============================] - 1s 12ms/step - loss: 2748842.2500
```

```
Epoch 80/150
57/57 [==============================] - 1s 12ms/step - loss: 2904489.0000
Epoch 81/150
57/57 [==============================] - 1s 12ms/step - loss: 3005080.7500
Epoch 82/150
57/57 [==============================] - 1s 12ms/step - loss: 228369184.0000
Epoch 83/150
57/57 [==============================] - 1s 12ms/step - loss: 43137180.0000
Epoch 84/150
57/57 [==============================] - 1s 12ms/step - loss: 27645078.0000
Epoch 85/150
57/57 [==============================] - 1s 12ms/step - loss: 25591468.0000
Epoch 86/150
57/57 [==============================] - 1s 12ms/step - loss: 25255886.0000
Epoch 87/150
57/57 [==============================] - 1s 12ms/step - loss: 24915636.0000
Epoch 88/150
57/57 [==============================] - 1s 12ms/step - loss: 24186474.0000
Epoch 89/150
57/57 [==============================] - 1s 12ms/step - loss: 23440858.0000
Epoch 90/150
57/57 [==============================] - 1s 12ms/step - loss: 23333088.0000
Epoch 91/150
57/57 [==============================] - 1s 11ms/step - loss: 23102664.0000
Epoch 92/150
57/57 [==============================] - 1s 12ms/step - loss: 21873714.0000
Epoch 93/150
57/57 [==============================] - 1s 12ms/step - loss: 22785840.0000
Epoch 94/150
57/57 [==============================] - 1s 13ms/step - loss: 21441862.0000
Epoch 95/150
57/57 [==============================] - 1s 13ms/step - loss: 21452258.0000
Epoch 96/150
57/57 [==============================] - 1s 12ms/step - loss: 20421556.0000
Epoch 97/150
57/57 [==============================] - 1s 13ms/step - loss: 19399964.0000
Epoch 98/150
57/57 [==============================] - 1s 12ms/step - loss: 18940758.0000
Epoch 99/150
57/57 [==============================] - 1s 11ms/step - loss: 17609680.0000
Epoch 100/150
57/57 [==============================] - 1s 12ms/step - loss: 16958424.0000
Epoch 101/150
57/57 [==============================] - 1s 12ms/step - loss: 16678050.0000
Epoch 102/150
57/57 [==============================] - 1s 11ms/step - loss: 15186306.0000
Epoch 103/150
57/57 [==============================] - 1s 11ms/step - loss: 15461470.0000
Epoch 104/150
57/57 [==============================] - 1s 11ms/step - loss: 14583754.0000
Epoch 105/150
57/57 [==============================] - 1s 11ms/step - loss: 13971434.0000
```

