```python
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
from sklearn.metrics import mean_squared_error

import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import pandas as pd
import numpy as np
import math
warnings.filterwarnings('ignore')

d =
pd.read_csv('https://raw.githubusercontent.com/mwitiderrick/stockprice
/master/NSE-TATAGLOBAL.csv')
d.head()
```

```
         Date    Open    High  ...   Close  Total Trade Quantity
Turnover (Lacs)
0  2018-09-28  234.05  235.95  ...  233.75               3069914
7162.35
1  2018-09-27  234.55  236.80  ...  233.25               5082859
11859.95
2  2018-09-26  240.00  240.00  ...  234.25               2240909
5248.60
3  2018-09-25  233.30  236.75  ...  236.10               2349368
5503.90
4  2018-09-24  233.55  239.20  ...  233.30               3423509
7999.55

[5 rows x 8 columns]
```

```python
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2035 entries, 0 to 2034
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Date                  2035 non-null   object
 1   Open                  2035 non-null   float64
 2   High                  2035 non-null   float64
 3   Low                   2035 non-null   float64
 4   Last                  2035 non-null   float64
 5   Close                 2035 non-null   float64
 6   Total Trade Quantity  2035 non-null   int64
 7   Turnover (Lacs)       2035 non-null   float64
dtypes: float64(6), int64(1), object(1)
memory usage: 127.3+ KB
```

```
d.describe()
```

```
               Open         High  ...  Total Trade Quantity  Turnover
(Lacs)
count  2035.000000  2035.000000  ...          2.035000e+03
2035.000000
mean    149.713735   151.992826  ...          2.335681e+06
3899.980565
std      48.664509    49.413109  ...          2.091778e+06
4570.767877
min      81.100000    82.800000  ...          3.961000e+04
37.040000
25%     120.025000   122.100000  ...          1.146444e+06
1427.460000
50%     141.500000   143.400000  ...          1.783456e+06
2512.030000
75%     157.175000   159.400000  ...          2.813594e+06
4539.015000
max     327.700000   328.750000  ...          2.919102e+07
55755.080000

[8 rows x 7 columns]
```

```
d['Date'] = pd.to_datetime(d['Date'])
d.dtypes
```

```
Date                    datetime64[ns]
Open                           float64
High                           float64
Low                            float64
Last                           float64
Close                          float64
Total Trade Quantity             int64
Turnover (Lacs)                float64
dtype: object
```

```
d = d.sort_values('Date')
d.head()
```

