Stock Prediction

By Swagata Naskar

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from datetime import date
import math
import pandas_datareader as web
```

In [2]:
```
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
```

In [3]:
```
data = web.get_data_yahoo('^NSEBANK', start = datetime.datetime(2010, 1, 2),
                             end = date.today())
data = data[['Adj Close']]
data.columns = ['Price']
data.head()
```

Out[3]:

Date	Price
2010-01-04	9112.349609
2010-01-05	9192.150391
2010-01-06	9223.000000
2010-01-07	9192.950195
2010-01-08	9160.700195

In [4]:
```
data.tail()
```

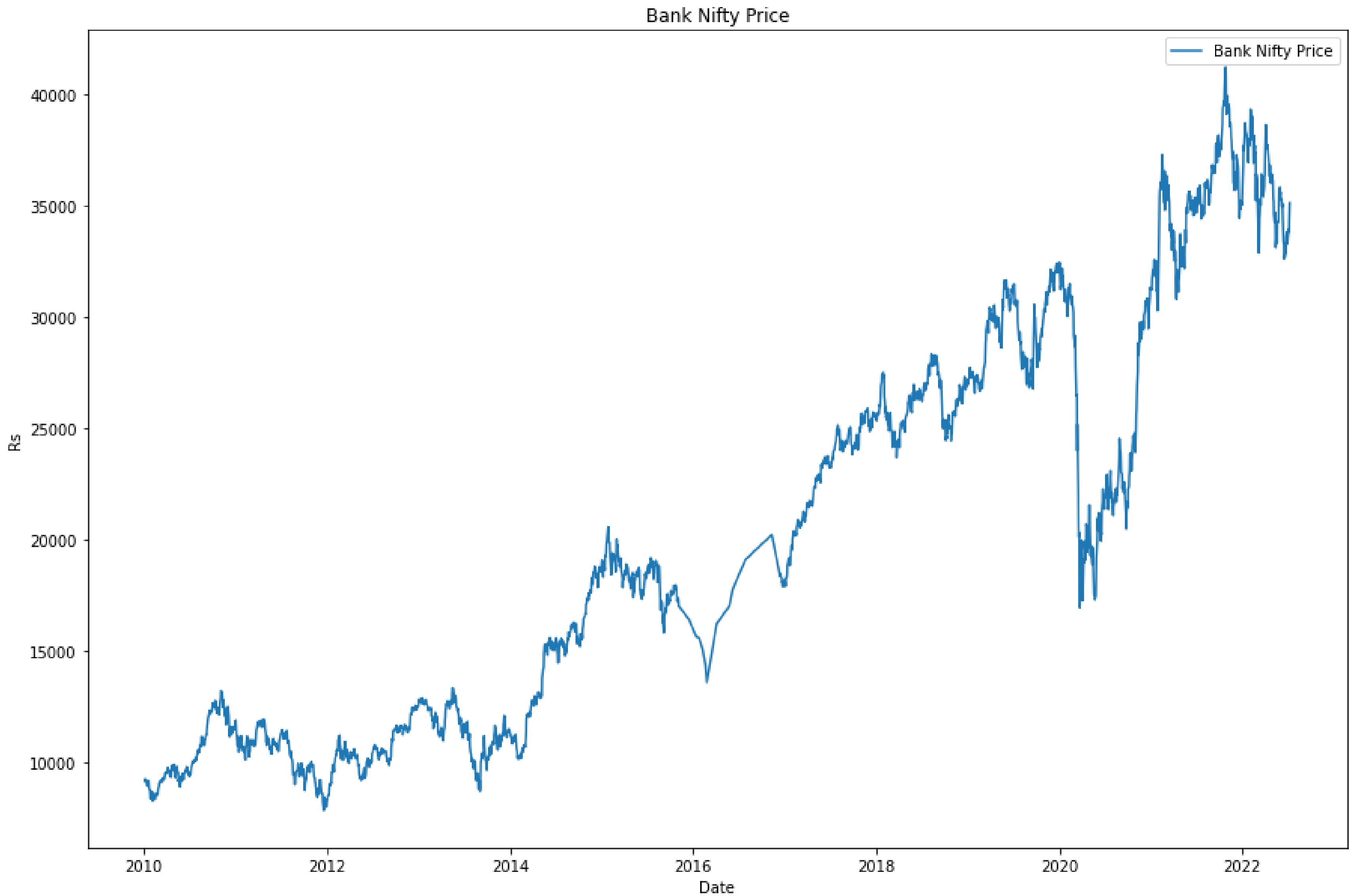
Out[4]:

Date	Price
2022-07-04	33940.898438
2022-07-05	33815.898438
2022-07-06	34324.250000
2022-07-07	34920.300781
2022-07-08	35124.050781

In [5]:
```
print('There are {} number of days in the dataset.'.format(data.shape[0]))
```

```
There are 2800 number of days in the dataset.
```

In [6]:
```
plt.figure(figsize=(15, 10))#, dpi=100)
plt.plot(data.index, data['Price'], label='Bank Nifty Price')
plt.xlabel('Date')
plt.ylabel('Rs')
plt.title('Bank Nifty Price')
plt.legend()
plt.show()
```



In [7]:
```
def get_technical_indicators(dataset):
    # Create 7 and 21 days Moving Average
    dataset['ma7'] = dataset['Price'].rolling(window=7).mean()
    dataset['ma21'] = dataset['Price'].rolling(window=21).mean()

    # Create MACD
    dataset['26ema'] = dataset['Price'].ewm(span=26).mean()
    dataset['12ema'] = dataset['Price'].ewm(span=12).mean()
```

```
    dataset['MACD'] = dataset['12ema']-dataset['26ema']

    # Create Bollinger Bands
    dataset['20sd'] = dataset['Price'].rolling(window = 21).std()
    dataset['upper_band'] = dataset['ma21'] + (dataset['20sd']*2)
    dataset['lower_band'] = dataset['ma21'] - (dataset['20sd']*2)

