In [116…
```
import pyforest
```

In [117…
```
df = pd.read_csv("HCLTECH.csv")
```

In [118…
```
df
```

Out[118…

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWA
0	2000-01-11	HCLTECH	EQ	580.00	1550.0	1725.00	1492.00	1560.00	1554.45	1582.7
1	2000-01-12	HCLTECH	EQ	1554.45	1560.0	1678.85	1560.00	1678.85	1678.85	1657.0
2	2000-01-13	HCLTECH	EQ	1678.85	1790.0	1813.20	1781.00	1813.20	1813.20	1804.6
3	2000-01-14	HCLTECH	EQ	1813.20	1958.3	1958.30	1835.00	1958.30	1958.30	1939.9
4	2000-01-17	HCLTECH	EQ	1958.30	2115.0	2115.00	1801.65	1801.65	1801.65	1990.5
...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.0	954.50	923.05	930.00	928.80	931.7
5296	2021-04-27	HCLTECH	EQ	928.80	931.0	938.55	923.40	930.30	928.85	928.0
5297	2021-04-28	HCLTECH	EQ	928.85	931.2	935.85	921.75	925.90	923.80	926.6
5298	2021-04-29	HCLTECH	EQ	923.80	929.7	929.70	907.10	910.30	909.55	914.3
5299	2021-04-30	HCLTECH	EQ	909.55	905.0	915.00	895.40	900.10	898.95	904.9

5300 rows × 15 columns

In [119…
```
df.index
```

Out[119…
```
RangeIndex(start=0, stop=5300, step=1)
```

In [120…
```
df = df[(df['Date'] > '2014-12-31')]
```

In [121…
```
df
```

Out[121…

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VW.
3734	2015-01-01	HCLTECH	EQ	1596.90	1599.00	1611.65	1585.60	1605.50	1606.80	1604.
3735	2015-01-02	HCLTECH	EQ	1606.80	1602.05	1618.95	1600.05	1607.00	1605.25	1608.
3736	2015-01-05	HCLTECH	EQ	1605.25	1615.00	1615.00	1568.70	1581.00	1578.25	1586.
3737	2015-01-06	HCLTECH	EQ	1578.25	1574.00	1574.40	1522.65	1543.05	1536.10	1553.
3738	2015-01-07	HCLTECH	EQ	1536.10	1539.50	1548.00	1492.70	1500.00	1499.60	1517.
...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.00	954.50	923.05	930.00	928.80	931.
5296	2021-04-27	HCLTECH	EQ	928.80	931.00	938.55	923.40	930.30	928.85	928.
5297	2021-04-28	HCLTECH	EQ	928.85	931.20	935.85	921.75	925.90	923.80	926.
5298	2021-04-29	HCLTECH	EQ	923.80	929.70	929.70	907.10	910.30	909.55	914.
5299	2021-04-30	HCLTECH	EQ	909.55	905.00	915.00	895.40	900.10	898.95	904.

1566 rows × 15 columns

In [122…

```
df1 = df.reset_index()
```

In [123…

```
df1
```

Out[123…

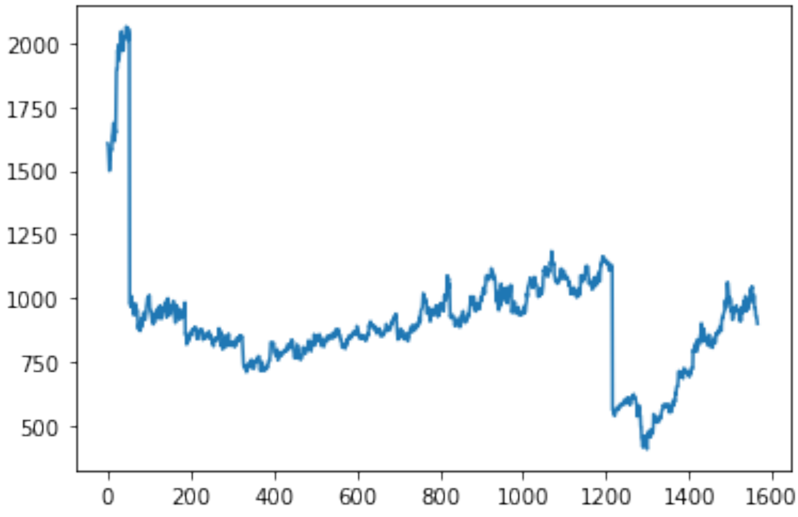
	index	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close
0	3734	2015-01-01	HCLTECH	EQ	1596.90	1599.00	1611.65	1585.60	1605.50	1606.80
1	3735	2015-01-02	HCLTECH	EQ	1606.80	1602.05	1618.95	1600.05	1607.00	1605.25
2	3736	2015-01-05	HCLTECH	EQ	1605.25	1615.00	1615.00	1568.70	1581.00	1578.25
3	3737	2015-01-06	HCLTECH	EQ	1578.25	1574.00	1574.40	1522.65	1543.05	1536.10
4	3738	2015-01-07	HCLTECH	EQ	1536.10	1539.50	1548.00	1492.70	1500.00	1499.60
...	...	...	...	...	...	...	...	...	...	...
1561	5295	2021-04-26	HCLTECH	EQ	955.65	940.00	954.50	923.05	930.00	928.80
1562	5296	2021-04-27	HCLTECH	EQ	928.80	931.00	938.55	923.40	930.30	928.85
1563	5297	2021-04-28	HCLTECH	EQ	928.85	931.20	935.85	921.75	925.90	923.80
1564	5298	2021-04-29	HCLTECH	EQ	923.80	929.70	929.70	907.10	910.30	909.55
1565	5299	2021-04-30	HCLTECH	EQ	909.55	905.00	915.00	895.40	900.10	898.95