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

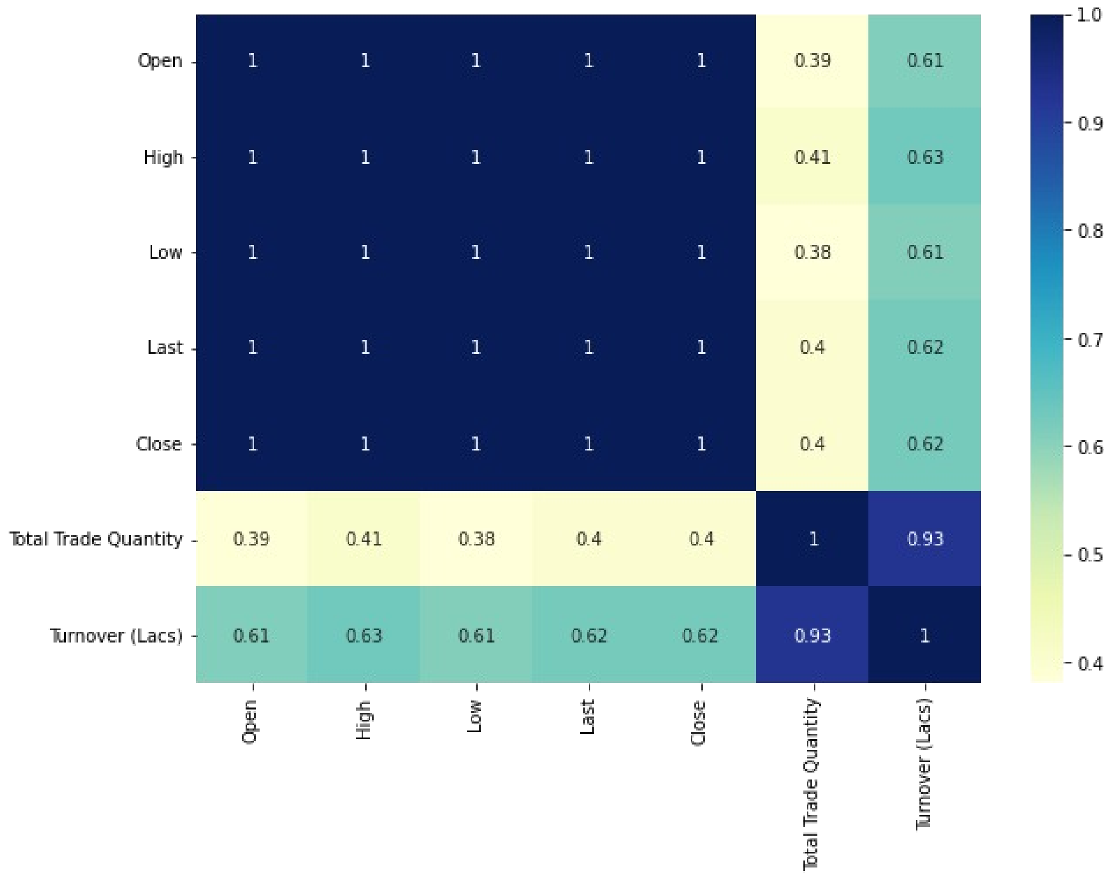
```
[5 rows x 8 columns]
```

```
plt.figure(figsize = (9,6))
plt.title('Tata Stocks Closing Price')
plt.plot(d['Close'],'g')
plt.xlabel('Date',fontsize=15)
plt.ylabel('Close',fontsize=15)
```

```
Text(0, 0.5, 'Close')
```


```
dcorr = d.corr()
top_corr_features = dcorr.index
plt.figure(figsize=(10,7))
sns.heatmap(d[top_corr_features].corr(), annot=True, cmap="YlGnBu")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f68da818e50>
```



MinMaxScaler

From the original dataset, we can tell that each of our target value are in close proximity to one another. So, we will use MinMaxScaler to scale down all the target variables in the range of (0, 1) for the ease of computation.

```
data_close = d.reset_index()['Close']
data_close.head()
scaler = MinMaxScaler(feature_range = (0, 1))
data_close = scaler.fit_transform(np.array(data_close).reshape(-1, 1))
```

Splitting train, Test data

```
train_size = int(len(data_close)*0.70)
test_size = len(data_close) - train_size
train, test = data_close[0 : train_size, :], data_close[train_size :
len(data_close), :1]

def create_matrix(ds, time_step=1):
    dataX, dataY = [], []
    for i in range(len(ds)-time_step-1):
        a = ds[i:(i+time_step),0]
        dataX.append(a)
        dataY.append(ds[i+time_step,0])
    return np.array(dataX), np.array(dataY)
```

```
step=100
X_train, y_train = create_matrix(train, step)
X_test, y_test = create_matrix(test, step)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(1323, 100) (1323,)
(510, 100) (510,)
```

```
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
```

LSTM Model

```
model = Sequential()
model.add(LSTM(50, return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

```
history = model.fit(X_train, y_train, validation_split=0.1, epochs=77,
batch_size=64, verbose=1, shuffle=True).history
```

```
Epoch 1/77
19/19 [==============================] - 8s 177ms/step - loss: 0.0099
- val_loss: 0.0021
Epoch 2/77
19/19 [==============================] - 2s 119ms/step - loss: 0.0017
- val_loss: 0.0016
Epoch 3/77
19/19 [==============================] - 2s 120ms/step - loss:
9.6523e-04 - val_loss: 0.0013
```