    # Create Exponential moving average
    dataset['ema'] = dataset['Price'].ewm(com=0.5).mean()

    # Create Momentum
    dataset['momentum'] = dataset['Price']-1
    dataset['log_momentum'] = np.log(dataset['momentum'])
    return dataset
```

In [8]:
```
df = get_technical_indicators(data)
```

In [9]:
```
df = df.dropna()
df.head()
```

Out[9]:

Date	Price	ma7	ma21	26ema	12ema	MACD	20sd	upper_band	lower_band	ema	mo
2010-02-02	8468.150391	8570.264230	8907.354678	8801.227916	8697.663730	-103.564187	277.087495	9461.529668	8353.179688	8520.889180	846
2010-02-03	8631.849609	8550.149833	8884.473726	8785.853374	8687.275187	-98.578187	279.144713	9442.763152	8326.184299	8594.862800	863
2010-02-04	8471.099609	8517.935547	8850.137974	8757.752225	8653.288548	-104.463677	283.715600	9417.569174	8282.706775	8512.354006	847
2010-02-05	8223.250000	8502.714146	8802.530831	8710.746733	8585.906040	-124.840693	301.351790	9405.234412	8199.827251	8319.618002	822
2010-02-08	8342.200195	8491.049944	8762.018927	8678.779150	8547.828169	-130.950980	303.419639	9368.858204	8155.179649	8334.672798	834

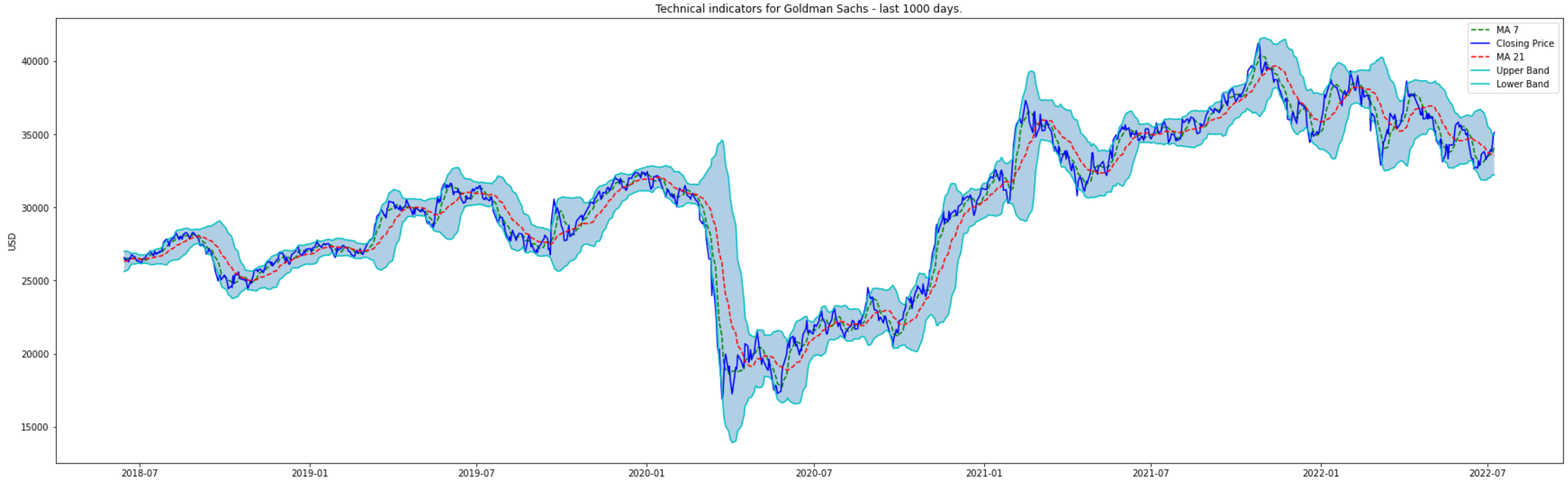
In [10]:
```
def plot_technical_indicators(dataset, last_days):
    plt.figure(figsize=(16, 10), dpi=100)
    shape_0 = dataset.shape[0]
    xmacd_ = shape_0-last_days
```

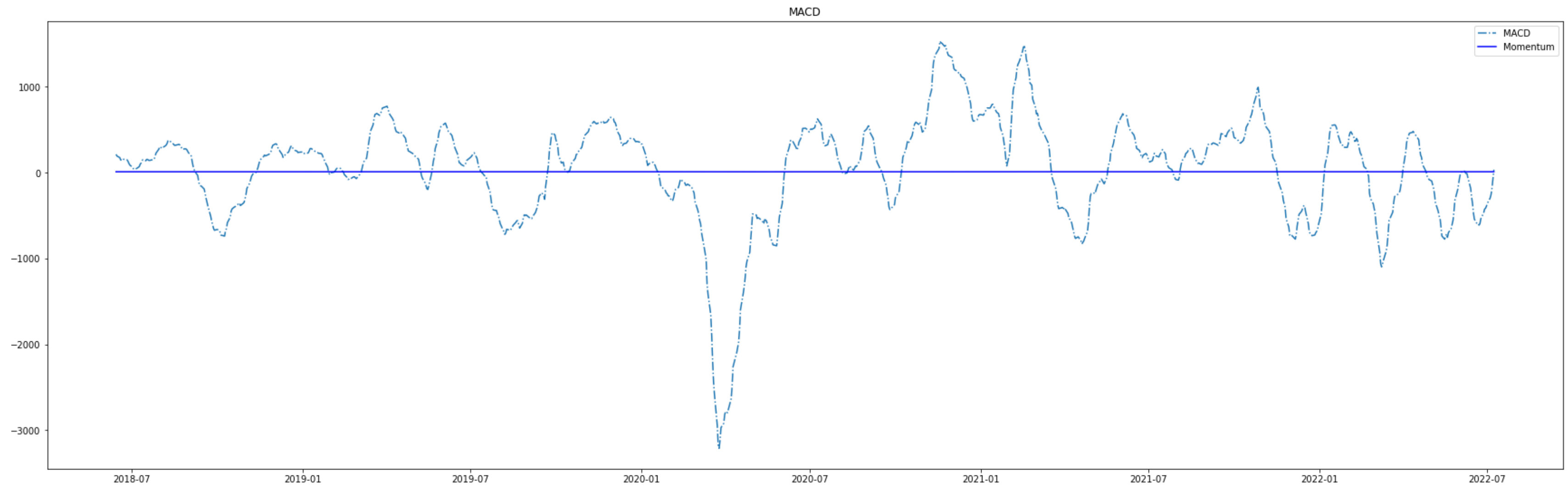
```
    dataset = dataset.iloc[-last_days:, :]
    x_ = range(3, dataset.shape[0])
    x_ =list(dataset.index)

    plt.figure(figsize=(30,20))
    # Plot first subplot
    plt.subplot(2, 1, 1)
    plt.plot(dataset['ma7'],label='MA 7', color='g',linestyle='--')
    plt.plot(dataset['Price'],label='Closing Price', color='b')
    plt.plot(dataset['ma21'],label='MA 21', color='r',linestyle='--')
    plt.plot(dataset['upper_band'],label='Upper Band', color='c')
    plt.plot(dataset['lower_band'],label='Lower Band', color='c')
    plt.fill_between(x_, dataset['lower_band'], dataset['upper_band'], alpha=0.35)
    plt.title('Technical indicators for Goldman Sachs - last {} days.'.format(last_days))
    plt.ylabel('USD')
    plt.legend()

    # Plot second subplot

    plt.subplot(2, 1, 2)
    plt.title('MACD')
    plt.plot(dataset['MACD'],label='MACD', linestyle='-.')
#     plt.hlines(15, xmacd_, shape_0, colors='g', linestyles='--')
#     plt.hlines(-15, xmacd_, shape_0, colors='g', linestyles='--')
    plt.plot(dataset['log_momentum'],label='Momentum', color='b',linestyle='-')

    plt.legend()
    plt.show()
```

In [11]:

```
plot_technical_indicators(df, 1000)
```

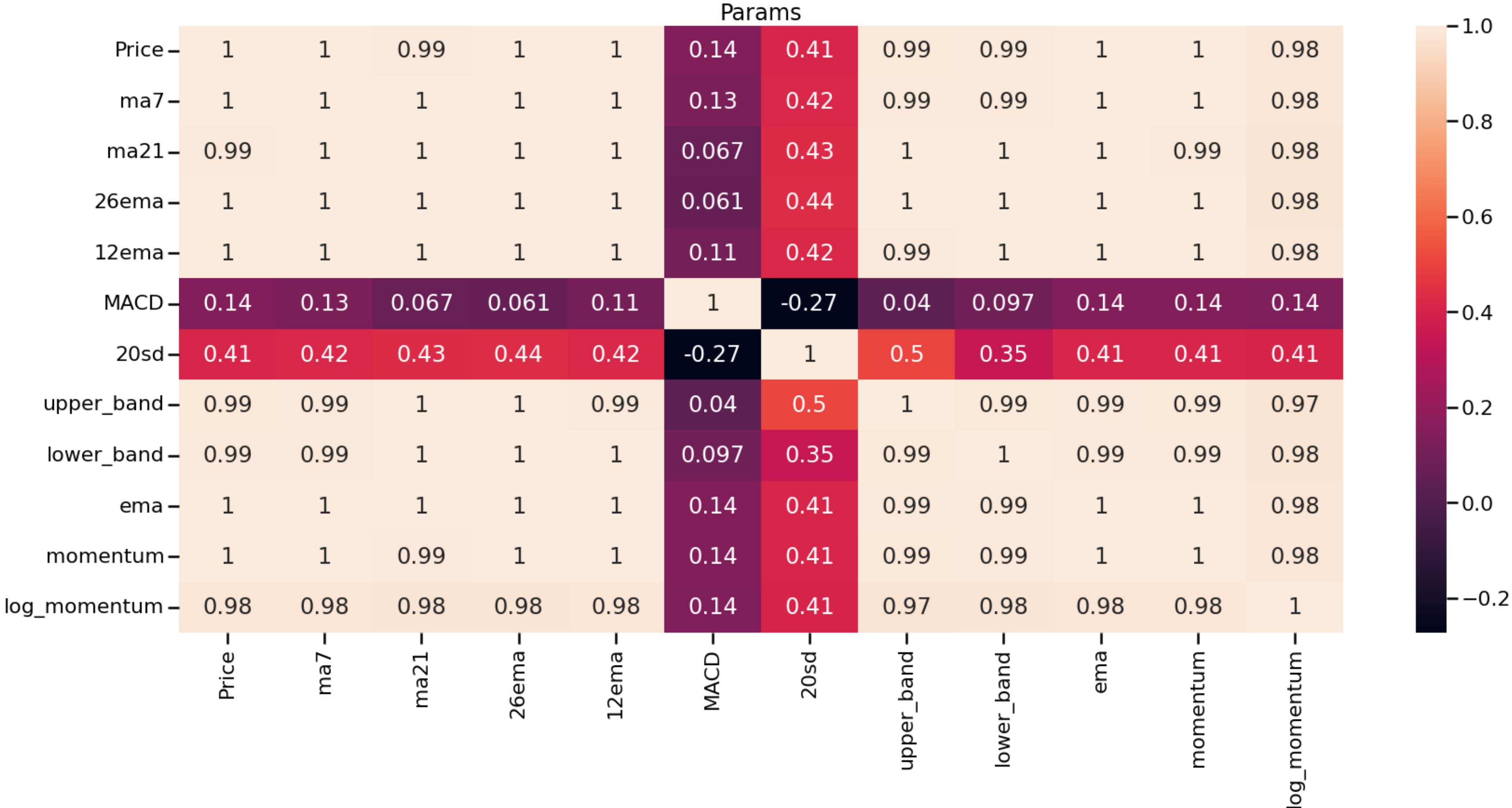
```
<Figure size 1600x1000 with 0 Axes>
```





In [12]:
```
plt.figure(figsize = (28,12))
sns.set_context('poster',font_scale=1)
sns.heatmap(df.corr(), annot = True).set_title('Params')
```

Out[12]:
```
Text(0.5, 1.0, 'Params')
```



In [13]:
```
print('Total dataset has {} samples, and {} features.'.format(df.shape[0], \
                                                              df.shape[1]))
```