1566 rows × 16 columns

In [124…

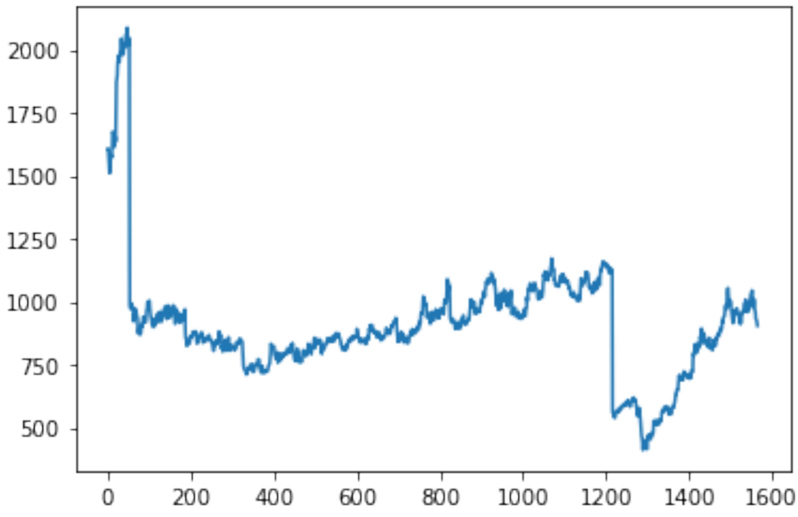
```
plt.plot(df1['Close'])
```

Out[124…
```
[<matplotlib.lines.Line2D at 0x7fcf99e08370>]
```



In [125…
```python
plt.plot(df1['VWAP'])
```

Out[125…
```
[<matplotlib.lines.Line2D at 0x7fcf9973d8b0>]
```



In [126…
```python
df1 = df1['VWAP']
```

In [127…
```python
df1
```

Out[127…
```
0       1604.76
1       1608.73
2       1586.98
3       1553.35
4       1517.47
          ...
1561     931.70
1562     928.06
1563     926.63
1564     914.34
1565     904.98
Name: VWAP, Length: 1566, dtype: float64
```

In [128…
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))
```

In [129…
```
df1
```

Out[129…
```
array([[0.71168799],
       [0.71405879],
       [0.70107014],
       ...,
       [0.30672304],
       [0.29938371],
       [0.29379412]])
```

In [130…
```
training_size=int(len(df1)*0.70)
test_size=len(df1)-training_size
train_data,test_data=df1[0:training_size,:],df1[training_size:len(df1),:1]
```

In [131…
```
training_size,test_size
```

Out[131…
```
(1096, 470)
```

In [132…
```
import numpy
# convert an array of values into a dataset matrix
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]###i=0, 0,1,2,3-----99   100

        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return numpy.array(dataX), numpy.array(dataY)
```

In [133…
```
time_step = 100
X_train, y_train = create_dataset(train_data, time_step)
X_test, ytest = create_dataset(test_data, time_step)
```

In [134…
```
X_train[0].shape
```

Out[134…
```
(100,)
```

In [135…
```
X_train.shape
```

Out[135…
```
(995, 100)
```

In [136…
```
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)
```

In [137…

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dropout
```

In [138…

```
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(Dropout(0.2))
model.add(LSTM(50,return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

In [139…

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_7 (LSTM)                (None, 100, 50)           10400     
_________________________________________________________________
dropout_2 (Dropout)          (None, 100, 50)           0         
_________________________________________________________________
lstm_8 (LSTM)                (None, 100, 50)           20200     
_________________________________________________________________
dropout_3 (Dropout)          (None, 100, 50)           0         
_________________________________________________________________
lstm_9 (LSTM)                (None, 50)                20200     
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51        
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

In [140…

```
model_history = model.fit(X_train,y_train,validation_data=(X_test,ytest),ep
```

```
Epoch 1/100
16/16 [==============================] - 9s 211ms/step - loss: 0.0459 - val
_loss: 0.0067
Epoch 2/100
16/16 [==============================] - 2s 154ms/step - loss: 0.0035 - val
_loss: 0.0084
Epoch 3/100
16/16 [==============================] - 2s 157ms/step - loss: 0.0014 - val
_loss: 0.0080
Epoch 4/100
16/16 [==============================] - 3s 159ms/step - loss: 0.0011 - val
_loss: 0.0070
Epoch 5/100
16/16 [==============================] - 3s 161ms/step - loss: 0.0011 - val
_loss: 0.0058
Epoch 6/100
16/16 [==============================] - 3s 163ms/step - loss: 8.1826e-04 -
val_loss: 0.0049
Epoch 7/100
```