```
Epoch 4/77
19/19 [==============================] - 2s 120ms/step - loss:
8.8069e-04 - val_loss: 0.0012
Epoch 5/77
19/19 [==============================] - 2s 120ms/step - loss:
8.7105e-04 - val_loss: 0.0012
Epoch 6/77
19/19 [==============================] - 2s 119ms/step - loss:
8.4772e-04 - val_loss: 0.0011
Epoch 7/77
19/19 [==============================] - 2s 120ms/step - loss:
8.2906e-04 - val_loss: 0.0011
Epoch 8/77
19/19 [==============================] - 2s 121ms/step - loss:
8.0207e-04 - val_loss: 0.0011
Epoch 9/77
19/19 [==============================] - 2s 117ms/step - loss:
7.6832e-04 - val_loss: 0.0011
Epoch 10/77
19/19 [==============================] - 2s 119ms/step - loss:
7.4760e-04 - val_loss: 9.9156e-04
Epoch 11/77
19/19 [==============================] - 2s 120ms/step - loss:
7.2058e-04 - val_loss: 9.1899e-04
Epoch 12/77
19/19 [==============================] - 2s 119ms/step - loss:
7.1255e-04 - val_loss: 8.6823e-04
Epoch 13/77
19/19 [==============================] - 2s 120ms/step - loss:
6.8644e-04 - val_loss: 8.3772e-04
Epoch 14/77
19/19 [==============================] - 2s 122ms/step - loss:
6.6018e-04 - val_loss: 9.0543e-04
Epoch 15/77
19/19 [==============================] - 2s 121ms/step - loss:
6.7876e-04 - val_loss: 8.0936e-04
Epoch 16/77
19/19 [==============================] - 2s 120ms/step - loss:
6.6418e-04 - val_loss: 7.5896e-04
Epoch 17/77
19/19 [==============================] - 2s 121ms/step - loss:
6.3820e-04 - val_loss: 8.2191e-04
Epoch 18/77
19/19 [==============================] - 2s 121ms/step - loss:
6.3613e-04 - val_loss: 6.8540e-04
Epoch 19/77
19/19 [==============================] - 2s 119ms/step - loss:
6.2921e-04 - val_loss: 7.7650e-04
Epoch 20/77
19/19 [==============================] - 2s 119ms/step - loss:
```

```
5.9491e-04 - val_loss: 6.3924e-04
Epoch 21/77
19/19 [==============================] - 2s 120ms/step - loss:
5.6257e-04 - val_loss: 7.2460e-04
Epoch 22/77
19/19 [==============================] - 2s 116ms/step - loss:
5.7016e-04 - val_loss: 6.5238e-04
Epoch 23/77
19/19 [==============================] - 2s 121ms/step - loss:
5.4305e-04 - val_loss: 6.1190e-04
Epoch 24/77
19/19 [==============================] - 2s 117ms/step - loss:
5.3482e-04 - val_loss: 5.7764e-04
Epoch 25/77
19/19 [==============================] - 2s 120ms/step - loss:
5.1611e-04 - val_loss: 7.0965e-04
Epoch 26/77
19/19 [==============================] - 2s 122ms/step - loss:
5.1336e-04 - val_loss: 5.6168e-04
Epoch 27/77
19/19 [==============================] - 2s 121ms/step - loss:
4.9406e-04 - val_loss: 5.0449e-04
Epoch 28/77
19/19 [==============================] - 2s 121ms/step - loss:
5.0277e-04 - val_loss: 5.0424e-04
Epoch 29/77
19/19 [==============================] - 2s 120ms/step - loss:
4.9594e-04 - val_loss: 4.8174e-04
Epoch 30/77
19/19 [==============================] - 2s 121ms/step - loss:
4.5865e-04 - val_loss: 5.1015e-04
Epoch 31/77
19/19 [==============================] - 2s 121ms/step - loss:
4.3627e-04 - val_loss: 5.0017e-04
Epoch 32/77
19/19 [==============================] - 2s 120ms/step - loss:
4.5932e-04 - val_loss: 4.4931e-04
Epoch 33/77
19/19 [==============================] - 2s 119ms/step - loss:
4.4421e-04 - val_loss: 4.1998e-04
Epoch 34/77
19/19 [==============================] - 2s 121ms/step - loss:
4.2662e-04 - val_loss: 4.7827e-04
Epoch 35/77
19/19 [==============================] - 2s 121ms/step - loss:
4.3652e-04 - val_loss: 4.0743e-04
Epoch 36/77
19/19 [==============================] - 2s 121ms/step - loss:
4.2853e-04 - val_loss: 4.7656e-04
Epoch 37/77
```