```
Total dataset has 2780 samples, and 12 features.
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['Price', 'ma7', 'ma21', '26ema', '12ema', 'MACD', '20sd', 'upper_band',
       'lower_band', 'ema', 'momentum', 'log_momentum'],
      dtype='object')
```

In [15]:
```
df
```

Out[15]:

Date	Price	ma7	ma21	26ema	12ema	MACD	20sd	upper_band	lower_band	e
2010-02-02	8468.150391	8570.264230	8907.354678	8801.227916	8697.663730	-103.564187	277.087495	9461.529668	8353.179688	8520.889
2010-02-03	8631.849609	8550.149833	8884.473726	8785.853374	8687.275187	-98.578187	279.144713	9442.763152	8326.184299	8594.862
2010-02-04	8471.099609	8517.935547	8850.137974	8757.752225	8653.288548	-104.463677	283.715600	9417.569174	8282.706775	8512.354
2010-02-05	8223.250000	8502.714146	8802.530831	8710.746733	8585.906040	-124.840693	301.351790	9405.234412	8199.827251	8319.618
2010-02-08	8342.200195	8491.049944	8762.018927	8678.779150	8547.828169	-130.950980	303.419639	9368.858204	8155.179649	8334.672
...	...	...	...	...	...	...	...	...	...	
2022-07-04	33940.898438	33608.070871	33683.395182	33860.420558	33564.752226	-295.668331	821.997085	35327.389353	32039.401012	33793.410
2022-07-05	33815.898438	33634.992188	33612.238002	33857.122623	33603.390105	-253.732518	734.108973	35080.455948	32144.020057	33808.402
2022-07-06	34324.250000	33708.277902	33580.249907	33891.724651	33714.291627	-177.433024	683.702987	34947.655881	32212.843933	34152.300
2022-07-07	34920.300781	33890.828125	33579.019066	33967.915475	33899.831497	-68.083978	681.139532	34941.298129	32216.740003	34664.300
2022-07-08	35124.050781	34155.707031	33580.857236	34053.555127	34088.172925	34.617798	685.446623	34951.750482	32209.963989	34970.800

2780 rows × 12 columns

In [16]:

```
data_training = df[df.index < '2019-01-31'].copy()
data_training
```

Out[16]:

Date	Price	ma7	ma21	26ema	12ema	MACD	20sd	upper_band	lower_band	e
2010-02-02	8468.150391	8570.264230	8907.354678	8801.227916	8697.663730	-103.564187	277.087495	9461.529668	8353.179688	8520.889
2010-02-03	8631.849609	8550.149833	8884.473726	8785.853374	8687.275187	-98.578187	279.144713	9442.763152	8326.184299	8594.862
2010-02-04	8471.099609	8517.935547	8850.137974	8757.752225	8653.288548	-104.463677	283.715600	9417.569174	8282.706775	8512.354
2010-02-05	8223.250000	8502.714146	8802.530831	8710.746733	8585.906040	-124.840693	301.351790	9405.234412	8199.827251	8319.618
2010-02-08	8342.200195	8491.049944	8762.018927	8678.779150	8547.828169	-130.950980	303.419639	9368.858204	8155.179649	8334.672
...	...	...	...	...	...	...	...	...	...	
2019-01-24	27266.400391	27428.878348	27316.590495	27208.914029	27369.590681	160.676652	220.658590	27757.907674	26875.273315	27288.044
2019-01-25	27115.300781	27376.250000	27322.709542	27201.979714	27330.469158	128.489444	212.693942	27748.097427	26897.321658	27172.881
2019-01-28	26653.050781	27251.150112	27311.971447	27161.318312	27226.250946	64.932634	240.158143	27792.287733	26831.655162	26826.327
2019-01-29	26573.400391	27124.964565	27285.692894	27117.768836	27125.812399	8.043563	287.196724	27860.086343	26711.299446	26657.709
2019-01-30	26825.500000	27023.807478	27269.754836	27096.119293	27079.610492	-16.508801	303.342509	27876.439854	26663.069819	26769.569

1935 rows × 12 columns

In [17]:
```
data_testing = df[df.index >= '2019-01-31'].copy()
data_testing
```

Out[17]:

Date	Price	ma7	ma21	26ema	12ema	MACD	20sd	upper_band	lower_band	e
2019-01-31	27295.449219	26997.121652	27275.504836	27110.884472	27112.816450	1.931977	302.594107	27880.693050	26670.316623	27120.156
2019-02-01	27085.949219	26973.578683	27281.509580	27109.037417	27108.683029	-0.354387	297.220439	27875.950457	26687.068702	27097.351
2019-02-04	27186.599609	26962.178571	27281.109561	27114.782764	27120.670196	5.887431	297.348316	27875.806193	26686.412929	27156.850
2019-02-05	27271.699219	26984.521205	27279.545201	27126.406205	27143.905430	17.499225	297.305238	27874.155677	26684.934725	27233.416
2019-02-06	27402.349609	27091.563895	27274.442801	27146.846457	27183.666073	36.819615	294.063180	27862.569162	26686.316441	27346.038
...	...	...	...	...	...	...	...	...	...	
2022-07-04	33940.898438	33608.070871	33683.395182	33860.420558	33564.752226	-295.668331	821.997085	35327.389353	32039.401012	33793.410
2022-07-05	33815.898438	33634.992188	33612.238002	33857.122623	33603.390105	-253.732518	734.108973	35080.455948	32144.020057	33808.402
2022-07-06	34324.250000	33708.277902	33580.249907	33891.724651	33714.291627	-177.433024	683.702987	34947.655881	32212.843933	34152.300
2022-07-07	34920.300781	33890.828125	33579.019066	33967.915475	33899.831497	-68.083978	681.139532	34941.298129	32216.740003	34664.300
2022-07-08	35124.050781	34155.707031	33580.857236	34053.555127	34088.172925	34.617798	685.446623	34951.750482	32209.963989	34970.800

845 rows × 12 columns

In [18]:

```
scalar = MinMaxScaler()

data_training_scaled = scalar.fit_transform(data_training)
print(data_training_scaled.shape)
data_training_scaled
```

```
(1935, 12)
```

Out[18]:
```
array([[0.0326293 , 0.02151149, 0.02926575, ..., 0.03154458, 0.0326293 ,
        0.06387762],
       [0.04060628, 0.02050703, 0.02810332, ..., 0.03517799, 0.04060628,
        0.07872494],
       [0.03277302, 0.01889834, 0.02635895, ..., 0.03112535, 0.03277302,
        0.06414764],
       ...,
       [0.9187704 , 0.95438137, 0.96428198, ..., 0.93066481, 0.9187704 ,
        0.95296134],
       [0.91488907, 0.94808003, 0.96294694, ..., 0.92238268, 0.91488907,
        0.95064069],
       [0.92717376, 0.94302853, 0.96213723, ..., 0.92787699, 0.92717376,
        0.95796207]])
```

In [19]:
```
X_train = []
y_train = []
```

In [20]:
```
for i in range(60, data_training.shape[0]):
    X_train.append(data_training_scaled[i-60: i])
    y_train.append(data_training_scaled[i, 0])
```

In [21]:
```
X_train, y_train = np.array(X_train), np.array(y_train)
X_train.shape, y_train.shape
```

Out[21]:
```
((1875, 60, 12), (1875,))
```

In [22]:
```
regressor = Sequential()

regressor.add(LSTM(units = 50, activation = 'relu', return_sequences = True, input_shape = (X_train.shape[1], 12)))
regressor.add(Dropout(0.2))