```
16/16 [==============================] - 3s 166ms/step - loss: 7.2642e-04 -
val_loss: 0.0040
Epoch 8/100
16/16 [==============================] - 3s 167ms/step - loss: 5.8487e-04 -
val_loss: 0.0034
Epoch 9/100
16/16 [==============================] - 3s 170ms/step - loss: 6.4277e-04 -
val_loss: 0.0036
Epoch 10/100
16/16 [==============================] - 3s 168ms/step - loss: 5.5356e-04 -
val_loss: 0.0032
Epoch 11/100
16/16 [==============================] - 3s 171ms/step - loss: 5.6640e-04 -
val_loss: 0.0033
Epoch 12/100
16/16 [==============================] - 3s 172ms/step - loss: 5.5360e-04 -
val_loss: 0.0036
Epoch 13/100
16/16 [==============================] - 3s 175ms/step - loss: 5.4012e-04 -
val_loss: 0.0033
Epoch 14/100
16/16 [==============================] - 3s 174ms/step - loss: 5.3940e-04 -
val_loss: 0.0032
Epoch 15/100
16/16 [==============================] - 3s 175ms/step - loss: 5.3071e-04 -
val_loss: 0.0034
Epoch 16/100
16/16 [==============================] - 3s 176ms/step - loss: 5.1110e-04 -
val_loss: 0.0032
Epoch 17/100
16/16 [==============================] - 3s 177ms/step - loss: 4.8160e-04 -
val_loss: 0.0031
Epoch 18/100
16/16 [==============================] - 3s 177ms/step - loss: 4.9566e-04 -
val_loss: 0.0031
Epoch 19/100
16/16 [==============================] - 3s 179ms/step - loss: 4.8867e-04 -
val_loss: 0.0030
Epoch 20/100
16/16 [==============================] - 3s 179ms/step - loss: 5.8056e-04 -
val_loss: 0.0029
Epoch 21/100
16/16 [==============================] - 3s 179ms/step - loss: 5.5191e-04 -
val_loss: 0.0029
Epoch 22/100
16/16 [==============================] - 3s 181ms/step - loss: 5.1044e-04 -
val_loss: 0.0029
Epoch 23/100
16/16 [==============================] - 3s 180ms/step - loss: 4.5716e-04 -
val_loss: 0.0030
Epoch 24/100
16/16 [==============================] - 3s 182ms/step - loss: 4.6980e-04 -
val_loss: 0.0028
Epoch 25/100
16/16 [==============================] - 3s 181ms/step - loss: 4.2648e-04 -
val_loss: 0.0028
Epoch 26/100
16/16 [==============================] - 3s 183ms/step - loss: 4.6671e-04 -
val_loss: 0.0030
Epoch 27/100
16/16 [==============================] - 3s 182ms/step - loss: 4.6889e-04 -
```