```
19/19 [==============================] - 2s 122ms/step - loss:
4.0684e-04 - val_loss: 3.7947e-04
Epoch 38/77
19/19 [==============================] - 2s 121ms/step - loss:
3.9165e-04 - val_loss: 4.2307e-04
Epoch 39/77
19/19 [==============================] - 2s 116ms/step - loss:
3.7025e-04 - val_loss: 3.7688e-04
Epoch 40/77
19/19 [==============================] - 2s 119ms/step - loss:
3.5431e-04 - val_loss: 3.7179e-04
Epoch 41/77
19/19 [==============================] - 2s 120ms/step - loss:
3.5245e-04 - val_loss: 3.1818e-04
Epoch 42/77
19/19 [==============================] - 2s 117ms/step - loss:
3.4207e-04 - val_loss: 3.0748e-04
Epoch 43/77
19/19 [==============================] - 2s 122ms/step - loss:
3.2870e-04 - val_loss: 3.2794e-04
Epoch 44/77
19/19 [==============================] - 2s 122ms/step - loss:
3.1959e-04 - val_loss: 3.5035e-04
Epoch 45/77
19/19 [==============================] - 2s 119ms/step - loss:
3.3311e-04 - val_loss: 3.4633e-04
Epoch 46/77
19/19 [==============================] - 2s 121ms/step - loss:
3.0250e-04 - val_loss: 2.7626e-04
Epoch 47/77
19/19 [==============================] - 2s 121ms/step - loss:
2.9041e-04 - val_loss: 2.5316e-04
Epoch 48/77
19/19 [==============================] - 2s 121ms/step - loss:
2.7200e-04 - val_loss: 2.4888e-04
Epoch 49/77
19/19 [==============================] - 2s 121ms/step - loss:
2.8571e-04 - val_loss: 2.6561e-04
Epoch 50/77
19/19 [==============================] - 2s 116ms/step - loss:
2.7434e-04 - val_loss: 2.5444e-04
Epoch 51/77
19/19 [==============================] - 2s 120ms/step - loss:
2.5727e-04 - val_loss: 2.4776e-04
Epoch 52/77
19/19 [==============================] - 2s 122ms/step - loss:
2.5447e-04 - val_loss: 2.1893e-04
Epoch 53/77
19/19 [==============================] - 2s 122ms/step - loss:
2.5158e-04 - val_loss: 2.1686e-04
```

```
Epoch 54/77
19/19 [==============================] - 2s 119ms/step - loss:
2.4431e-04 - val_loss: 2.0498e-04
Epoch 55/77
19/19 [==============================] - 2s 122ms/step - loss:
2.4052e-04 - val_loss: 2.0202e-04
Epoch 56/77
19/19 [==============================] - 2s 124ms/step - loss:
2.4500e-04 - val_loss: 2.0017e-04
Epoch 57/77
19/19 [==============================] - 2s 124ms/step - loss:
2.2550e-04 - val_loss: 2.2498e-04
Epoch 58/77
19/19 [==============================] - 2s 118ms/step - loss:
2.5473e-04 - val_loss: 2.3388e-04
Epoch 59/77
19/19 [==============================] - 2s 122ms/step - loss:
2.2358e-04 - val_loss: 1.8385e-04
Epoch 60/77
19/19 [==============================] - 2s 121ms/step - loss:
2.1942e-04 - val_loss: 1.8844e-04
Epoch 61/77
19/19 [==============================] - 2s 121ms/step - loss:
2.1516e-04 - val_loss: 1.7768e-04
Epoch 62/77
19/19 [==============================] - 2s 121ms/step - loss:
2.0536e-04 - val_loss: 1.6565e-04
Epoch 63/77
19/19 [==============================] - 2s 120ms/step - loss:
2.0824e-04 - val_loss: 1.6849e-04
Epoch 64/77
19/19 [==============================] - 2s 121ms/step - loss:
2.1441e-04 - val_loss: 1.7405e-04
Epoch 65/77
19/19 [==============================] - 2s 122ms/step - loss:
2.0268e-04 - val_loss: 1.6122e-04
Epoch 66/77
19/19 [==============================] - 2s 122ms/step - loss:
2.1173e-04 - val_loss: 1.6777e-04
Epoch 67/77
19/19 [==============================] - 2s 121ms/step - loss:
1.8564e-04 - val_loss: 1.5794e-04
Epoch 68/77
19/19 [==============================] - 2s 119ms/step - loss:
1.8235e-04 - val_loss: 1.6955e-04
Epoch 69/77
19/19 [==============================] - 2s 121ms/step - loss:
1.8559e-04 - val_loss: 1.5113e-04
Epoch 70/77
19/19 [==============================] - 2s 122ms/step - loss:
```