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

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

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

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

In [23]:
```
regressor.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm (LSTM)                 (None, 60, 50)            12600     
                                                                 
 dropout (Dropout)           (None, 60, 50)            0         
                                                                 
 lstm_1 (LSTM)               (None, 60, 60)            26640     
                                                                 
 dropout_1 (Dropout)         (None, 60, 60)            0         
                                                                 
 lstm_2 (LSTM)               (None, 60, 80)            45120     
                                                                 
 dropout_2 (Dropout)         (None, 60, 80)            0         
                                                                 
 lstm_3 (LSTM)               (None, 120)               96480     
                                                                 
 dropout_3 (Dropout)         (None, 120)               0         
                                                                 
 dense (Dense)               (None, 1)                 121       
                                                                 
=================================================================
Total params: 180,961
Trainable params: 180,961
Non-trainable params: 0
_________________________________________________________________
```

In [24]:
```
# Compiling the RNN
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

In [25]:
```
regressor.fit(X_train, y_train, epochs=50, batch_size = 64)
```

```
Epoch 1/50
30/30 [==============================] - 9s 175ms/step - loss: 0.0867
Epoch 2/50
30/30 [==============================] - 5s 173ms/step - loss: 0.0151
Epoch 3/50
30/30 [==============================] - 5s 173ms/step - loss: 0.0107
Epoch 4/50
30/30 [==============================] - 5s 177ms/step - loss: 0.0087
Epoch 5/50
30/30 [==============================] - 5s 174ms/step - loss: 0.0091
Epoch 6/50
30/30 [==============================] - 5s 174ms/step - loss: 0.0087
Epoch 7/50
30/30 [==============================] - 5s 174ms/step - loss: 0.0073
Epoch 8/50
30/30 [==============================] - 5s 175ms/step - loss: 0.0072
Epoch 9/50
30/30 [==============================] - 5s 174ms/step - loss: 0.0065
Epoch 10/50
30/30 [==============================] - 5s 175ms/step - loss: 0.0061
Epoch 11/50
30/30 [==============================] - 5s 175ms/step - loss: 0.0061
Epoch 12/50
30/30 [==============================] - 5s 175ms/step - loss: 0.0058
Epoch 13/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0059
Epoch 14/50
30/30 [==============================] - 5s 178ms/step - loss: 0.0057
Epoch 15/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0057
Epoch 16/50
30/30 [==============================] - 5s 180ms/step - loss: 0.0059
Epoch 17/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0053
Epoch 18/50
30/30 [==============================] - 5s 177ms/step - loss: 0.0054
Epoch 19/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0054
Epoch 20/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0050
Epoch 21/50
30/30 [==============================] - 5s 178ms/step - loss: 0.0049
Epoch 22/50
30/30 [==============================] - 5s 178ms/step - loss: 0.0057
Epoch 23/50
```

```
30/30 [==============================] - 5s 180ms/step - loss: 0.0048
Epoch 24/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0046
Epoch 25/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0047
Epoch 26/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0046
Epoch 27/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0046
Epoch 28/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0046
Epoch 29/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0041
Epoch 30/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0045
Epoch 31/50
30/30 [==============================] - 5s 182ms/step - loss: 0.0041
Epoch 32/50
30/30 [==============================] - 5s 179ms/step - loss: 0.0045
Epoch 33/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0044
Epoch 34/50
30/30 [==============================] - 6s 184ms/step - loss: 0.0040
Epoch 35/50
30/30 [==============================] - 5s 182ms/step - loss: 0.0041
Epoch 36/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0041
Epoch 37/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0036
Epoch 38/50
30/30 [==============================] - 5s 181ms/step - loss: 0.0039
Epoch 39/50
30/30 [==============================] - 6s 201ms/step - loss: 0.0038
Epoch 40/50
30/30 [==============================] - 6s 204ms/step - loss: 0.0041
Epoch 41/50
30/30 [==============================] - 6s 200ms/step - loss: 0.0039
Epoch 42/50
30/30 [==============================] - 6s 191ms/step - loss: 0.0037
Epoch 43/50
30/30 [==============================] - 6s 192ms/step - loss: 0.0039
Epoch 44/50
30/30 [==============================] - 6s 192ms/step - loss: 0.0040
Epoch 45/50
30/30 [==============================] - 6s 191ms/step - loss: 0.0040
```

```
Epoch 46/50
30/30 [==============================] - 6s 200ms/step - loss: 0.0037
Epoch 47/50
30/30 [==============================] - 6s 192ms/step - loss: 0.0034
Epoch 48/50
30/30 [==============================] - 6s 201ms/step - loss: 0.0032
Epoch 49/50
30/30 [==============================] - 6s 193ms/step - loss: 0.0032
Epoch 50/50
30/30 [==============================] - 6s 191ms/step - loss: 0.0035
```

Out[25]:
```
<keras.callbacks.History at 0x2ceba6ec970>
```

In [26]:
```python
past_100 = data_training.tail(100)

dt = past_100.append(data_testing, ignore_index = True)
dt
```

```
C:\Users\US\AppData\Local\Temp\ipykernel_1828\1412723301.py:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  dt = past_100.append(data_testing, ignore_index = True)
```

Out[26]:

	Price	ma7	ma21	26ema	12ema	MACD	20sd	upper_band	lower_band	em
0	27819.500000	28082.449777	28011.473772	27843.299406	28038.335626	195.036220	239.913058	28491.299888	27531.647656	27909.40813
1	27430.750000	28024.742746	28015.035714	27812.740190	27944.860914	132.120724	230.046379	28475.128473	27554.942956	27590.30271
2	27376.050781	27897.864397	27999.823847	27780.392827	27857.351663	76.958836	260.745660	28521.315167	27478.332527	27447.46809
3	27468.699219	27783.442801	27979.357143	27757.304411	27797.558979	40.254568	284.850507	28549.058157	27409.656129	27461.62217
4	27481.449219	27677.349888	27960.573754	27736.870693	27748.926708	12.056015	304.351450	28569.276653	27351.870854	27474.84020
...	...	...	...	...	...	...	...	...	...	...
940	33940.898438	33608.070871	33683.395182	33860.420558	33564.752226	-295.668331	821.997085	35327.389353	32039.401012	33793.41065
941	33815.898438	33634.992188	33612.238002	33857.122623	33603.390105	-253.732518	734.108973	35080.455948	32144.020057	33808.40251
942	34324.250000	33708.277902	33580.249907	33891.724651	33714.291627	-177.433024	683.702987	34947.655881	32212.843933	34152.30083
943	34920.300781	33890.828125	33579.019066	33967.915475	33899.831497	-68.083978	681.139532	34941.298129	32216.740003	34664.30080
944	35124.050781	34155.707031	33580.857236	34053.555127	34088.172925	34.617798	685.446623	34951.750482	32209.963989	34970.80078

945 rows × 12 columns

In [27]:
```
inputs = scalar.fit_transform(dt)
print(inputs.shape)
inputs
```

```
(945, 12)
```

Out[27]:
```
array([[0.44825485, 0.45705141, 0.44013283, ..., 0.44260293, 0.44825485,
        0.55822432],
       [0.4322705 , 0.45450529, 0.44030402, ..., 0.42912793, 0.4322705 ,
        0.54243042],
       [0.43002141, 0.44890724, 0.43957292, ..., 0.42309639, 0.43002141,
        0.54019018],
       ...,
       [0.71571275, 0.70527101, 0.70777372, ..., 0.70622413, 0.71571275,
        0.79403928],
       [0.74022076, 0.71332539, 0.70771457, ..., 0.72784457, 0.74022076,
        0.81336125],
       [0.74859842, 0.72501223, 0.70780291, ..., 0.74078727, 0.74859842,
        0.81989058]])
```

In [28]:
```
X_test = []
y_test = []

for i in range(60, inputs.shape[0]):
    X_test.append(inputs[i-60:i])
    y_test.append(inputs[i, 0])

X_test, y_test = np.array(X_test), np.array(y_test)
X_test.shape, y_test.shape
```

Out[28]:
```
((885, 60, 12), (885,))
```

In [29]:
```
y_pred = regressor.predict(X_test)
```