```
val_loss: 0.0029
Epoch 28/100
16/16 [==============================] - 3s 183ms/step - loss: 4.4751e-04 -
val_loss: 0.0027
Epoch 29/100
16/16 [==============================] - 3s 182ms/step - loss: 4.1759e-04 -
val_loss: 0.0027
Epoch 30/100
16/16 [==============================] - 3s 210ms/step - loss: 4.7441e-04 -
val_loss: 0.0028
Epoch 31/100
16/16 [==============================] - 3s 190ms/step - loss: 5.0300e-04 -
val_loss: 0.0027
Epoch 32/100
16/16 [==============================] - 3s 183ms/step - loss: 4.4247e-04 -
val_loss: 0.0025
Epoch 33/100
16/16 [==============================] - 3s 184ms/step - loss: 4.9162e-04 -
val_loss: 0.0030
Epoch 34/100
16/16 [==============================] - 3s 183ms/step - loss: 5.6066e-04 -
val_loss: 0.0026
Epoch 35/100
16/16 [==============================] - 3s 183ms/step - loss: 4.5899e-04 -
val_loss: 0.0025
Epoch 36/100
16/16 [==============================] - 3s 185ms/step - loss: 4.2949e-04 -
val_loss: 0.0024
Epoch 37/100
16/16 [==============================] - 3s 184ms/step - loss: 4.9201e-04 -
val_loss: 0.0024
Epoch 38/100
16/16 [==============================] - 3s 184ms/step - loss: 4.4461e-04 -
val_loss: 0.0024
Epoch 39/100
16/16 [==============================] - 3s 186ms/step - loss: 4.2793e-04 -
val_loss: 0.0024
Epoch 40/100
16/16 [==============================] - 3s 184ms/step - loss: 4.1838e-04 -
val_loss: 0.0023
Epoch 41/100
16/16 [==============================] - 3s 184ms/step - loss: 4.3077e-04 -
val_loss: 0.0023
Epoch 42/100
16/16 [==============================] - 3s 184ms/step - loss: 4.1951e-04 -
val_loss: 0.0024
Epoch 43/100
16/16 [==============================] - 3s 186ms/step - loss: 4.3685e-04 -
val_loss: 0.0023
Epoch 44/100
16/16 [==============================] - 3s 184ms/step - loss: 4.1043e-04 -
val_loss: 0.0023
Epoch 45/100
16/16 [==============================] - 3s 184ms/step - loss: 3.4503e-04 -
val_loss: 0.0022
Epoch 46/100
16/16 [==============================] - 3s 185ms/step - loss: 3.8354e-04 -
val_loss: 0.0022
Epoch 47/100
16/16 [==============================] - 3s 186ms/step - loss: 3.7161e-04 -
val_loss: 0.0021
```

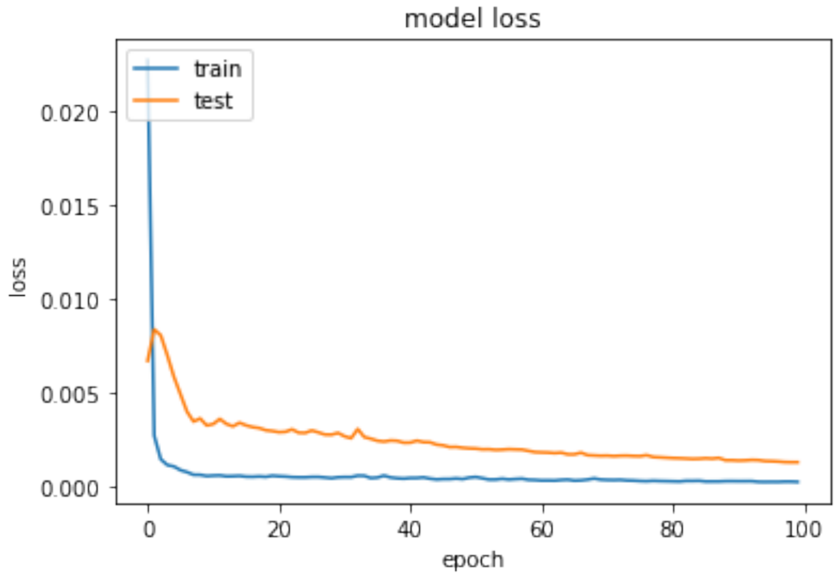
```
Epoch 48/100
16/16 [==============================] - 3s 186ms/step - loss: 4.3026e-04 -
val_loss: 0.0021
Epoch 49/100
16/16 [==============================] - 3s 186ms/step - loss: 3.4986e-04 -
val_loss: 0.0020
Epoch 50/100
16/16 [==============================] - 3s 185ms/step - loss: 4.2257e-04 -
val_loss: 0.0020
Epoch 51/100
16/16 [==============================] - 3s 187ms/step - loss: 4.5924e-04 -
val_loss: 0.0020
Epoch 52/100
16/16 [==============================] - 3s 187ms/step - loss: 4.6267e-04 -
val_loss: 0.0020
Epoch 53/100
16/16 [==============================] - 3s 187ms/step - loss: 3.6313e-04 -
val_loss: 0.0020
Epoch 54/100
16/16 [==============================] - 3s 191ms/step - loss: 3.4840e-04 -
val_loss: 0.0019
Epoch 55/100
16/16 [==============================] - 3s 185ms/step - loss: 3.5764e-04 -
val_loss: 0.0019
Epoch 56/100
16/16 [==============================] - 3s 195ms/step - loss: 3.3540e-04 -
val_loss: 0.0020
Epoch 57/100
16/16 [==============================] - 4s 228ms/step - loss: 3.4757e-04 -
val_loss: 0.0020
Epoch 58/100
16/16 [==============================] - 3s 184ms/step - loss: 3.9833e-04 -
val_loss: 0.0019
Epoch 59/100
16/16 [==============================] - 3s 184ms/step - loss: 3.3683e-04 -
val_loss: 0.0019
Epoch 60/100
16/16 [==============================] - 3s 183ms/step - loss: 3.2252e-04 -
val_loss: 0.0018
Epoch 61/100
16/16 [==============================] - 3s 182ms/step - loss: 2.8940e-04 -
val_loss: 0.0018
Epoch 62/100
16/16 [==============================] - 3s 182ms/step - loss: 2.9624e-04 -
val_loss: 0.0018
Epoch 63/100
16/16 [==============================] - 3s 184ms/step - loss: 2.5924e-04 -
val_loss: 0.0018
Epoch 64/100
16/16 [==============================] - 3s 184ms/step - loss: 3.0748e-04 -
val_loss: 0.0018
Epoch 65/100
16/16 [==============================] - 3s 182ms/step - loss: 3.6770e-04 -
val_loss: 0.0017
Epoch 66/100
16/16 [==============================] - 3s 184ms/step - loss: 3.0845e-04 -
val_loss: 0.0017
Epoch 67/100
16/16 [==============================] - 3s 185ms/step - loss: 2.8446e-04 -
val_loss: 0.0018
Epoch 68/100
```