```
Epoch 106/150
57/57 [==============================] - 1s 11ms/step - loss: 13924656.0000
Epoch 107/150
57/57 [==============================] - 1s 12ms/step - loss: 12835134.0000
Epoch 108/150
57/57 [==============================] - 1s 12ms/step - loss: 11820585.0000
Epoch 109/150
57/57 [==============================] - 1s 11ms/step - loss: 13825164.0000
Epoch 110/150
57/57 [==============================] - 1s 11ms/step - loss: 12139418.0000
Epoch 111/150
57/57 [==============================] - 1s 11ms/step - loss: 10828062.0000
Epoch 112/150
57/57 [==============================] - 1s 11ms/step - loss: 10920853.0000
Epoch 113/150
57/57 [==============================] - 1s 11ms/step - loss: 11023298.0000
Epoch 114/150
57/57 [==============================] - 1s 12ms/step - loss: 12602467.0000
Epoch 115/150
57/57 [==============================] - 1s 12ms/step - loss: 11118615.0000
Epoch 116/150
57/57 [==============================] - 1s 12ms/step - loss: 13452498.0000
Epoch 117/150
57/57 [==============================] - 1s 11ms/step - loss: 11325595.0000
Epoch 118/150
57/57 [==============================] - 1s 11ms/step - loss: 10826127.0000
Epoch 119/150
57/57 [==============================] - 1s 11ms/step - loss: 10520372.0000
Epoch 120/150
57/57 [==============================] - 1s 12ms/step - loss: 9821913.0000
Epoch 121/150
57/57 [==============================] - 1s 12ms/step - loss: 9934604.0000
Epoch 122/150
57/57 [==============================] - 1s 11ms/step - loss: 10122568.0000
Epoch 123/150
57/57 [==============================] - 1s 11ms/step - loss: 10814144.0000
Epoch 124/150
57/57 [==============================] - 1s 13ms/step - loss: 10450345.0000
Epoch 125/150
57/57 [==============================] - 1s 13ms/step - loss: 9672498.0000
Epoch 126/150
57/57 [==============================] - 1s 13ms/step - loss: 9830407.0000
Epoch 127/150
57/57 [==============================] - 1s 13ms/step - loss: 13257461.0000
Epoch 128/150
57/57 [==============================] - 1s 12ms/step - loss: 10823718.0000
Epoch 129/150
57/57 [==============================] - 1s 12ms/step - loss: 9365658.0000
Epoch 130/150
57/57 [==============================] - 1s 12ms/step - loss: 10354946.0000
Epoch 131/150
57/57 [==============================] - 1s 12ms/step - loss: 10745900.0000
```

```
Epoch 132/150
57/57 [==============================] - 1s 12ms/step - loss: 10470199.0000
Epoch 133/150
57/57 [==============================] - 1s 11ms/step - loss: 9161721.0000
Epoch 134/150
57/57 [==============================] - 1s 12ms/step - loss: 9559226.0000
Epoch 135/150
57/57 [==============================] - 1s 12ms/step - loss: 9957368.0000
Epoch 136/150
57/57 [==============================] - 1s 13ms/step - loss: 11443612.0000
Epoch 137/150
57/57 [==============================] - 1s 13ms/step - loss: 8712895.0000
Epoch 138/150
57/57 [==============================] - 1s 11ms/step - loss: 8514532.0000
Epoch 139/150
57/57 [==============================] - 1s 12ms/step - loss: 8971298.0000
Epoch 140/150
57/57 [==============================] - 1s 13ms/step - loss: 8994197.0000
Epoch 141/150
57/57 [==============================] - 1s 13ms/step - loss: 9391470.0000
Epoch 142/150
57/57 [==============================] - 1s 13ms/step - loss: 12619939.0000
Epoch 143/150
57/57 [==============================] - 1s 13ms/step - loss: 12557395.0000
Epoch 144/150
57/57 [==============================] - 1s 13ms/step - loss: 9439162.0000
Epoch 145/150
57/57 [==============================] - 1s 13ms/step - loss: 8983986.0000
Epoch 146/150
57/57 [==============================] - 1s 13ms/step - loss: 10307691.0000
Epoch 147/150
57/57 [==============================] - 1s 12ms/step - loss: 8618048.0000
Epoch 148/150
57/57 [==============================] - 1s 11ms/step - loss: 8704341.0000
Epoch 149/150
57/57 [==============================] - 1s 12ms/step - loss: 8451666.0000
Epoch 150/150
57/57 [==============================] - 1s 12ms/step - loss: 9396456.0000
```

Out[28]:

```
<keras.callbacks.History at 0x1de7b61b160>
```

In [29]:

```
plt.figure(figsize=(16,8))
plt.plot(model.predict(X),color='red',label='Predicted')
plt.plot(data['Close'].values,color='green',label='Actual')
```

Out[29]:

```
[<matplotlib.lines.Line2D at 0x1de7feb41f0>]
```

In [30]:

```
model.predict(X)
```

```
57/57 [==============================] - 0s 4ms/step
```

Out[30]:

```
array([[ 5485.7954],
       [ 5347.0317],
       [ 5200.5825],
       ...,
       [30114.852 ],
       [30472.59  ],
       [30969.932 ]], dtype=float32)
```

In [31]:

```
data['Close']
```

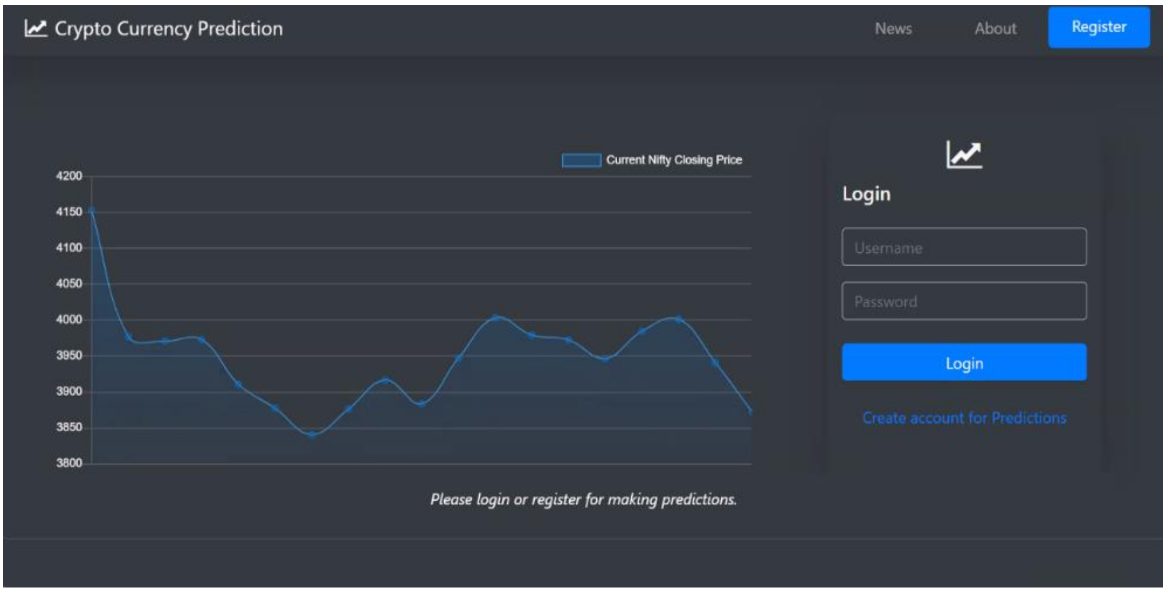
```
Out[31]: Date
         2018-10-28     6486.390137
         2018-10-29     6332.629883
         2018-10-30     6334.270020
         2018-10-31     6317.609863
         2018-11-01     6377.779785
                           ...
         2023-10-24    33901.527344
         2023-10-25    34502.820313
         2023-10-26    34156.648438
         2023-10-27    33909.800781
         2023-10-28    34095.496094
         Name: Close, Length: 1827, dtype: float64
```

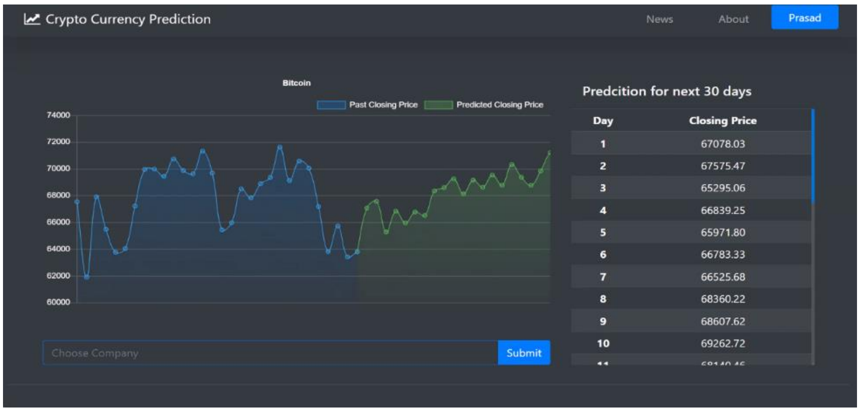
```
In [32]: yhat2=model.predict(X)
         yhat2.shape