```
28/28 [==============================] - 2s 47ms/step
```

In [30]:
```
y_pred
```

Out[30]:
```
array([[0.30307862],
       [0.30494213],
       [0.30680764],
       [0.3086143 ],
       [0.3103286 ],
       [0.3118968 ],
       [0.31327605],
       [0.3144725 ],
       [0.31552482],
       [0.316484  ],
       [0.3174126 ],
       [0.3183757 ],
       [0.31943733],
       [0.3206396 ],
       [0.32197225],
       [0.32338607],
       [0.32483906],
       [0.32628745],
       [0.32771033],
       [0.329109  ],
       [0.33047718],
       [0.33179712],
       [0.3330652 ],
       [0.33429056],
       [0.3355012 ],
       [0.33673877],
       [0.33802706],
       [0.33936518],
       [0.34072715],
       [0.34208935],
       [0.34343755],
       [0.34476522],
       [0.34606367],
       [0.34732747],
       [0.34855014],
       [0.3497079 ],
       [0.35077643],
       [0.3517267 ],
       [0.35250318],
       [0.35305142],
       [0.35335046],
       [0.35343564],
       [0.35334975],
       [0.3531446 ],
       [0.35287714],
```

```
       [0.3526076 ],
       [0.3523866 ],
       [0.35224417],
       [0.35219002],
       [0.35220507],
       [0.3522581 ],
       [0.3523034 ],
       [0.35228878],
       [0.35217166],
       [0.3519436 ],
       [0.35162097],
       [0.35121864],
       [0.35077333],
       [0.35031152],
       [0.34984356],
       [0.34937257],
       [0.3489167 ],
       [0.348531  ],
       [0.34827974],
       [0.34822434],
       [0.34840697],
       [0.34886426],
       [0.3496495 ],
       [0.35083348],
       [0.35247338],
       [0.35462242],
       [0.35733885],
       [0.36065453],
       [0.36455208],
       [0.3689381 ],
       [0.3736418 ],
       [0.37851542],
       [0.38344204],
       [0.38836044],
       [0.39322513],
       [0.39799905],
       [0.40262812],
       [0.40704405],
       [0.4111991 ],
       [0.41504395],
       [0.41856503],
       [0.42173105],
       [0.42451394],
       [0.42691946],
       [0.42899674],
```

```
[0.43085104],
[0.43257302],
[0.43418917],
[0.435682  ],
[0.43705586],
[0.43829966],
[0.439435  ],
[0.44047254],
[0.44141603],
[0.44228554],
[0.44308   ],
[0.4437679 ],
[0.44429427],
[0.44459933],
[0.4446482 ],
[0.44440222],
[0.44385314],
[0.4430011 ],
[0.441876  ],
[0.44056004],
[0.4392364 ],
[0.43806773],
[0.43720073],
[0.43673468],
[0.43679112],
[0.43749115],
[0.4389032 ],
[0.4409852 ],
[0.44365108],
[0.44677597],
[0.45025033],
[0.45396346],
[0.4577511 ],
[0.46146452],
[0.46498126],
[0.46822834],
[0.47114998],
[0.4737156 ],
[0.47589403],
[0.4776547 ],
[0.47899133],
[0.47991472],
[0.48047328],
[0.4807263 ],
[0.48074108],
```

```
       [0.48060662],
       [0.48043668],
       [0.48034483],
       [0.48041016],
       [0.48070395],
       [0.48126364],
       [0.48210543],
       [0.48323166],
       [0.48462826],
       [0.48619217],
       [0.4877934 ],
       [0.48930526],
       [0.4906404 ],
       [0.49173552],
       [0.49254215],
       [0.49304533],
       [0.49326986],
       [0.49323875],
       [0.49293238],
       [0.49229336],
       [0.49126017],
       [0.48978126],
       [0.4878348 ],
       [0.48544818],
       [0.4826802 ],
       [0.47957402],
       [0.47618765],
       [0.4725532 ],
       [0.46868956],
       [0.4645871 ],
       [0.460263  ],
       [0.45573485],
       [0.45105654],
       [0.44631582],
       [0.44156408],
       [0.43685985],
       [0.43226925],
       [0.42784745],
       [0.42362583],
       [0.41960752],
       [0.41575024],
       [0.41200095],
       [0.40837613],
       [0.40492707],
       [0.40168765],
```

```
       [0.39864984],
       [0.39580017],
       [0.3930901 ],
       [0.390486  ],
       [0.38795608],
       [0.38549536],
       [0.38312557],
       [0.3808934 ],
       [0.3788514 ],
       [0.3770597 ],
       [0.37555426],
       [0.37430614],
       [0.37326062],
       [0.37233144],
       [0.37156478],
       [0.37116274],
       [0.37133437],
       [0.37222254],
       [0.37391016],
       [0.37638962],
       [0.37951183],
       [0.3830438 ],
       [0.38671362],
       [0.390221  ],
       [0.3933132 ],
       [0.39587224],
       [0.3978093 ],
       [0.39909434],
       [0.39976108],
       [0.3999098 ],
       [0.3996673 ],
       [0.3991953 ],
       [0.39865568],
       [0.3982036 ],
       [0.39796174],
       [0.39799234],
       [0.3983307 ],
       [0.3990202 ],
       [0.4001057 ],
       [0.40161377],
       [0.4035621 ],
       [0.4059416 ],
       [0.40870735],
       [0.41181397],
       [0.41520366],
```

```
       [0.4188152 ],
       [0.4226113 ],
       [0.4265145 ],
       [0.43045413],
       [0.43438643],
       [0.43827927],
       [0.442128  ],
       [0.44593936],
       [0.44971752],
       [0.4534403 ],
       [0.4571085 ],
       [0.4607393 ],
       [0.46435916],
       [0.4680053 ],
       [0.47168982],
       [0.47540027],
       [0.4790874 ],
       [0.48271877],
       [0.48624915],
       [0.48960483],
       [0.49271375],
       [0.49550122],
       [0.49792176],
       [0.49999493],
       [0.5017983 ],
       [0.5034245 ],
       [0.5049754 ],
       [0.50654733],
       [0.5082152 ],
       [0.51003546],
       [0.512036  ],
       [0.5142098 ],
       [0.5165014 ],
       [0.5188703 ],
       [0.5212782 ],
       [0.5236711 ],
       [0.5260156 ],
       [0.5282609 ],
       [0.53029585],
       [0.5320204 ],
       [0.5333587 ],
       [0.5343248 ],
       [0.53497756],
       [0.5354038 ],
       [0.53568625],
```

```
[0.53588015],
[0.5360255 ],
[0.53613055],
[0.53614557],
[0.5359993 ],
[0.5356134 ],
[0.5349472 ],
[0.5340061 ],
[0.53279394],
[0.5313188 ],
[0.52960783],
[0.5276843 ],
[0.5255892 ],
[0.5233154 ],
[0.52089643],
[0.5184022 ],
[0.5159359 ],
[0.5135918 ],
[0.511435  ],
[0.509524  ],
[0.50791746],
[0.506639  ],
[0.50566006],
[0.50491774],
[0.5043333 ],
[0.5038504 ],
[0.50343925],
[0.5030475 ],
[0.5026226 ],
[0.50211406],
[0.50147563],
[0.50060034],
[0.49936354],
[0.4976843 ],
[0.4954952 ],
[0.49278754],
[0.48952967],
[0.48563606],
[0.48105395],
[0.47563672],
[0.4693455 ],
[0.46214092],
[0.4540136 ],
[0.44505602],
[0.43479836],
```

```
[0.42250687],
[0.40690517],
[0.38483703],
[0.35312226],
[0.31080604],
[0.26455432],
[0.22330375],
[0.1903993 ],
[0.1655853 ],
[0.14708012],
[0.13440783],
[0.126908  ],
[0.1237844 ],
[0.12368607],
[0.12528852],
[0.12794265],
[0.13132668],
[0.13519497],
[0.13912007],
[0.14280492],
[0.14617266],
[0.14912626],
[0.15165877],
[0.15390873],
[0.15602237],
[0.15810871],
[0.1601043 ],
[0.16184914],
[0.16323034],
[0.16402079],
[0.16407081],
[0.16318804],
[0.16144611],
[0.15919222],
[0.15664576],
[0.15393925],
[0.15085435],
[0.14721812],
[0.14310834],
[0.1386821 ],
[0.13400897],
[0.12928036],
[0.12511572],
[0.12211743],
[0.12064645],
```

```
       [0.12099706],
       [0.12313305],
       [0.12629189],
       [0.13032699],
       [0.13491973],
       [0.13995093],
       [0.14508107],
       [0.15000324],
       [0.15456983],
       [0.15872225],
       [0.16236949],
       [0.16546503],
       [0.16795145],
       [0.16977611],
       [0.17094727],
       [0.17171288],
       [0.17247319],
       [0.17339149],
       [0.17449692],
       [0.17578726],
       [0.17722504],
       [0.1787114 ],
       [0.18023925],
       [0.18178187],
       [0.1833051 ],
       [0.18485628],
       [0.18648322],
       [0.18821512],
       [0.1900456 ],
       [0.19195935],
       [0.19390939],
       [0.19570409],
       [0.1970764 ],
       [0.19788627],
       [0.19813076],
       [0.19794051],
       [0.19751716],
       [0.19711511],
       [0.19691081],
       [0.19696645],
       [0.19722526],
       [0.19759652],
       [0.19798006],
       [0.19825323],
       [0.19830665],
```