```
16/16 [==============================] - 3s 185ms/step - loss: 2.9430e-04 -
val_loss: 0.0016
Epoch 69/100
16/16 [==============================] - 3s 184ms/step - loss: 4.1267e-04 -
val_loss: 0.0016
Epoch 70/100
16/16 [==============================] - 3s 183ms/step - loss: 3.4008e-04 -
val_loss: 0.0016
Epoch 71/100
16/16 [==============================] - 3s 184ms/step - loss: 3.0620e-04 -
val_loss: 0.0016
Epoch 72/100
16/16 [==============================] - 3s 191ms/step - loss: 3.0371e-04 -
val_loss: 0.0016
Epoch 73/100
16/16 [==============================] - 3s 186ms/step - loss: 3.2999e-04 -
val_loss: 0.0016
Epoch 74/100
16/16 [==============================] - 3s 184ms/step - loss: 3.1439e-04 -
val_loss: 0.0016
Epoch 75/100
16/16 [==============================] - 3s 182ms/step - loss: 2.9583e-04 -
val_loss: 0.0016
Epoch 76/100
16/16 [==============================] - 3s 184ms/step - loss: 2.6079e-04 -
val_loss: 0.0016
Epoch 77/100
16/16 [==============================] - 3s 184ms/step - loss: 2.5552e-04 -
val_loss: 0.0016
Epoch 78/100
16/16 [==============================] - 3s 178ms/step - loss: 3.0525e-04 -
val_loss: 0.0015
Epoch 79/100
16/16 [==============================] - 3s 177ms/step - loss: 2.9137e-04 -
val_loss: 0.0015
Epoch 80/100
16/16 [==============================] - 3s 176ms/step - loss: 2.5719e-04 -
val_loss: 0.0015
Epoch 81/100
16/16 [==============================] - 3s 176ms/step - loss: 2.2872e-04 -
val_loss: 0.0015
Epoch 82/100
16/16 [==============================] - 3s 177ms/step - loss: 2.4173e-04 -
val_loss: 0.0015
Epoch 83/100
16/16 [==============================] - 3s 176ms/step - loss: 2.8932e-04 -
val_loss: 0.0015
Epoch 84/100
16/16 [==============================] - 3s 178ms/step - loss: 2.8469e-04 -
val_loss: 0.0014
Epoch 85/100
16/16 [==============================] - 4s 231ms/step - loss: 2.6488e-04 -
val_loss: 0.0015
Epoch 86/100
16/16 [==============================] - 3s 189ms/step - loss: 2.1185e-04 -
val_loss: 0.0015
Epoch 87/100
16/16 [==============================] - 3s 186ms/step - loss: 2.4081e-04 -
val_loss: 0.0015
Epoch 88/100
16/16 [==============================] - 3s 206ms/step - loss: 2.2639e-04 -
```

```
val_loss: 0.0015
Epoch 89/100
16/16 [==============================] - 3s 191ms/step - loss: 2.4833e-04 -
val_loss: 0.0014
Epoch 90/100
16/16 [==============================] - 3s 200ms/step - loss: 2.4585e-04 -
val_loss: 0.0014
Epoch 91/100
16/16 [==============================] - 3s 209ms/step - loss: 2.3628e-04 -
val_loss: 0.0014
Epoch 92/100
16/16 [==============================] - 3s 214ms/step - loss: 2.4955e-04 -
val_loss: 0.0014
Epoch 93/100
16/16 [==============================] - 3s 182ms/step - loss: 2.6782e-04 -
val_loss: 0.0014
Epoch 94/100
16/16 [==============================] - 4s 236ms/step - loss: 2.2507e-04 -
val_loss: 0.0014
Epoch 95/100
16/16 [==============================] - 4s 253ms/step - loss: 2.0245e-04 -
val_loss: 0.0013
Epoch 96/100
16/16 [==============================] - 4s 219ms/step - loss: 1.9407e-04 -
val_loss: 0.0013
Epoch 97/100
16/16 [==============================] - 4s 235ms/step - loss: 2.3590e-04 -
val_loss: 0.0013
Epoch 98/100
16/16 [==============================] - 4s 238ms/step - loss: 2.0410e-04 -
val_loss: 0.0013
Epoch 99/100
16/16 [==============================] - 3s 197ms/step - loss: 2.2587e-04 -
val_loss: 0.0013
Epoch 100/100
16/16 [==============================] - 3s 199ms/step - loss: 2.4409e-04 -
val_loss: 0.0013
```

In [141…

```
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

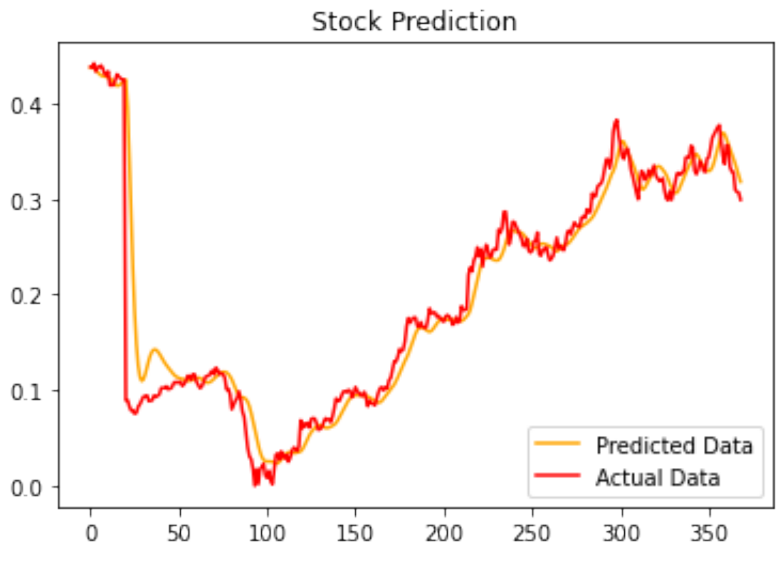


In [142…

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

In [143…

```
plt.plot(test_predict, color='orange', label='Predicted Data')
plt.plot(ytest, color='red', label='Actual Data')
plt.title('Stock Prediction')
plt.legend()
plt.show()
```