```
1.7340e-04 - val_loss: 1.4682e-04
Epoch 71/77
19/19 [==============================] - 2s 120ms/step - loss:
1.8076e-04 - val_loss: 1.4419e-04
Epoch 72/77
19/19 [==============================] - 2s 120ms/step - loss:
1.6943e-04 - val_loss: 2.6988e-04
Epoch 73/77
19/19 [==============================] - 2s 120ms/step - loss:
1.9948e-04 - val_loss: 1.7220e-04
Epoch 74/77
19/19 [==============================] - 2s 120ms/step - loss:
2.0470e-04 - val_loss: 1.5004e-04
Epoch 75/77
19/19 [==============================] - 2s 120ms/step - loss:
1.8241e-04 - val_loss: 1.5724e-04
Epoch 76/77
19/19 [==============================] - 2s 118ms/step - loss:
1.7173e-04 - val_loss: 1.4769e-04
Epoch 77/77
19/19 [==============================] - 2s 122ms/step - loss:
1.7669e-04 - val_loss: 1.3493e-04
```

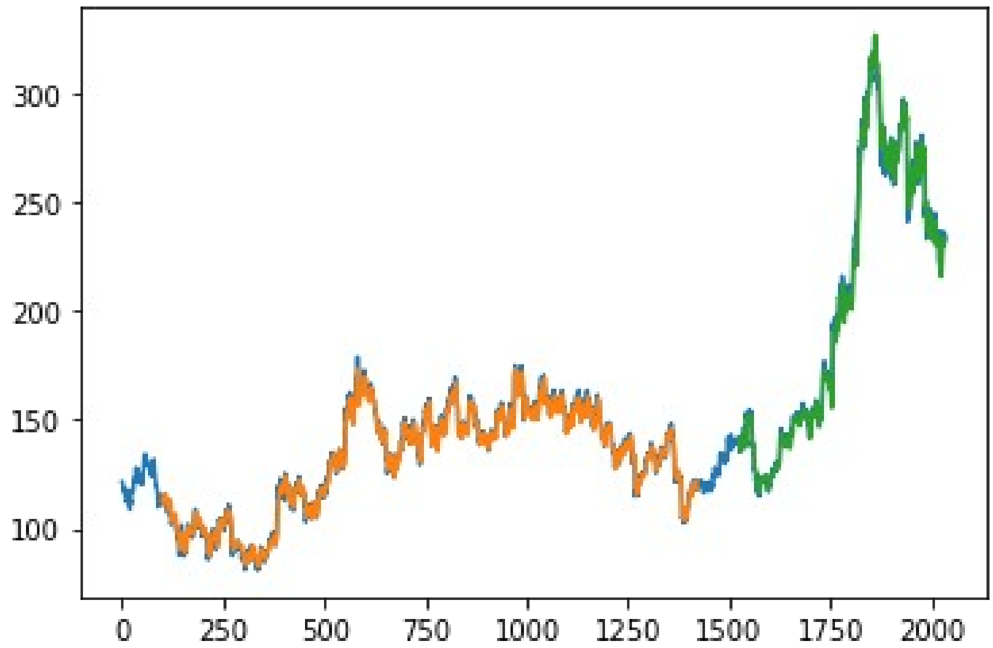
```
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)
```