```
       [0.19797434],
       [0.19715227],
       [0.19588701],
       [0.19427691],
       [0.19250175],
       [0.19076431],
       [0.18931851],
       [0.18829527],
       [0.18774046],
       [0.18757862],
       [0.18768655],
       [0.18797661],
       [0.18842848],
       [0.18898539],
       [0.18961602],
       [0.1903396 ],
       [0.19118622],
       [0.19219476],
       [0.19341789],
       [0.19495706],
       [0.19688326],
       [0.1991821 ],
       [0.20173825],
       [0.20445524],
       [0.20717707],
       [0.2097379 ],
       [0.21194582],
       [0.21356073],
       [0.21447599],
       [0.21461774],
       [0.21391588],
       [0.21247663],
       [0.21048592],
       [0.20812504],
       [0.20556429],
       [0.20288296],
       [0.2000694 ],
       [0.1970749 ],
       [0.19373514],
       [0.19008783],
       [0.1863951 ],
       [0.18289961],
       [0.17981161],
       [0.17753895],
       [0.17621657],
```

```
       [0.1758467 ],
       [0.17627063],
       [0.17727557],
       [0.17880516],
       [0.18077953],
       [0.18306884],
       [0.18563096],
       [0.18846497],
       [0.19150035],
       [0.19465008],
       [0.1978572 ],
       [0.20111859],
       [0.20447484],
       [0.20792641],
       [0.21138138],
       [0.21475202],
       [0.21793711],
       [0.22083181],
       [0.22333689],
       [0.22542793],
       [0.22725506],
       [0.2290431 ],
       [0.23106968],
       [0.23361295],
       [0.2368943 ],
       [0.24112469],
       [0.24639946],
       [0.25272465],
       [0.25997227],
       [0.2679757 ],
       [0.2765972 ],
       [0.28563488],
       [0.29486227],
       [0.3040431 ],
       [0.31302756],
       [0.3216647 ],
       [0.32985896],
       [0.33755344],
       [0.34472835],
       [0.35140544],
       [0.35754305],
       [0.36318266],
       [0.36842322],
       [0.37337407],
       [0.37816924],
```

```
       [0.38289273],
       [0.38760805],
       [0.39237028],
       [0.3972035 ],
       [0.4021034 ],
       [0.40705776],
       [0.4120307 ],
       [0.41685253],
       [0.4213503 ],
       [0.4254043 ],
       [0.42899728],
       [0.4322086 ],
       [0.43518358],
       [0.4380682 ],
       [0.44098216],
       [0.4440018 ],
       [0.44720644],
       [0.45065176],
       [0.45437294],
       [0.45836943],
       [0.4625978 ],
       [0.46701592],
       [0.47159058],
       [0.47627634],
       [0.48099613],
       [0.48562998],
       [0.4901017 ],
       [0.4943692 ],
       [0.49838626],
       [0.5020398 ],
       [0.5052245 ],
       [0.50779873],
       [0.50965637],
       [0.5107564 ],
       [0.511284  ],
       [0.5115764 ],
       [0.5120284 ],
       [0.51302594],
       [0.5148826 ],
       [0.51781285],
       [0.5219114 ],
       [0.52714294],
       [0.53334916],
       [0.5403272 ],
       [0.54794186],
```

```
       [0.5560458 ],
       [0.56446296],
       [0.5729811 ],
       [0.58133197],
       [0.5892213 ],
       [0.5963979 ],
       [0.6027894 ],
       [0.60843897],
       [0.61331165],
       [0.61742675],
       [0.62084967],
       [0.6237527 ],
       [0.6262911 ],
       [0.6285505 ],
       [0.6305825 ],
       [0.6324703 ],
       [0.63430417],
       [0.6361227 ],
       [0.6379078 ],
       [0.63960314],
       [0.64109904],
       [0.64226305],
       [0.643011  ],
       [0.64327246],
       [0.6430507 ],
       [0.64233196],
       [0.6410948 ],
       [0.6393551 ],
       [0.6371978 ],
       [0.63470113],
       [0.63197976],
       [0.629076  ],
       [0.6259957 ],
       [0.62276953],
       [0.6194291 ],
       [0.6159883 ],
       [0.61235136],
       [0.6084679 ],
       [0.6043499 ],
       [0.6000443 ],
       [0.5955698 ],
       [0.59093946],
       [0.58621395],
       [0.58147246],
       [0.57683235],
```

```
       [0.5724424 ],
       [0.56850046],
       [0.5651878 ],
       [0.5625768 ],
       [0.56064636],
       [0.55930775],
       [0.55847526],
       [0.5580642 ],
       [0.5579976 ],
       [0.55821776],
       [0.55865777],
       [0.5592249 ],
       [0.559816  ],
       [0.56041956],
       [0.5610904 ],
       [0.56189096],
       [0.5628492 ],
       [0.56406265],
       [0.56565076],
       [0.56768155],
       [0.57017195],
       [0.5731181 ],
       [0.5764831 ],
       [0.5802259 ],
       [0.5842685 ],
       [0.5885135 ],
       [0.5928776 ],
       [0.59725845],
       [0.6015686 ],
       [0.60570204],
       [0.6095411 ],
       [0.61301816],
       [0.6161029 ],
       [0.618799  ],
       [0.6211577 ],
       [0.6232303 ],
       [0.62504727],
       [0.6266404 ],
       [0.6280711 ],
       [0.629397  ],
       [0.6306592 ],
       [0.63190246],
       [0.6332067 ],
       [0.6346442 ],
       [0.6362436 ],
```

```
       [0.6379824 ],
       [0.63980955],
       [0.64167553],
       [0.64356726],
       [0.64550835],
       [0.6475424 ],
       [0.6496645 ],
       [0.6518289 ],
       [0.6539865 ],
       [0.65612644],
       [0.6582457 ],
       [0.6603569 ],
       [0.6624585 ],
       [0.6644874 ],
       [0.6663118 ],
       [0.66782635],
       [0.6689911 ],
       [0.66980255],
       [0.6702716 ],
       [0.6704014 ],
       [0.67021644],
       [0.66974884],
       [0.6690506 ],
       [0.6682173 ],
       [0.66740763],
       [0.66676176],
       [0.6663818 ],
       [0.6663422 ],
       [0.6666784 ],
       [0.6673712 ],
       [0.6683736 ],
       [0.6696364 ],
       [0.6710964 ],
       [0.67266834],
       [0.6742405 ],
       [0.67567146],
       [0.67684484],
       [0.67773   ],
       [0.6783369 ],
       [0.6787048 ],
       [0.678892  ],
       [0.6790144 ],
       [0.679206  ],
       [0.67959666],
       [0.6802995 ],
```

```
       [0.68137693],
       [0.68282837],
       [0.68460107],
       [0.68663275],
       [0.6888448 ],
       [0.69113755],
       [0.693424  ],
       [0.69564843],
       [0.6978327 ],
       [0.70002615],
       [0.7022215 ],
       [0.7043996 ],
       [0.70651877],
       [0.7085962 ],
       [0.71067035],
       [0.712806  ],
       [0.7150438 ],
       [0.7173789 ],
       [0.71975565],
       [0.72208667],
       [0.72431445],
       [0.72641563],
       [0.728369  ],
       [0.7301843 ],
       [0.73188543],
       [0.73354113],
       [0.7352475 ],
       [0.7371033 ],
       [0.73922896],
       [0.7417263 ],
       [0.74462306],
       [0.74788433],
       [0.75146925],
       [0.7553381 ],
       [0.7594928 ],
       [0.76391876],
       [0.76853573],
       [0.7731076 ],
       [0.7773505 ],
       [0.78108704],
       [0.7842437 ],
       [0.78677404],
       [0.78868866],
       [0.79003376],
       [0.7908832 ],
```