In [144…

```
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)
```

In [ ]:

In [145…
```
len(test_data)
```

Out[145… 470

Taking last 100 values for future prediction

In [166…
```
x_input=test_data[370:].reshape(1,-1)
x_input.shape
```

Out[166… (1, 100)

In [167…
```
test_data
```

Out[167…
```
array([[0.4060339 ],
       [0.41626357],
       [0.41049482],
       [0.40967669],
       [0.40515007],
       [0.39813919],
       [0.40082052],
       [0.40322715],
       [0.40023529],
       [0.397333  ],
       [0.39543994],
       [0.39296165],
       [0.39965005],
       [0.3925496 ],
       [0.39063265],
       [0.38331124],
       [0.38137638],
       [0.38304251],
       [0.37398927],
       [0.36851912],
       [0.37213205],
       [0.36569446],
       [0.36332963],
       [0.36299521],
       [0.36337741],
       [0.36708589],
       [0.36239803],
       [0.37292032],
       [0.36447621],
       [0.36078565],
       [0.36077371],
       [0.36080356],
       [0.35950769],
       [0.35817598],
       [0.35359561],
       [0.35861192],
       [0.36216513],
       [0.36339532],
       [0.36613637],
       [0.35203698],
       [0.35607391],
       [0.36439858],
       [0.37007178],
```

```
[0.3924421 ],
[0.40600404],
[0.39768533],
[0.39483082],
[0.38898444],
[0.3904953 ],
[0.39866471],
[0.39438891],
[0.39524884],
[0.40282107],
[0.40185962],
[0.40267178],
[0.41904045],
[0.42035425],
[0.409187  ],
[0.41961374],
[0.42089171],
[0.41810885],
[0.41368376],
[0.40136396],
[0.38868585],
[0.38400397],
[0.38888889],
[0.39006533],
[0.38414132],
[0.3864942 ],
[0.37630633],
[0.38014619],
[0.37056147],
[0.37584053],
[0.38347248],
[0.37619884],
[0.37504628],
[0.39631182],
[0.38687042],
[0.39726731],
[0.39535036],
[0.39607295],
[0.37965053],
[0.38457129],
[0.39693886],
[0.40458872],
[0.40449915],
[0.40632651],
[0.40492911],
[0.4088705 ],
[0.38974285],
[0.40532922],
[0.42675003],
[0.42414036],
[0.43033311],
[0.43415505],
[0.43668709],
[0.4436502 ],
[0.4410465 ],
[0.44560894],
[0.44222891],
[0.4385264 ],
[0.43768438],
[0.44213336],
[0.43368925],
```

```
       [0.43807255],
       [0.43846668],
       [0.43991186],
       [0.4366632 ],
       [0.43111541],
       [0.42929999],
       [0.43373703],
       [0.41918975],
       [0.4193689 ],
       [0.41923155],
       [0.42488683],
       [0.43060184],
       [0.428028  ],
       [0.42660671],
       [0.42506002],
       [0.42532278],
       [0.0893738 ],
       [0.08942157],
       [0.08305565],
       [0.07866041],
       [0.07889331],
       [0.07534009],
       [0.07634933],
       [0.08269734],
       [0.0853727 ],
       [0.08953504],
       [0.09330921],
       [0.09355405],
       [0.09452745],
       [0.08855566],
       [0.08895577],
       [0.08943352],
       [0.09446773],
       [0.09295687],
       [0.09404374],
       [0.09568598],
       [0.10219523],
       [0.102243  ],
       [0.10238633],
       [0.10400468],
       [0.1009113 ],
       [0.10162194],
       [0.10267297],
       [0.10709807],
       [0.10867462],
       [0.10758776],
       [0.10847755],
       [0.1085731 ],
       [0.10437493],
       [0.10648297],
       [0.10960622],
       [0.11499277],
       [0.11467627],
       [0.11071697],
       [0.11676042],
       [0.11625879],
       [0.10871045],
       [0.10552749],
       [0.10208177],
       [0.10371206],
       [0.10828645],
```

```
       [0.11381633],
       [0.1147121 ],
       [0.11507041],
       [0.11811005],
       [0.12051668],
       [0.1165335 ],
       [0.12350257],
       [0.12094068],
       [0.11650961],
       [0.11769799],
       [0.11672459],
       [0.11315346],
       [0.10094115],
       [0.10146667],
       [0.09562029],
       [0.07998615],
       [0.0841664 ],
       [0.08883037],
       [0.09176251],
       [0.09873756],
       [0.09046066],
       [0.07766312],
       [0.07213324],
       [0.05559736],
       [0.03983183],
       [0.03053376],
       [0.02769716],
       [0.01754512],
       [0.        ],
       [0.01725847],
       [0.00164821],
       [0.01747346],
       [0.01917542],
       [0.02291973],
       [0.01089254],
       [0.00812163],
       [0.01556845],
       [0.00524323],
       [0.00147503],
       [0.01802286],
       [0.03244473],
       [0.03379436],
       [0.02724927],
       [0.03584865],
       [0.03104136],
       [0.02841975],
       [0.03247459],
       [0.02521887],
       [0.02895123],
       [0.03625473],
       [0.0397542 ],
       [0.03910328],
       [0.03635625],
       [0.04102022],
       [0.06837699],
       [0.06028521],
       [0.06513431],
       [0.06258435],
       [0.06582703],
       [0.06090628],
       [0.06986993],
```

```
       [0.07054475],
       [0.07010881],