         57/57 [==============================] - 0s 4ms/step
Out[32]: (1797, 1)
```

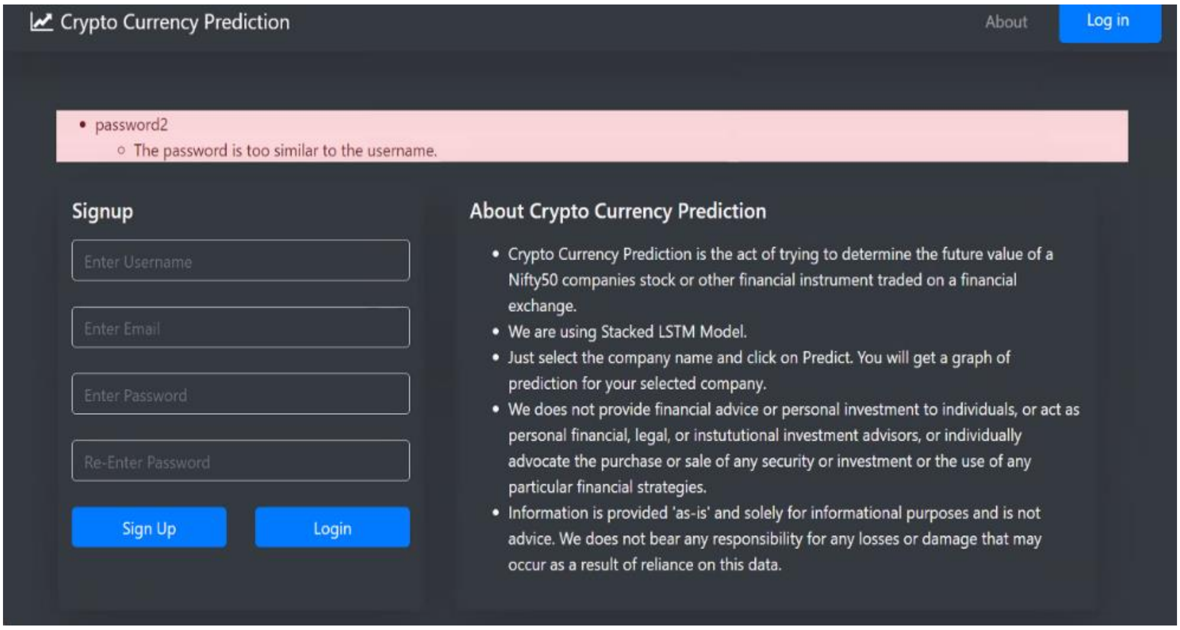
```
In [33]: X.shape

Out[33]: (1797, 30, 1)
```

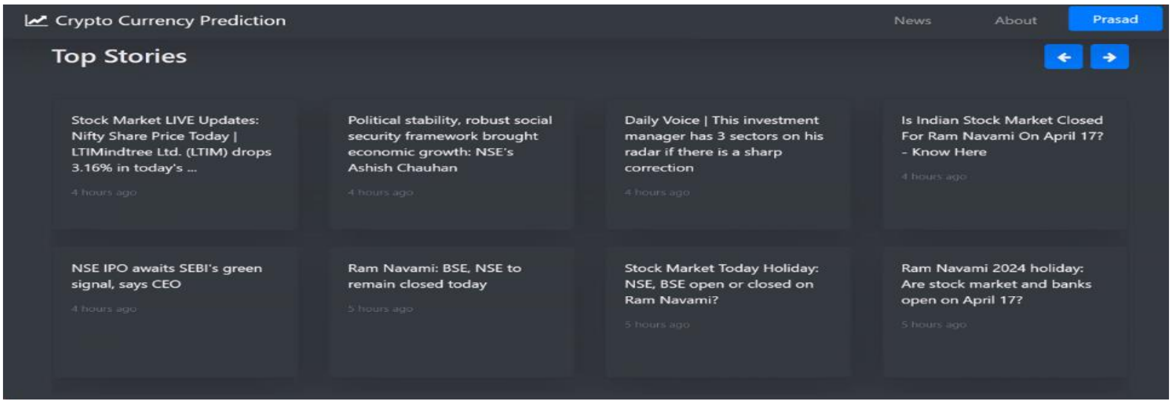
```
In [34]: y.shape

Out[34]: (1797,)
```

```
In [35]: from sklearn.metrics import r2_score
         print("R2 aquare",r2_score(y, yhat2))

         from sklearn.metrics import mean_absolute_error

         print("Mean Absolute Error", mean_absolute_error(y, yhat2))

         from sklearn.metrics import mean_squared_error
         mean_squared_error= mean_squared_error(y, yhat2)
         print("Mean Squared Erorr", mean_squared_error)

         from math import sqrt
         rootMeanSquaredError = sqrt(mean_squared_error)
         print("Root Mean Squared Error",rootMeanSquaredError)

         R2 aquare 0.966139401623341
         Mean Absolute Error 1886.7023871354952
         Mean Squared Erorr 8854569.96775436
         Root Mean Squared Error 2975.6629459255564
```




Crypto Currency Prediction
News
About
Prasad
Bitcoin
Past Closing Price
Predicted Closing Price
74000
72000
70000
68000
66000
64000
62000
60000
Choose Company
Submit
Predcition for next 30 days
Day
Closing Price
1 67078.03
2 67575.47
3 65295.06
4 66839.25
5 65971.80
6 66783.33
7 66525.68
8 68360.22
9 68607.62
10 69262.72


Crypto Currency Prediction
About
Log in
password2
The password is too similar to the username.
Signup
Enter Username
Enter Email
Enter Password
Re-Enter Password
Sign Up
Login
About Crypto Currency Prediction
Crypto Currency Prediction is the act of trying to determine the future value of a Nifty50 companies stock or other financial instrument traded on a financial exchange.
We are using Stacked LSTM Model.
Just select the company name and click on Predict. You will get a graph of prediction for your selected company.
We does not provide financial advice or personal investment to individuals, or act as personal financial, legal, or instututional investment advisors, or individually advocate the purchase or sale of any security or investment or the use of any particular financial strategies.
Information is provided 'as-is' and solely for informational purposes and is not advice. We does not bear any responsibility for any losses or damage that may occur as a result of reliance on this data.


Crypto Currency Prediction
News
About
Prasad
Top Stories
Stock Market LIVE Updates: Nifty Share Price Today | LTIMindtree Ltd. (LTIM) drops 3.16% in today's ...
4 hours ago
Political stability, robust social security framework brought economic growth: NSE's Ashish Chauhan
4 hours ago
Daily Voice | This investment manager has 3 sectors on his radar if there is a sharp correction
4 hours ago
Is Indian Stock Market Closed For Ram Navami On April 17? – Know Here
4 hours ago
NSE IPO awaits SEBI's green signal, says CEO
4 hours ago
Ram Navami: BSE, NSE to remain closed today
5 hours ago
Stock Market Today Holiday: NSE, BSE open or closed on Ram Navami?
5 hours ago
Ram Navami 2024 holiday: Are stock market and banks open on April 17?
5 hours ago

Analysis of Results:

- Evaluation and analysis of the results obtained from the predictive model.
- Discussion on the accuracy and performance of the model in predicting stock prices in the crypto market.
- Identification of limitations and challenges encountered during the analysis process.
- Exploration of potential improvements and future directions for enhancing the predictive model's effectiveness and applicability.