```
       [0.79130656],
       [0.7913383 ],
       [0.7910253 ],
       [0.79042363],
       [0.78955984],
       [0.7884379 ],
       [0.78707254],
       [0.7854123 ],
       [0.7834163 ],
       [0.78109306],
       [0.7784817 ],
       [0.7755555 ],
       [0.7722781 ],
       [0.7686094 ],
       [0.7645766 ],
       [0.76026183],
       [0.7557562 ],
       [0.7511241 ],
       [0.7464794 ],
       [0.7419929 ],
       [0.73782635],
       [0.73411226],
       [0.7309315 ],
       [0.7283224 ],
       [0.72628367],
       [0.72477067],
       [0.723652  ],
       [0.7226929 ],
       [0.7216589 ],
       [0.72038674],
       [0.7187934 ],
       [0.7168369 ],
       [0.7145281 ],
       [0.71192247],
       [0.7090912 ],
       [0.70611227],
       [0.70308745],
       [0.70017606],
       [0.69755864],
       [0.6954392 ],
       [0.69397664],
       [0.6932875 ],
       [0.6934543 ],
       [0.69448894],
       [0.6963477 ],
```

```
       [0.6989062 ],
       [0.70198023],
       [0.7053673 ],
       [0.7088721 ],
       [0.71231186],
       [0.7155241 ],
       [0.71836627],
       [0.7206948 ],
       [0.7224742 ],
       [0.7237639 ],
       [0.7246537 ],
       [0.7252815 ],
       [0.7258459 ],
       [0.7266053 ],
       [0.727759  ],
       [0.72941124],
       [0.7315223 ],
       [0.7339897 ],
       [0.7367362 ],
       [0.73972356],
       [0.7428736 ],
       [0.74593824],
       [0.7487695 ],
       [0.75127554],
       [0.75337535],
       [0.7550253 ],
       [0.7562248 ],
       [0.7569791 ],
       [0.757323  ],
       [0.7571262 ],
       [0.75632584],
       [0.75490797],
       [0.7528523 ],
       [0.75012904],
       [0.7466961 ],
       [0.742434  ],
       [0.7372596 ],
       [0.7311943 ],
       [0.72437984],
       [0.7170265 ],
       [0.7094178 ],
       [0.7018267 ],
       [0.69453233],
       [0.6878209 ],
       [0.6819    ],
```

```
       [0.67691475],
       [0.6729277 ],
       [0.66989154],
       [0.66769046],
       [0.66620064],
       [0.66530585],
       [0.66493046],
       [0.66501534],
       [0.6655606 ],
       [0.6666788 ],
       [0.66843027],
       [0.67079043],
       [0.6736775 ],
       [0.6770027 ],
       [0.6806506 ],
       [0.68451047],
       [0.68845075],
       [0.6922902 ],
       [0.695828  ],
       [0.69889414],
       [0.7014174 ],
       [0.7033309 ],
       [0.70461345],
       [0.7053124 ],
       [0.7054918 ],
       [0.7052627 ],
       [0.70472616],
       [0.70397913],
       [0.7030402 ],
       [0.7019056 ],
       [0.7005259 ],
       [0.6988276 ],
       [0.6967636 ],
       [0.69433343],
       [0.69149613],
       [0.688192  ],
       [0.6844127 ],
       [0.6802401 ],
       [0.6757948 ],
       [0.67115414],
       [0.6664398 ],
       [0.661782  ],
       [0.6573048 ],
       [0.6531155 ],
       [0.6493435 ],
```

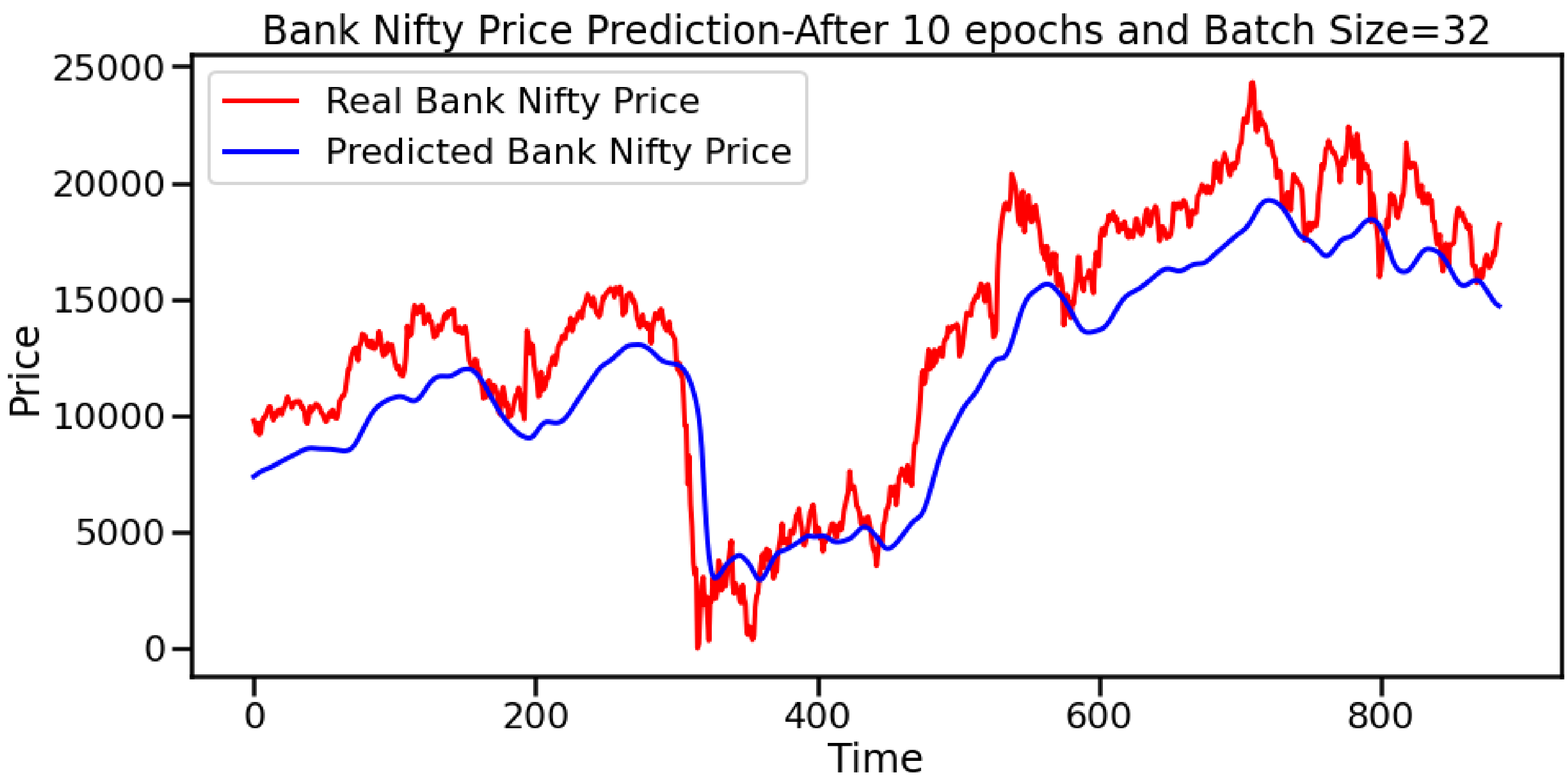
```
       [0.6461303 ],
       [0.6436073 ],
       [0.6418404 ],
       [0.64085096],
       [0.6406192 ],
       [0.641057  ],
       [0.64203763],
       [0.6433923 ],
       [0.6449465 ],
       [0.6465528 ],
       [0.6480553 ],
       [0.64925593],
       [0.6499867 ],
       [0.6501334 ],
       [0.6495917 ],
       [0.6483096 ],
       [0.6462817 ],
       [0.6435793 ],
       [0.6402948 ],
       [0.63655615],
       [0.63253075],
       [0.6284049 ],
       [0.624342  ],
       [0.6204503 ],
       [0.6168095 ],
       [0.6134818 ],
       [0.6105354 ],
       [0.6080185 ],
       [0.60598946],
       [0.6045278 ]], dtype=float32)
```

In [31]:
```
scale = 1/scalar.scale_[0]
```

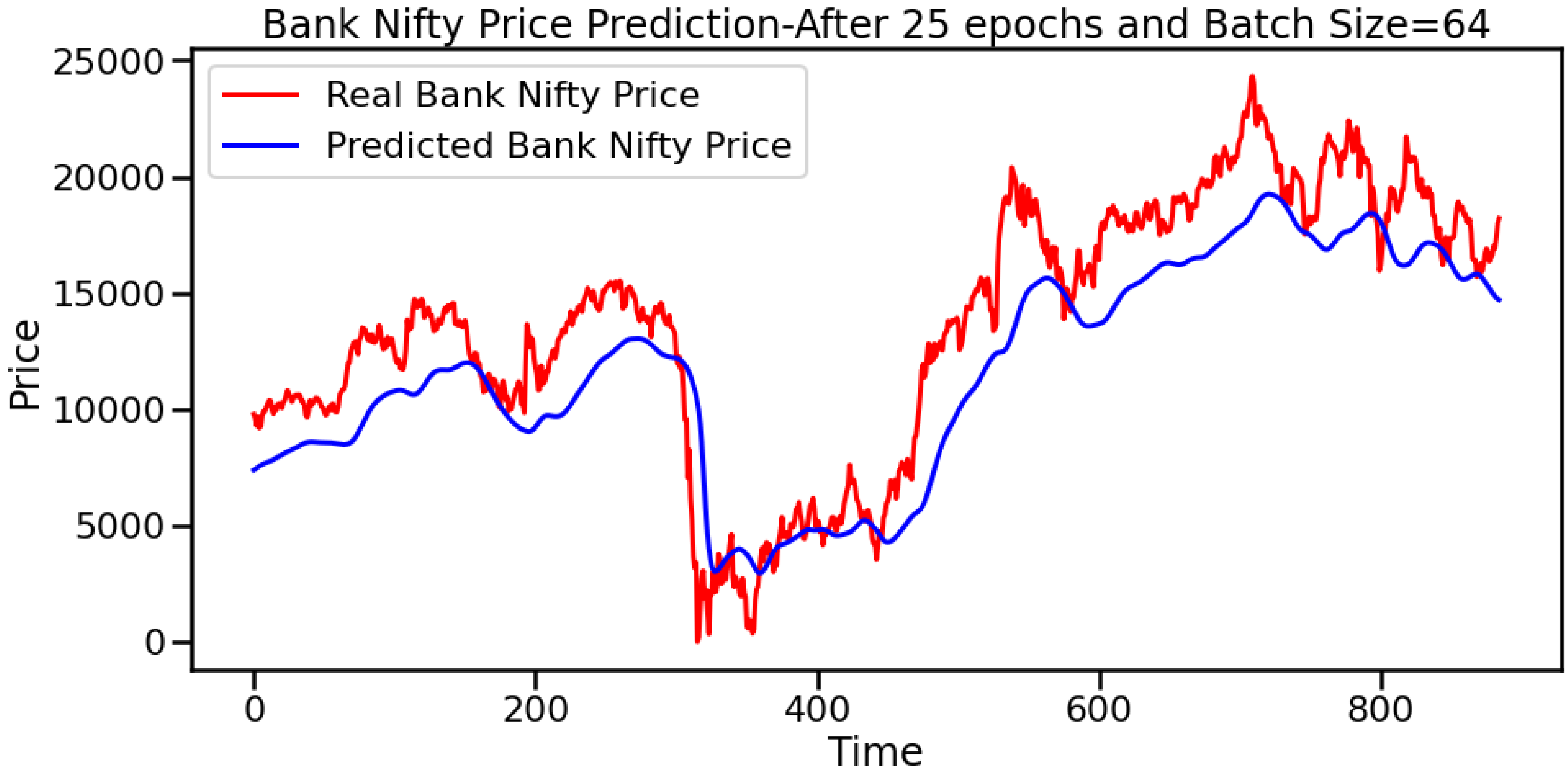
In [32]:
```
y_pred = y_pred*scale
y_test = y_test*scale
```

In [33]:
```
# Visualising the results

plt.figure(figsize=(15,7))
plt.plot(y_test, color = 'red', label = 'Real Bank Nifty Price')
plt.plot(y_pred, color = 'blue', label = 'Predicted Bank Nifty Price')
plt.title('Bank Nifty Price Prediction-After 10 epochs and Batch Size=32')
plt.xlabel('Time')
plt.ylabel('Price')
```

```
plt.legend()
plt.show()
```

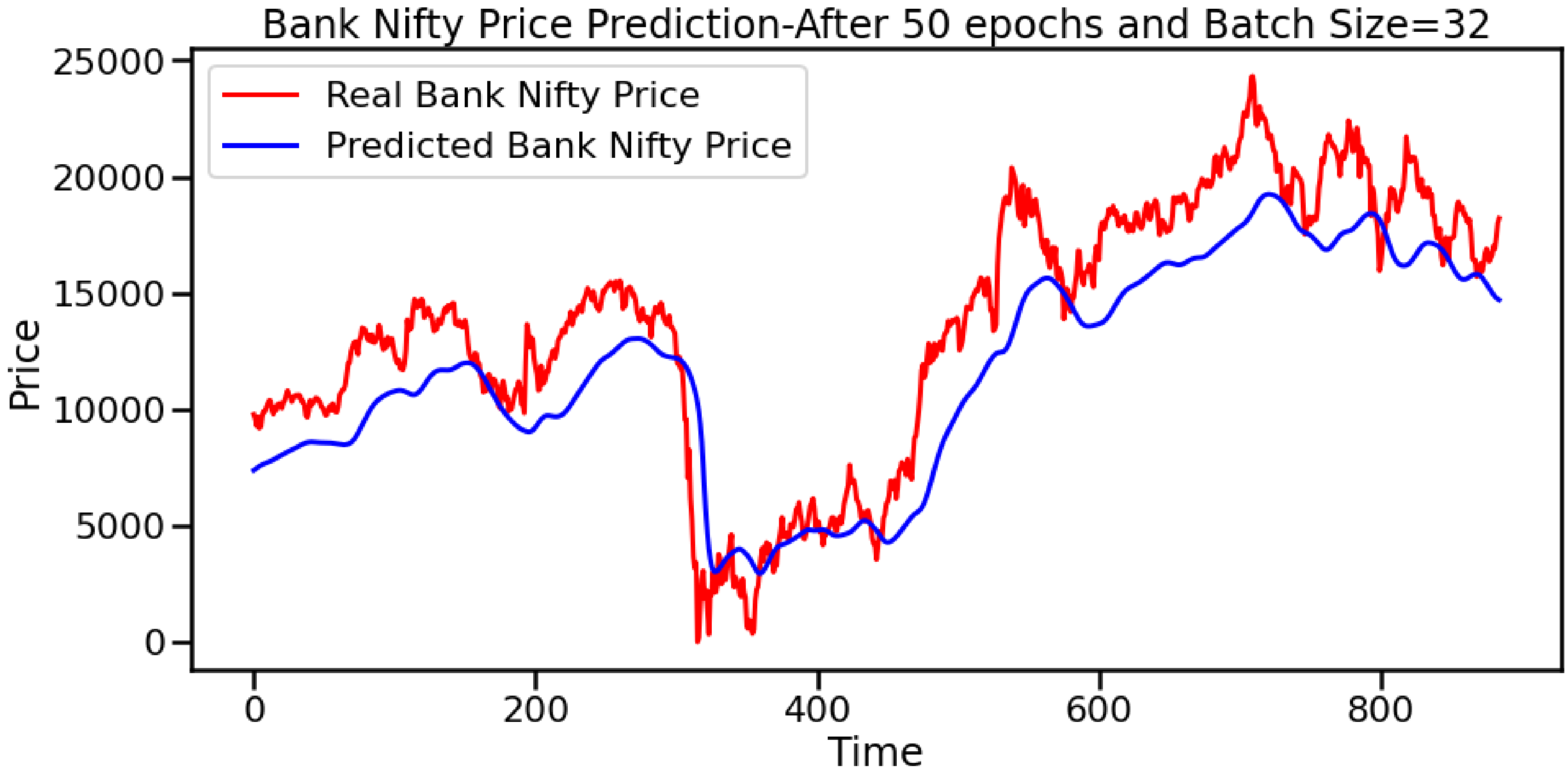


In [34]:
```
# Visualising the results

plt.figure(figsize=(15,7))
plt.plot(y_test, color = 'red', label = 'Real Bank Nifty Price')
plt.plot(y_pred, color = 'blue', label = 'Predicted Bank Nifty Price')
plt.title('Bank Nifty Price Prediction-After 25 epochs and Batch Size=64')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
```



In [35]:

```
# Visualising the results

plt.figure(figsize=(15,7))
plt.plot(y_test, color = 'red', label = 'Real Bank Nifty Price')
plt.plot(y_pred, color = 'blue', label = 'Predicted Bank Nifty Price')
plt.title('Bank Nifty Price Prediction-After 50 epochs and Batch Size=32')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
```


Bank Nifty Price Prediction-After 50 epochs and Batch Size=32
Real Bank Nifty Price
Predicted Bank Nifty Price
Price
Time
25000
20000
15000
10000
5000
0
0
200
400
600
800