       [0.06508653],
       [0.05846382],
       [0.05938347],
       [0.06109738],
       [0.06636449],
       [0.070491  ],
       [0.0695176 ],
       [0.06927276],
       [0.06654962],
       [0.07246169],
       [0.08072068],
       [0.09117131],
       [0.08836457],
       [0.08897966],
       [0.09380487],
       [0.0982001 ],
       [0.09883311],
       [0.0979254 ],
       [0.10012899],
       [0.09922128],
       [0.09256871],
       [0.09645037],
       [0.10315669],
       [0.09845092],
       [0.09745363],
       [0.09430052],
       [0.09442593],
       [0.09806275],
       [0.09149976],
       [0.08322286],
       [0.08908118],
       [0.08623264],
       [0.08580864],
       [0.08411265],
       [0.09050844],
       [0.09759695],
       [0.1021594 ],
       [0.10314474],
       [0.09976471],
       [0.10275061],
       [0.10192053],
       [0.1104363 ],
       [0.11292654],
       [0.12143036],
       [0.13026861],
       [0.13007154],
       [0.13515951],
       [0.14291686],
       [0.14089242],
       [0.14221219],
       [0.15093697],
       [0.16727579],
       [0.17543923],
       [0.1707036 ],
       [0.17403585],
       [0.175911  ],
       [0.17588711],
       [0.1674848 ],
       [0.16602769],
```

```
       [0.1712112 ],
       [0.16616504],
       [0.16556189],
       [0.16543648],
       [0.17388059],
       [0.1854778 ],
       [0.18058691],
       [0.18086161],
       [0.18130352],
       [0.17677691],
       [0.1778578 ],
       [0.17569004],
       [0.17303857],
       [0.17162922],
       [0.176968  ],
       [0.17850275],
       [0.17717104],
       [0.17435833],
       [0.16842237],
       [0.17024377],
       [0.17643054],
       [0.17068568],
       [0.17259665],
       [0.18793818],
       [0.18411026],
       [0.18378779],
       [0.18525685],
       [0.22131451],
       [0.22839705],
       [0.22448553],
       [0.23568264],
       [0.23987483],
       [0.24948941],
       [0.24031077],
       [0.24753067],
       [0.22950183],
       [0.24420438],
       [0.25213491],
       [0.24634825],
       [0.23883574],
       [0.2404959 ],
       [0.24708875],
       [0.2479248 ],
       [0.24705889],
       [0.2681632 ],
       [0.2646279 ],
       [0.26990099],
       [0.28650256],
       [0.28687878],
       [0.26932172],
       [0.25267835],
       [0.25994602],
       [0.27631469],
       [0.275825  ],
       [0.27096994],
       [0.26711216],
       [0.26265124],
       [0.2587636 ],
       [0.25102416],
       [0.25085695],
       [0.25862625],
```

```
[0.24647366],
[0.24421035],
[0.24590037],
[0.25559855],
[0.25684666],
[0.26506384],
[0.24251436],
[0.24074671],
[0.2474172 ],
[0.24685585],
[0.24971037],
[0.24218591],
[0.23591554],
[0.23783248],
[0.24106322],
[0.25042101],
[0.259737  ],
[0.24846824],
[0.25166314],
[0.24739929],
[0.24656324],
[0.25454752],
[0.26548784],
[0.26726146],
[0.26348131],
[0.27027721],
[0.27558016],
[0.27284508],
[0.2714656 ],
[0.27088633],
[0.28075173],
[0.28123544],
[0.28015455],
[0.28924361],
[0.28629952],
[0.28659811],
[0.30591088],
[0.30228003],
[0.30345647],
[0.31268289],
[0.31523284],
[0.31601514],
[0.32106728],
[0.33185233],
[0.34159232],
[0.34156843],
[0.33244951],
[0.34258363],
[0.37081228],
[0.37994315],
[0.38300668],
[0.36635733],
[0.35639638],
[0.34462002],
[0.34216561],
[0.35124273],
[0.352879  ],
[0.34456627],
[0.33055645],
[0.32399345],
[0.31532242],
```

```
       [0.30737994],
       [0.299957  ],
       [0.31992667],
       [0.32954125],
       [0.32636426],
       [0.32015359],
       [0.32308574],
       [0.3303773 ],
       [0.32418455],
       [0.32959499],
       [0.3352025 ],
       [0.32490714],
       [0.32204665],
       [0.31885772],
       [0.31904284],
       [0.32174806],
       [0.30892663],
       [0.30070945],
       [0.29886417],
       [0.30851458],
       [0.29932997],
       [0.30824585],
       [0.31793209],
       [0.32659716],
       [0.3274989 ],
       [0.3256357 ],
       [0.32786317],
       [0.32931432],
       [0.34353315],
       [0.34421991],
       [0.34304346],
       [0.35586489],
       [0.3540017 ],
       [0.3299115 ],
       [0.32614927],
       [0.33104017],
       [0.3399023 ],
       [0.33695224],
       [0.32978012],
       [0.328323  ],
       [0.34230296],
       [0.34378994],
       [0.35314176],
       [0.36408208],
       [0.36785028],
       [0.37124822],
       [0.37516572],
       [0.37739319],
       [0.36007501],
       [0.34517539],
       [0.3367492 ],
       [0.35649193],
       [0.35588281],
       [0.33386482],
       [0.32887241],
       [0.32825731],
       [0.30975074],
       [0.30757701],
       [0.30672304],
       [0.29938371],
       [0.29379412]])
```