```
# Reversing the MinMax Scaler
train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)
```

```
math.sqrt(mean_squared_error(y_train, train_predict))
math.sqrt(mean_squared_error(y_test,test_predict))
```

```
212.4209276195953
```

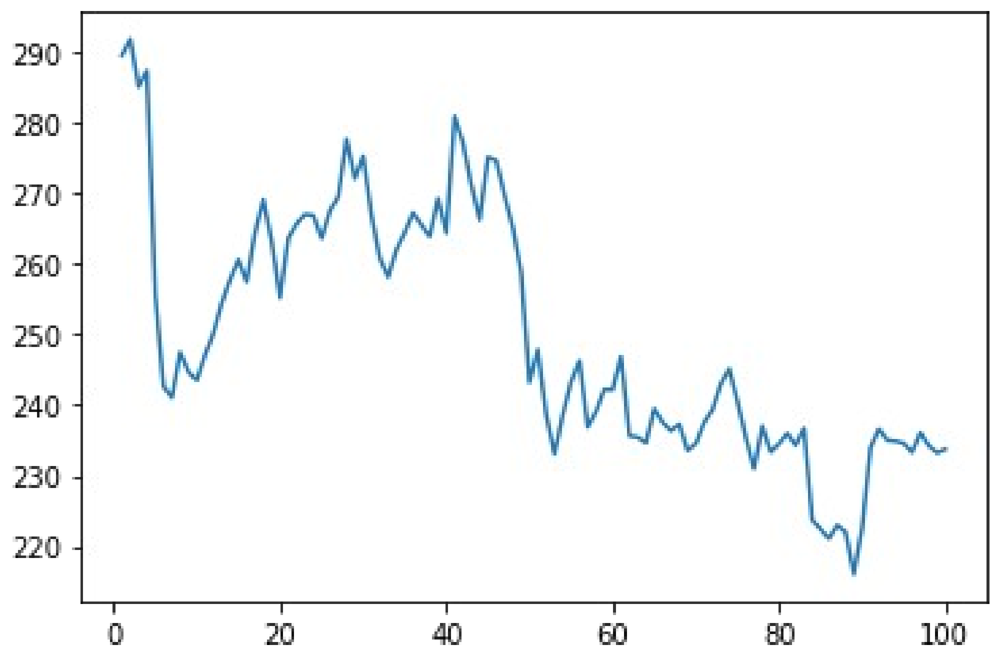
```
### Visualise the Predictions
look_back = 100
train_num_pyredict_plot = np.empty_like(data_close)
train_num_pyredict_plot[:, :] = np.nan
train_num_pyredict_plot[look_back : len(train_predict) + look_back, :]
= train_predict
test_predict_plot = np.empty_like(data_close)
test_predict_plot[:, :] = np.nan
test_predict_plot[len(train_predict) + (look_back * 2) + 1 :
len(data_close) - 1, :] = test_predict
plt.plot(scaler.inverse_transform(data_close))
plt.plot(train_num_pyredict_plot)
plt.plot(test_predict_plot)
plt.show()
```




```
### Future Prediction Model
x_inum_pyut=test[307:].reshape(1, -1)
x_inum_pyut.shape
temp_inum_pyut = list(x_inum_pyut)
temp_inum_pyut = temp_inum_pyut[0].tolist()
temp_inum_pyut = list(x_inum_pyut)
temp_inum_pyut = temp_inum_pyut[0].tolist()

day_new = np.arange(1, 101)
day_pred = np.arange(101, 131)
plt.plot(day_new, scaler.inverse_transform(data_close[1935 : ]))
```

```
[<matplotlib.lines.Line2D at 0x7f68d61a7f50>]
```



290
280
270
260
250
240
230
220
0
20
40
60
80
100