In [168…

```
x_input
```

Out[168…

```
array([[0.26548784, 0.26726146, 0.26348131, 0.27027721, 0.27558016,
        0.27284508, 0.2714656 , 0.27088633, 0.28075173, 0.28123544,
        0.28015455, 0.28924361, 0.28629952, 0.28659811, 0.30591088,
        0.30228003, 0.30345647, 0.31268289, 0.31523284, 0.31601514,
        0.32106728, 0.33185233, 0.34159232, 0.34156843, 0.33244951,
        0.34258363, 0.37081228, 0.37994315, 0.38300668, 0.36635733,
        0.35639638, 0.34462002, 0.34216561, 0.35124273, 0.352879  ,
        0.34456627, 0.33055645, 0.32399345, 0.31532242, 0.30737994,
        0.299957  , 0.31992667, 0.32954125, 0.32636426, 0.32015359,
        0.32308574, 0.3303773 , 0.32418455, 0.32959499, 0.3352025 ,
        0.32490714, 0.32204665, 0.31885772, 0.31904284, 0.32174806,
        0.30892663, 0.30070945, 0.29886417, 0.30851458, 0.29932997,
        0.30824585, 0.31793209, 0.32659716, 0.3274989 , 0.3256357 ,
        0.32786317, 0.32931432, 0.34353315, 0.34421991, 0.34304346,
        0.35586489, 0.3540017 , 0.3299115 , 0.32614927, 0.33104017,
        0.3399023 , 0.33695224, 0.32978012, 0.328323  , 0.34230296,
        0.34378994, 0.35314176, 0.36408208, 0.36785028, 0.37124822,
        0.37516572, 0.37739319, 0.36007501, 0.34517539, 0.3367492 ,
        0.35649193, 0.35588281, 0.33386482, 0.32887241, 0.32825731,
        0.30975074, 0.30757701, 0.30672304, 0.29938371, 0.29379412]])
```

In [169…

```
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
```

Function to predict future and add it as input so that next day prediction can be made based on previous day prediction

Making Prediction for next 2 months

In [170…

```
from numpy import array

lst_output=[]
n_steps=100
i=0
while(i<60):

    if(len(temp_input)>100):
        #print(temp_input)
        x_input=np.array(temp_input[1:])
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
        yhat_actual = scaler.inverse_transform(yhat)
        print("{} day output {}".format(i+1,yhat_actual[0][0]))
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        #print(temp_input)
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = model.predict(x_input, verbose=0)
        yhat_actual = scaler.inverse_transform(yhat)
        print("{} day output {}".format(i+1,yhat_actual[0][0]))

        temp_input.extend(yhat[0].tolist())
        lst_output.extend(yhat.tolist())
        i=i+1
```

```
1 day output 928.9598999023438

2 day output 924.3707885742188

3 day output 923.0628051757812

4 day output 924.1746826171875

5 day output 926.8358154296875

6 day output 930.3850708007812

7 day output 934.381591796875

8 day output 938.545166015625

9 day output 942.694091796875

10 day output 946.70166015625

11 day output 950.4720458984375

12 day output 953.9291381835938

13 day output 957.0140991210938

14 day output 959.6846923828125

15 day output 961.9151000976562
```

```
16 day output 963.6959228515625
17 day output 965.0333251953125
18 day output 965.9468994140625
19 day output 966.4677124023438
20 day output 966.635986328125
21 day output 966.4985961914062
22 day output 966.1063842773438
23 day output 965.512451171875
24 day output 964.7698364257812
25 day output 963.9293212890625
26 day output 963.0386352539062
27 day output 962.1412353515625
28 day output 961.2742919921875
29 day output 960.4700317382812
30 day output 959.7537841796875
31 day output 959.1448364257812
32 day output 958.6560668945312
33 day output 958.2954711914062
34 day output 958.0643920898438
35 day output 957.9609985351562
36 day output 957.9788818359375
37 day output 958.1083984375
38 day output 958.337890625
39 day output 958.6537475585938
40 day output 959.0413208007812
41 day output 959.4852905273438
42 day output 959.970947265625
43 day output 960.48388671875
44 day output 961.010498046875
45 day output 961.5386962890625
46 day output 962.0579833984375
47 day output 962.5594482421875
48 day output 963.0355224609375
49 day output 963.4810791015625
```

```
50 day output 963.8920288085938
51 day output 964.2664794921875
52 day output 964.6035766601562
53 day output 964.90380859375
54 day output 965.1688842773438
55 day output 965.4015502929688
56 day output 965.60498046875
57 day output 965.7831420898438
58 day output 965.9402465820312
59 day output 966.0803833007812
60 day output 966.2079467773438
```

In [ ]: