Assignment4_Insurance_NLIC

July 30, 2022

1 Stock Price Prediction of NLIC

1.1 Import the Required Libraries

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
import pandas as pd
from keras import Sequential
from keras.layers import GRU, LSTM, SimpleRNN,  Dense, Dropout
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.metrics import accuracy_score, mean_absolute_error,␣
 ↪mean_squared_error
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

```
2022-07-30 05:03:22.853369: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open
shared object file: No such file or directory
2022-07-30 05:03:22.853401: I tensorflow/stream_executor/cuda/cudart_stub.cc:29]
Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

1.2 Load Data

[3]:
```
nlic_df = pd.read_csv("data/NLIC.csv")
nlic_df.shape
```

[3]:
```
(2361, 8)
```

[4]:
```
nlic_df.head()
```

[4]:

	S.N.	Date	Total Transactions	Total Traded Shares	\
0	1	2021-12-29	617	34091.0	
1	2	2021-12-28	1009	64705.0	
2	3	2021-12-27	2842	156993.0	
3	4	2021-12-26	1765	125395.0	

```
4     5  2021-12-23                 645              38404.0

   Total Traded Amount  Max. Price  Min. Price  Close Price
0           51845988.9      1550.0      1505.0       1510.0
1           99059083.3      1570.0      1490.0       1535.0
2          249797913.1      1652.0      1530.0       1540.0
3          190751050.3      1533.4      1420.0       1533.0
4           53584721.2      1412.7      1375.0       1394.0
```

1.3 Renaming the Columns

[5]:
```
nlic_df.columns = ['SN', 'Date', 'TTrans', 'TTS','TTA', 'MaxPrice', 'MinPrice',␣
↪'ClosePrice']
```

[6]:
```
nlic_df.head()
```

[6]:
```
   SN        Date  TTrans       TTS          TTA  MaxPrice  MinPrice  \
0   1  2021-12-29     617   34091.0   51845988.9    1550.0    1505.0
1   2  2021-12-28    1009   64705.0   99059083.3    1570.0    1490.0
2   3  2021-12-27    2842  156993.0  249797913.1    1652.0    1530.0
3   4  2021-12-26    1765  125395.0  190751050.3    1533.4    1420.0
4   5  2021-12-23     645   38404.0   53584721.2    1412.7    1375.0

   ClosePrice
0      1510.0
1      1535.0
2      1540.0
3      1533.0
4      1394.0
```

[7]:
```
nlic_df.shape
```

[7]:
```
(2361, 8)
```

Converting the Date into Panda's Date Time

[9]:
```
nlic_df['Date'] = pd.to_datetime(nlic_df['Date'])
```

1.4 Sorting the Date by Date in Ascending Order

[13]:
```
nlic_df=nlic_df.sort_values(by='Date')
```

1.5 Setting Features and Target Column

[14]:
```
features = ['Date','ClosePrice']
```

[15]:
```
X = nlic_df[features]
```

[16]:
```
X.set_index("Date",inplace=True)
```

1.6 Splitting the Data Into Training, Validation and Test Set

[17]:
```
X_train_split, X_test_split = train_test_split(X, train_size=0.8,shuffle=False)
X_test_split, X_valid_split = train_test_split(X_test_split, train_size=0.
↪5,shuffle=False)
```

1.7 Fucntion to slice data to Predict next day's closing price by looking into previous 5 day's data

[18]:
```
def SliceData(data,step):
    X,Y = [],[]
    for i in range(len(data)-step):
        X.append(data[i:(i+step),])
        Y.append(data[(i+step),])
    return np.array(X),np.array(Y)
```

1.8 Normalizing the Data Using Standard Scalar

[19]:
```
std_scalar = StandardScaler()
X_train = std_scalar.fit_transform(X_train_split)
X_valid = std_scalar.fit_transform(X_valid_split)
X_test = std_scalar.fit_transform(X_test_split)
```

1.9 Getting the Sliced Data

[20]:
```
steps = 5
X_train,y_train = SliceData(X_train,steps)
X_test,y_test = SliceData(X_test,steps)
X_valid,y_valid = SliceData(X_valid,steps)
```

1.10 Building the RNN Model

[21]:
```
RNN_Model = Sequential()
RNN_Model.add(SimpleRNN(50,input_shape=(steps,1),return_sequences=True ))
RNN_Model.add(Dropout(0.5))
RNN_Model.add(SimpleRNN(50))
RNN_Model.add(Dropout(0.5))
RNN_Model.add(Dense(50))
RNN_Model.compile(optimizer='adam',loss='mean_squared_error', metrics=['mae'])
```

```
2022-07-30 05:03:47.508580: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object
file: No such file or directory
2022-07-30 05:03:47.508629: W
tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit:
UNKNOWN ERROR (303)
2022-07-30 05:03:47.508674: I
tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not
appear to be running on this host (xenon-Inspiron-3442):
/proc/driver/nvidia/version does not exist
2022-07-30 05:03:47.509017: I tensorflow/core/platform/cpu_feature_guard.cc:151]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
```

[22]:
```
RNN_Model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 5, 50)             2600

 dropout (Dropout)           (None, 5, 50)             0

 simple_rnn_1 (SimpleRNN)    (None, 50)                5050

 dropout_1 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 50)                2550

=================================================================
Total params: 10,200
Trainable params: 10,200
Non-trainable params: 0
_________________________________________________________________
```

1.11 Building LSTM Model

[23]:
```
LSTM_Model = Sequential()
LSTM_Model.add(LSTM(50,input_shape=(steps,1),return_sequences=True ))
LSTM_Model.add(Dropout(0.5))
LSTM_Model.add(LSTM(50))
LSTM_Model.add(Dropout(0.5))
```

```
LSTM_Model.add(Dense(50))
LSTM_Model.compile(optimizer='adam',loss='mean_squared_error', metrics=['mae'])
```

[24]:
```
LSTM_Model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 5, 50)             10400

 dropout_2 (Dropout)         (None, 5, 50)             0

 lstm_1 (LSTM)               (None, 50)                20200

 dropout_3 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 50)                2550

=================================================================
Total params: 33,150
Trainable params: 33,150
Non-trainable params: 0
_________________________________________________________________
```

1.12 Fitting the RNN Model

[25]:
```
RNN_History = RNN_Model.fit(X_train,y_train,epochs=100,batch_size =␣
↪50,validation_data=(X_valid,y_valid),shuffle=False,
                    verbose = 2)
```

```
Epoch 1/100
38/38 - 2s - loss: 0.7605 - mae: 0.6460 - val_loss: 0.2494 - val_mae: 0.3208 -
2s/epoch - 60ms/step
Epoch 2/100
38/38 - 0s - loss: 0.3379 - mae: 0.4257 - val_loss: 0.1658 - val_mae: 0.2495 -
231ms/epoch - 6ms/step
Epoch 3/100
38/38 - 0s - loss: 0.2462 - mae: 0.3657 - val_loss: 0.1401 - val_mae: 0.2306 -
246ms/epoch - 6ms/step
Epoch 4/100
38/38 - 0s - loss: 0.1957 - mae: 0.3229 - val_loss: 0.1192 - val_mae: 0.2128 -
195ms/epoch - 5ms/step
Epoch 5/100
38/38 - 0s - loss: 0.1719 - mae: 0.3025 - val_loss: 0.1139 - val_mae: 0.2093 -
204ms/epoch - 5ms/step
Epoch 6/100
```

```
38/38 - 0s - loss: 0.1551 - mae: 0.2873 - val_loss: 0.1032 - val_mae: 0.1999 -
208ms/epoch - 5ms/step
Epoch 7/100
38/38 - 0s - loss: 0.1380 - mae: 0.2718 - val_loss: 0.1013 - val_mae: 0.2029 -
205ms/epoch - 5ms/step
Epoch 8/100
38/38 - 0s - loss: 0.1291 - mae: 0.2594 - val_loss: 0.0920 - val_mae: 0.1923 -
255ms/epoch - 7ms/step
Epoch 9/100
38/38 - 0s - loss: 0.1158 - mae: 0.2470 - val_loss: 0.0847 - val_mae: 0.1836 -
284ms/epoch - 7ms/step
Epoch 10/100
38/38 - 0s - loss: 0.1085 - mae: 0.2398 - val_loss: 0.0861 - val_mae: 0.1909 -
340ms/epoch - 9ms/step
Epoch 11/100
38/38 - 0s - loss: 0.1001 - mae: 0.2308 - val_loss: 0.0799 - val_mae: 0.1805 -
246ms/epoch - 6ms/step
Epoch 12/100
38/38 - 0s - loss: 0.0972 - mae: 0.2270 - val_loss: 0.0738 - val_mae: 0.1756 -
264ms/epoch - 7ms/step
Epoch 13/100
38/38 - 0s - loss: 0.0984 - mae: 0.2257 - val_loss: 0.0792 - val_mae: 0.1871 -
320ms/epoch - 8ms/step
Epoch 14/100
38/38 - 0s - loss: 0.0871 - mae: 0.2179 - val_loss: 0.0853 - val_mae: 0.1930 -
343ms/epoch - 9ms/step
Epoch 15/100
38/38 - 0s - loss: 0.0883 - mae: 0.2152 - val_loss: 0.0724 - val_mae: 0.1760 -
238ms/epoch - 6ms/step
Epoch 16/100
38/38 - 0s - loss: 0.0781 - mae: 0.2033 - val_loss: 0.0743 - val_mae: 0.1792 -
243ms/epoch - 6ms/step
Epoch 17/100
38/38 - 0s - loss: 0.0831 - mae: 0.2081 - val_loss: 0.0732 - val_mae: 0.1789 -
247ms/epoch - 7ms/step
Epoch 18/100
38/38 - 0s - loss: 0.0791 - mae: 0.2035 - val_loss: 0.0708 - val_mae: 0.1745 -
235ms/epoch - 6ms/step
Epoch 19/100
38/38 - 0s - loss: 0.0744 - mae: 0.1959 - val_loss: 0.0640 - val_mae: 0.1644 -
283ms/epoch - 7ms/step
Epoch 20/100
38/38 - 0s - loss: 0.0707 - mae: 0.1903 - val_loss: 0.0733 - val_mae: 0.1808 -
206ms/epoch - 5ms/step
Epoch 21/100
38/38 - 0s - loss: 0.0694 - mae: 0.1910 - val_loss: 0.0669 - val_mae: 0.1694 -
284ms/epoch - 7ms/step
Epoch 22/100
```

```
38/38 - 0s - loss: 0.0680 - mae: 0.1853 - val_loss: 0.0651 - val_mae: 0.1668 -
208ms/epoch - 5ms/step
Epoch 23/100
38/38 - 0s - loss: 0.0674 - mae: 0.1864 - val_loss: 0.0679 - val_mae: 0.1759 -
203ms/epoch - 5ms/step
Epoch 24/100
38/38 - 0s - loss: 0.0698 - mae: 0.1885 - val_loss: 0.0744 - val_mae: 0.1848 -
263ms/epoch - 7ms/step
Epoch 25/100
38/38 - 0s - loss: 0.0701 - mae: 0.1913 - val_loss: 0.0590 - val_mae: 0.1583 -
219ms/epoch - 6ms/step
Epoch 26/100
38/38 - 0s - loss: 0.0651 - mae: 0.1804 - val_loss: 0.0632 - val_mae: 0.1643 -
201ms/epoch - 5ms/step
Epoch 27/100
38/38 - 0s - loss: 0.0633 - mae: 0.1800 - val_loss: 0.0684 - val_mae: 0.1747 -
205ms/epoch - 5ms/step
Epoch 28/100
38/38 - 0s - loss: 0.0653 - mae: 0.1818 - val_loss: 0.0574 - val_mae: 0.1545 -
200ms/epoch - 5ms/step
Epoch 29/100
38/38 - 0s - loss: 0.0686 - mae: 0.1840 - val_loss: 0.0592 - val_mae: 0.1599 -
208ms/epoch - 5ms/step
Epoch 30/100
38/38 - 0s - loss: 0.0676 - mae: 0.1845 - val_loss: 0.0630 - val_mae: 0.1694 -
197ms/epoch - 5ms/step
Epoch 31/100
38/38 - 0s - loss: 0.0697 - mae: 0.1874 - val_loss: 0.0698 - val_mae: 0.1738 -
200ms/epoch - 5ms/step
Epoch 32/100
38/38 - 0s - loss: 0.0641 - mae: 0.1815 - val_loss: 0.0571 - val_mae: 0.1532 -
217ms/epoch - 6ms/step
Epoch 33/100
38/38 - 0s - loss: 0.0559 - mae: 0.1676 - val_loss: 0.0610 - val_mae: 0.1604 -
199ms/epoch - 5ms/step
Epoch 34/100
38/38 - 0s - loss: 0.0556 - mae: 0.1676 - val_loss: 0.0534 - val_mae: 0.1477 -
193ms/epoch - 5ms/step
Epoch 35/100
38/38 - 0s - loss: 0.0505 - mae: 0.1599 - val_loss: 0.0541 - val_mae: 0.1532 -
209ms/epoch - 5ms/step
Epoch 36/100
38/38 - 0s - loss: 0.0568 - mae: 0.1672 - val_loss: 0.0566 - val_mae: 0.1537 -
199ms/epoch - 5ms/step
Epoch 37/100
38/38 - 0s - loss: 0.0597 - mae: 0.1704 - val_loss: 0.0567 - val_mae: 0.1552 -
201ms/epoch - 5ms/step
Epoch 38/100
```

```
38/38 - 0s - loss: 0.0541 - mae: 0.1643 - val_loss: 0.0650 - val_mae: 0.1684 -
230ms/epoch - 6ms/step
Epoch 39/100
38/38 - 0s - loss: 0.0546 - mae: 0.1672 - val_loss: 0.0549 - val_mae: 0.1546 -
240ms/epoch - 6ms/step
Epoch 40/100
38/38 - 0s - loss: 0.0554 - mae: 0.1685 - val_loss: 0.0589 - val_mae: 0.1596 -
201ms/epoch - 5ms/step
Epoch 41/100
38/38 - 0s - loss: 0.0551 - mae: 0.1684 - val_loss: 0.0490 - val_mae: 0.1408 -
204ms/epoch - 5ms/step
Epoch 42/100
38/38 - 0s - loss: 0.0509 - mae: 0.1560 - val_loss: 0.0519 - val_mae: 0.1434 -
202ms/epoch - 5ms/step
Epoch 43/100
38/38 - 0s - loss: 0.0512 - mae: 0.1616 - val_loss: 0.0563 - val_mae: 0.1581 -
209ms/epoch - 6ms/step
Epoch 44/100
38/38 - 0s - loss: 0.0615 - mae: 0.1751 - val_loss: 0.0647 - val_mae: 0.1690 -
212ms/epoch - 6ms/step
Epoch 45/100
38/38 - 0s - loss: 0.0585 - mae: 0.1705 - val_loss: 0.0571 - val_mae: 0.1559 -
198ms/epoch - 5ms/step
Epoch 46/100
38/38 - 0s - loss: 0.0564 - mae: 0.1696 - val_loss: 0.0516 - val_mae: 0.1482 -
199ms/epoch - 5ms/step
Epoch 47/100
38/38 - 0s - loss: 0.0570 - mae: 0.1705 - val_loss: 0.0542 - val_mae: 0.1469 -
203ms/epoch - 5ms/step
Epoch 48/100
38/38 - 0s - loss: 0.0581 - mae: 0.1703 - val_loss: 0.0501 - val_mae: 0.1421 -
235ms/epoch - 6ms/step
Epoch 49/100
38/38 - 0s - loss: 0.0486 - mae: 0.1554 - val_loss: 0.0483 - val_mae: 0.1400 -
214ms/epoch - 6ms/step
Epoch 50/100
38/38 - 0s - loss: 0.0503 - mae: 0.1559 - val_loss: 0.0537 - val_mae: 0.1481 -
237ms/epoch - 6ms/step
Epoch 51/100
38/38 - 0s - loss: 0.0506 - mae: 0.1559 - val_loss: 0.0527 - val_mae: 0.1484 -
204ms/epoch - 5ms/step
Epoch 52/100
38/38 - 0s - loss: 0.0551 - mae: 0.1627 - val_loss: 0.0604 - val_mae: 0.1610 -
240ms/epoch - 6ms/step
Epoch 53/100
38/38 - 0s - loss: 0.0561 - mae: 0.1661 - val_loss: 0.0585 - val_mae: 0.1607 -
205ms/epoch - 5ms/step
Epoch 54/100
```

```
38/38 - 0s - loss: 0.0479 - mae: 0.1527 - val_loss: 0.0509 - val_mae: 0.1438 -
235ms/epoch - 6ms/step
Epoch 55/100
38/38 - 0s - loss: 0.0516 - mae: 0.1560 - val_loss: 0.0449 - val_mae: 0.1300 -
364ms/epoch - 10ms/step
Epoch 56/100
38/38 - 0s - loss: 0.0490 - mae: 0.1518 - val_loss: 0.0409 - val_mae: 0.1218 -
276ms/epoch - 7ms/step
Epoch 57/100
38/38 - 0s - loss: 0.0483 - mae: 0.1566 - val_loss: 0.0531 - val_mae: 0.1469 -
202ms/epoch - 5ms/step
Epoch 58/100
38/38 - 0s - loss: 0.0510 - mae: 0.1615 - val_loss: 0.0584 - val_mae: 0.1548 -
269ms/epoch - 7ms/step
Epoch 59/100
38/38 - 0s - loss: 0.0484 - mae: 0.1566 - val_loss: 0.0540 - val_mae: 0.1528 -
235ms/epoch - 6ms/step
Epoch 60/100
38/38 - 0s - loss: 0.0505 - mae: 0.1578 - val_loss: 0.0557 - val_mae: 0.1519 -
182ms/epoch - 5ms/step
Epoch 61/100
38/38 - 0s - loss: 0.0500 - mae: 0.1539 - val_loss: 0.0614 - val_mae: 0.1663 -
242ms/epoch - 6ms/step
Epoch 62/100
38/38 - 0s - loss: 0.0569 - mae: 0.1717 - val_loss: 0.0514 - val_mae: 0.1492 -
286ms/epoch - 8ms/step
Epoch 63/100
38/38 - 0s - loss: 0.0578 - mae: 0.1664 - val_loss: 0.0490 - val_mae: 0.1367 -
218ms/epoch - 6ms/step
Epoch 64/100
38/38 - 0s - loss: 0.0485 - mae: 0.1540 - val_loss: 0.0454 - val_mae: 0.1304 -
261ms/epoch - 7ms/step
Epoch 65/100
38/38 - 0s - loss: 0.0484 - mae: 0.1544 - val_loss: 0.0570 - val_mae: 0.1558 -
180ms/epoch - 5ms/step
Epoch 66/100
38/38 - 0s - loss: 0.0520 - mae: 0.1607 - val_loss: 0.0507 - val_mae: 0.1453 -
182ms/epoch - 5ms/step
Epoch 67/100
38/38 - 0s - loss: 0.0487 - mae: 0.1534 - val_loss: 0.0558 - val_mae: 0.1548 -
177ms/epoch - 5ms/step
Epoch 68/100
38/38 - 0s - loss: 0.0486 - mae: 0.1550 - val_loss: 0.0521 - val_mae: 0.1470 -
245ms/epoch - 6ms/step
Epoch 69/100
38/38 - 0s - loss: 0.0512 - mae: 0.1563 - val_loss: 0.0500 - val_mae: 0.1420 -
277ms/epoch - 7ms/step
Epoch 70/100
```

```
38/38 - 0s - loss: 0.0489 - mae: 0.1501 - val_loss: 0.0524 - val_mae: 0.1442 -
285ms/epoch - 8ms/step
Epoch 71/100
38/38 - 0s - loss: 0.0425 - mae: 0.1422 - val_loss: 0.0454 - val_mae: 0.1308 -
296ms/epoch - 8ms/step
Epoch 72/100
38/38 - 0s - loss: 0.0441 - mae: 0.1455 - val_loss: 0.0533 - val_mae: 0.1497 -
223ms/epoch - 6ms/step
Epoch 73/100
38/38 - 0s - loss: 0.0559 - mae: 0.1630 - val_loss: 0.0588 - val_mae: 0.1573 -
273ms/epoch - 7ms/step
Epoch 74/100
38/38 - 0s - loss: 0.0483 - mae: 0.1542 - val_loss: 0.0494 - val_mae: 0.1423 -
195ms/epoch - 5ms/step
Epoch 75/100
38/38 - 0s - loss: 0.0524 - mae: 0.1617 - val_loss: 0.0549 - val_mae: 0.1483 -
178ms/epoch - 5ms/step
Epoch 76/100
38/38 - 0s - loss: 0.0486 - mae: 0.1555 - val_loss: 0.0488 - val_mae: 0.1393 -
206ms/epoch - 5ms/step
Epoch 77/100
38/38 - 0s - loss: 0.0511 - mae: 0.1597 - val_loss: 0.0527 - val_mae: 0.1458 -
178ms/epoch - 5ms/step
Epoch 78/100
38/38 - 0s - loss: 0.0541 - mae: 0.1638 - val_loss: 0.0454 - val_mae: 0.1345 -
186ms/epoch - 5ms/step
Epoch 79/100
38/38 - 0s - loss: 0.0508 - mae: 0.1552 - val_loss: 0.0516 - val_mae: 0.1464 -
207ms/epoch - 5ms/step
Epoch 80/100
38/38 - 0s - loss: 0.0507 - mae: 0.1537 - val_loss: 0.0558 - val_mae: 0.1493 -
292ms/epoch - 8ms/step
Epoch 81/100
38/38 - 0s - loss: 0.0484 - mae: 0.1536 - val_loss: 0.0465 - val_mae: 0.1353 -
309ms/epoch - 8ms/step
Epoch 82/100
38/38 - 0s - loss: 0.0491 - mae: 0.1529 - val_loss: 0.0508 - val_mae: 0.1407 -
244ms/epoch - 6ms/step
Epoch 83/100
38/38 - 0s - loss: 0.0498 - mae: 0.1564 - val_loss: 0.0485 - val_mae: 0.1412 -
179ms/epoch - 5ms/step
Epoch 84/100
38/38 - 0s - loss: 0.0531 - mae: 0.1600 - val_loss: 0.0597 - val_mae: 0.1622 -
175ms/epoch - 5ms/step
Epoch 85/100
38/38 - 0s - loss: 0.0556 - mae: 0.1671 - val_loss: 0.0565 - val_mae: 0.1567 -
254ms/epoch - 7ms/step
Epoch 86/100
```

```
38/38 - 0s - loss: 0.0463 - mae: 0.1526 - val_loss: 0.0553 - val_mae: 0.1546 -
252ms/epoch - 7ms/step
Epoch 87/100
38/38 - 0s - loss: 0.0446 - mae: 0.1490 - val_loss: 0.0524 - val_mae: 0.1476 -
180ms/epoch - 5ms/step
Epoch 88/100
38/38 - 0s - loss: 0.0445 - mae: 0.1445 - val_loss: 0.0488 - val_mae: 0.1420 -
181ms/epoch - 5ms/step
Epoch 89/100
38/38 - 0s - loss: 0.0440 - mae: 0.1426 - val_loss: 0.0502 - val_mae: 0.1406 -
189ms/epoch - 5ms/step
Epoch 90/100
38/38 - 0s - loss: 0.0491 - mae: 0.1553 - val_loss: 0.0482 - val_mae: 0.1391 -
248ms/epoch - 7ms/step
Epoch 91/100
38/38 - 0s - loss: 0.0494 - mae: 0.1539 - val_loss: 0.0546 - val_mae: 0.1517 -
201ms/epoch - 5ms/step
Epoch 92/100
38/38 - 0s - loss: 0.0502 - mae: 0.1532 - val_loss: 0.0546 - val_mae: 0.1502 -
180ms/epoch - 5ms/step
Epoch 93/100
38/38 - 0s - loss: 0.0454 - mae: 0.1509 - val_loss: 0.0437 - val_mae: 0.1306 -
185ms/epoch - 5ms/step
Epoch 94/100
38/38 - 0s - loss: 0.0451 - mae: 0.1445 - val_loss: 0.0457 - val_mae: 0.1314 -
211ms/epoch - 6ms/step
Epoch 95/100
38/38 - 0s - loss: 0.0483 - mae: 0.1530 - val_loss: 0.0458 - val_mae: 0.1341 -
322ms/epoch - 8ms/step
Epoch 96/100
38/38 - 0s - loss: 0.0515 - mae: 0.1571 - val_loss: 0.0562 - val_mae: 0.1519 -
222ms/epoch - 6ms/step
Epoch 97/100
38/38 - 0s - loss: 0.0499 - mae: 0.1563 - val_loss: 0.0452 - val_mae: 0.1338 -
181ms/epoch - 5ms/step
Epoch 98/100
38/38 - 0s - loss: 0.0505 - mae: 0.1561 - val_loss: 0.0533 - val_mae: 0.1439 -
237ms/epoch - 6ms/step
Epoch 99/100
38/38 - 0s - loss: 0.0461 - mae: 0.1514 - val_loss: 0.0444 - val_mae: 0.1282 -
213ms/epoch - 6ms/step
Epoch 100/100
38/38 - 0s - loss: 0.0401 - mae: 0.1368 - val_loss: 0.0442 - val_mae: 0.1277 -
178ms/epoch - 5ms/step
```

[26]:
```
LSTM_History = LSTM_Model.fit(X_train,y_train,epochs=100,batch_size =␣
↪50,validation_data=(X_valid,y_valid),shuffle=False,
```

```
                    verbose = 2)
```

```
Epoch 1/100
38/38 - 6s - loss: 0.8425 - mae: 0.7812 - val_loss: 0.4477 - val_mae: 0.4636 -
6s/epoch - 158ms/step
Epoch 2/100
38/38 - 0s - loss: 0.2807 - mae: 0.3981 - val_loss: 0.1832 - val_mae: 0.2948 -
345ms/epoch - 9ms/step
Epoch 3/100
38/38 - 0s - loss: 0.1445 - mae: 0.2747 - val_loss: 0.1790 - val_mae: 0.2597 -
355ms/epoch - 9ms/step
Epoch 4/100
38/38 - 0s - loss: 0.1368 - mae: 0.2726 - val_loss: 0.1420 - val_mae: 0.2368 -
482ms/epoch - 13ms/step
Epoch 5/100
38/38 - 0s - loss: 0.1077 - mae: 0.2373 - val_loss: 0.1477 - val_mae: 0.2384 -
362ms/epoch - 10ms/step
Epoch 6/100
38/38 - 0s - loss: 0.1120 - mae: 0.2427 - val_loss: 0.1342 - val_mae: 0.2281 -
399ms/epoch - 10ms/step
Epoch 7/100
38/38 - 1s - loss: 0.0976 - mae: 0.2258 - val_loss: 0.1216 - val_mae: 0.2178 -
552ms/epoch - 15ms/step
Epoch 8/100
38/38 - 0s - loss: 0.0918 - mae: 0.2159 - val_loss: 0.1245 - val_mae: 0.2228 -
485ms/epoch - 13ms/step
Epoch 9/100
38/38 - 0s - loss: 0.0939 - mae: 0.2209 - val_loss: 0.1222 - val_mae: 0.2217 -
341ms/epoch - 9ms/step
Epoch 10/100
38/38 - 0s - loss: 0.0910 - mae: 0.2174 - val_loss: 0.1148 - val_mae: 0.2156 -
346ms/epoch - 9ms/step
Epoch 11/100
38/38 - 0s - loss: 0.0843 - mae: 0.2048 - val_loss: 0.1089 - val_mae: 0.2111 -
345ms/epoch - 9ms/step
Epoch 12/100
38/38 - 0s - loss: 0.0768 - mae: 0.1948 - val_loss: 0.1007 - val_mae: 0.2022 -
347ms/epoch - 9ms/step
Epoch 13/100
38/38 - 0s - loss: 0.0715 - mae: 0.1877 - val_loss: 0.0987 - val_mae: 0.2004 -
404ms/epoch - 11ms/step
Epoch 14/100
38/38 - 0s - loss: 0.0723 - mae: 0.1901 - val_loss: 0.0980 - val_mae: 0.2014 -
380ms/epoch - 10ms/step
Epoch 15/100
38/38 - 0s - loss: 0.0668 - mae: 0.1824 - val_loss: 0.0919 - val_mae: 0.1948 -
335ms/epoch - 9ms/step
```

```
Epoch 16/100
38/38 - 0s - loss: 0.0651 - mae: 0.1778 - val_loss: 0.0924 - val_mae: 0.1938 -
364ms/epoch - 10ms/step
Epoch 17/100
38/38 - 0s - loss: 0.0700 - mae: 0.1869 - val_loss: 0.0938 - val_mae: 0.1986 -
336ms/epoch - 9ms/step
Epoch 18/100
38/38 - 0s - loss: 0.0649 - mae: 0.1801 - val_loss: 0.0910 - val_mae: 0.1951 -
480ms/epoch - 13ms/step
Epoch 19/100
38/38 - 0s - loss: 0.0685 - mae: 0.1829 - val_loss: 0.0851 - val_mae: 0.1875 -
385ms/epoch - 10ms/step
Epoch 20/100
38/38 - 0s - loss: 0.0606 - mae: 0.1731 - val_loss: 0.0813 - val_mae: 0.1826 -
378ms/epoch - 10ms/step
Epoch 21/100
38/38 - 0s - loss: 0.0577 - mae: 0.1692 - val_loss: 0.0807 - val_mae: 0.1812 -
393ms/epoch - 10ms/step
Epoch 22/100
38/38 - 0s - loss: 0.0600 - mae: 0.1704 - val_loss: 0.0784 - val_mae: 0.1785 -
410ms/epoch - 11ms/step
Epoch 23/100
38/38 - 0s - loss: 0.0558 - mae: 0.1639 - val_loss: 0.0765 - val_mae: 0.1764 -
411ms/epoch - 11ms/step
Epoch 24/100
38/38 - 0s - loss: 0.0499 - mae: 0.1582 - val_loss: 0.0725 - val_mae: 0.1689 -
371ms/epoch - 10ms/step
Epoch 25/100
38/38 - 0s - loss: 0.0540 - mae: 0.1635 - val_loss: 0.0730 - val_mae: 0.1699 -
392ms/epoch - 10ms/step
Epoch 26/100
38/38 - 0s - loss: 0.0497 - mae: 0.1576 - val_loss: 0.0727 - val_mae: 0.1687 -
425ms/epoch - 11ms/step
Epoch 27/100
38/38 - 0s - loss: 0.0526 - mae: 0.1632 - val_loss: 0.0718 - val_mae: 0.1711 -
398ms/epoch - 10ms/step
Epoch 28/100
38/38 - 0s - loss: 0.0494 - mae: 0.1589 - val_loss: 0.0737 - val_mae: 0.1760 -
410ms/epoch - 11ms/step
Epoch 29/100
38/38 - 0s - loss: 0.0487 - mae: 0.1570 - val_loss: 0.0719 - val_mae: 0.1788 -
376ms/epoch - 10ms/step
Epoch 30/100
38/38 - 0s - loss: 0.0489 - mae: 0.1592 - val_loss: 0.0704 - val_mae: 0.1718 -
358ms/epoch - 9ms/step
Epoch 31/100
38/38 - 0s - loss: 0.0534 - mae: 0.1648 - val_loss: 0.0758 - val_mae: 0.1815 -
362ms/epoch - 10ms/step
```

```
Epoch 32/100
38/38 - 0s - loss: 0.0497 - mae: 0.1595 - val_loss: 0.0771 - val_mae: 0.1904 -
459ms/epoch - 12ms/step
Epoch 33/100
38/38 - 0s - loss: 0.0503 - mae: 0.1584 - val_loss: 0.0748 - val_mae: 0.1914 -
356ms/epoch - 9ms/step
Epoch 34/100
38/38 - 0s - loss: 0.0506 - mae: 0.1588 - val_loss: 0.0699 - val_mae: 0.1822 -
372ms/epoch - 10ms/step
Epoch 35/100
38/38 - 0s - loss: 0.0478 - mae: 0.1557 - val_loss: 0.0715 - val_mae: 0.1787 -
431ms/epoch - 11ms/step
Epoch 36/100
38/38 - 1s - loss: 0.0505 - mae: 0.1653 - val_loss: 0.0675 - val_mae: 0.1727 -
536ms/epoch - 14ms/step
Epoch 37/100
38/38 - 0s - loss: 0.0467 - mae: 0.1529 - val_loss: 0.0692 - val_mae: 0.1732 -
398ms/epoch - 10ms/step
Epoch 38/100
38/38 - 0s - loss: 0.0459 - mae: 0.1518 - val_loss: 0.0671 - val_mae: 0.1736 -
335ms/epoch - 9ms/step
Epoch 39/100
38/38 - 0s - loss: 0.0438 - mae: 0.1477 - val_loss: 0.0685 - val_mae: 0.1693 -
337ms/epoch - 9ms/step
Epoch 40/100
38/38 - 0s - loss: 0.0483 - mae: 0.1540 - val_loss: 0.0761 - val_mae: 0.1824 -
323ms/epoch - 8ms/step
Epoch 41/100
38/38 - 0s - loss: 0.0496 - mae: 0.1568 - val_loss: 0.0768 - val_mae: 0.1834 -
325ms/epoch - 9ms/step
Epoch 42/100
38/38 - 0s - loss: 0.0532 - mae: 0.1641 - val_loss: 0.0826 - val_mae: 0.1922 -
324ms/epoch - 9ms/step
Epoch 43/100
38/38 - 0s - loss: 0.0591 - mae: 0.1708 - val_loss: 0.0870 - val_mae: 0.1964 -
328ms/epoch - 9ms/step
Epoch 44/100
38/38 - 0s - loss: 0.0628 - mae: 0.1754 - val_loss: 0.0858 - val_mae: 0.1961 -
317ms/epoch - 8ms/step
Epoch 45/100
38/38 - 0s - loss: 0.0594 - mae: 0.1713 - val_loss: 0.0783 - val_mae: 0.1842 -
441ms/epoch - 12ms/step
Epoch 46/100
38/38 - 0s - loss: 0.0569 - mae: 0.1654 - val_loss: 0.0769 - val_mae: 0.1839 -
380ms/epoch - 10ms/step
Epoch 47/100
38/38 - 0s - loss: 0.0501 - mae: 0.1581 - val_loss: 0.0772 - val_mae: 0.1839 -
389ms/epoch - 10ms/step
```

```
Epoch 48/100
38/38 - 1s - loss: 0.0537 - mae: 0.1609 - val_loss: 0.0776 - val_mae: 0.1831 -
586ms/epoch - 15ms/step
Epoch 49/100
38/38 - 0s - loss: 0.0514 - mae: 0.1585 - val_loss: 0.0698 - val_mae: 0.1731 -
368ms/epoch - 10ms/step
Epoch 50/100
38/38 - 0s - loss: 0.0508 - mae: 0.1536 - val_loss: 0.0787 - val_mae: 0.1849 -
350ms/epoch - 9ms/step
Epoch 51/100
38/38 - 0s - loss: 0.0587 - mae: 0.1730 - val_loss: 0.0753 - val_mae: 0.1814 -
336ms/epoch - 9ms/step
Epoch 52/100
38/38 - 0s - loss: 0.0573 - mae: 0.1663 - val_loss: 0.0744 - val_mae: 0.1861 -
325ms/epoch - 9ms/step
Epoch 53/100
38/38 - 0s - loss: 0.0493 - mae: 0.1565 - val_loss: 0.0715 - val_mae: 0.1851 -
331ms/epoch - 9ms/step
Epoch 54/100
38/38 - 0s - loss: 0.0494 - mae: 0.1549 - val_loss: 0.0724 - val_mae: 0.1835 -
329ms/epoch - 9ms/step
Epoch 55/100
38/38 - 0s - loss: 0.0496 - mae: 0.1549 - val_loss: 0.0749 - val_mae: 0.1840 -
325ms/epoch - 9ms/step
Epoch 56/100
38/38 - 0s - loss: 0.0539 - mae: 0.1682 - val_loss: 0.0722 - val_mae: 0.1823 -
326ms/epoch - 9ms/step
Epoch 57/100
38/38 - 0s - loss: 0.0491 - mae: 0.1557 - val_loss: 0.0659 - val_mae: 0.1695 -
320ms/epoch - 8ms/step
Epoch 58/100
38/38 - 0s - loss: 0.0475 - mae: 0.1498 - val_loss: 0.0653 - val_mae: 0.1680 -
328ms/epoch - 9ms/step
Epoch 59/100
38/38 - 0s - loss: 0.0471 - mae: 0.1499 - val_loss: 0.0615 - val_mae: 0.1646 -
322ms/epoch - 8ms/step
Epoch 60/100
38/38 - 0s - loss: 0.0400 - mae: 0.1395 - val_loss: 0.0606 - val_mae: 0.1618 -
337ms/epoch - 9ms/step
Epoch 61/100
38/38 - 0s - loss: 0.0428 - mae: 0.1433 - val_loss: 0.0583 - val_mae: 0.1554 -
332ms/epoch - 9ms/step
Epoch 62/100
38/38 - 0s - loss: 0.0438 - mae: 0.1457 - val_loss: 0.0586 - val_mae: 0.1571 -
316ms/epoch - 8ms/step
Epoch 63/100
38/38 - 0s - loss: 0.0419 - mae: 0.1405 - val_loss: 0.0581 - val_mae: 0.1566 -
322ms/epoch - 8ms/step
```

```
Epoch 64/100
38/38 - 0s - loss: 0.0396 - mae: 0.1378 - val_loss: 0.0558 - val_mae: 0.1499 -
320ms/epoch - 8ms/step
Epoch 65/100
38/38 - 0s - loss: 0.0420 - mae: 0.1395 - val_loss: 0.0563 - val_mae: 0.1515 -
338ms/epoch - 9ms/step
Epoch 66/100
38/38 - 0s - loss: 0.0385 - mae: 0.1335 - val_loss: 0.0557 - val_mae: 0.1497 -
338ms/epoch - 9ms/step
Epoch 67/100
38/38 - 0s - loss: 0.0411 - mae: 0.1387 - val_loss: 0.0578 - val_mae: 0.1540 -
342ms/epoch - 9ms/step
Epoch 68/100
38/38 - 0s - loss: 0.0423 - mae: 0.1424 - val_loss: 0.0556 - val_mae: 0.1511 -
333ms/epoch - 9ms/step
Epoch 69/100
38/38 - 0s - loss: 0.0432 - mae: 0.1408 - val_loss: 0.0542 - val_mae: 0.1494 -
325ms/epoch - 9ms/step
Epoch 70/100
38/38 - 0s - loss: 0.0397 - mae: 0.1384 - val_loss: 0.0548 - val_mae: 0.1494 -
325ms/epoch - 9ms/step
Epoch 71/100
38/38 - 0s - loss: 0.0428 - mae: 0.1399 - val_loss: 0.0522 - val_mae: 0.1444 -
350ms/epoch - 9ms/step
Epoch 72/100
38/38 - 0s - loss: 0.0395 - mae: 0.1359 - val_loss: 0.0502 - val_mae: 0.1424 -
324ms/epoch - 9ms/step
Epoch 73/100
38/38 - 0s - loss: 0.0380 - mae: 0.1316 - val_loss: 0.0511 - val_mae: 0.1442 -
337ms/epoch - 9ms/step
Epoch 74/100
38/38 - 1s - loss: 0.0387 - mae: 0.1342 - val_loss: 0.0499 - val_mae: 0.1405 -
512ms/epoch - 13ms/step
Epoch 75/100
38/38 - 0s - loss: 0.0365 - mae: 0.1322 - val_loss: 0.0496 - val_mae: 0.1420 -
321ms/epoch - 8ms/step
Epoch 76/100
38/38 - 0s - loss: 0.0365 - mae: 0.1298 - val_loss: 0.0488 - val_mae: 0.1396 -
332ms/epoch - 9ms/step
Epoch 77/100
38/38 - 0s - loss: 0.0401 - mae: 0.1371 - val_loss: 0.0491 - val_mae: 0.1391 -
404ms/epoch - 11ms/step
Epoch 78/100
38/38 - 0s - loss: 0.0385 - mae: 0.1369 - val_loss: 0.0498 - val_mae: 0.1405 -
392ms/epoch - 10ms/step
Epoch 79/100
38/38 - 0s - loss: 0.0419 - mae: 0.1443 - val_loss: 0.0481 - val_mae: 0.1373 -
469ms/epoch - 12ms/step
```

```
Epoch 80/100
38/38 - 0s - loss: 0.0371 - mae: 0.1344 - val_loss: 0.0472 - val_mae: 0.1453 -
376ms/epoch - 10ms/step
Epoch 81/100
38/38 - 0s - loss: 0.0390 - mae: 0.1371 - val_loss: 0.0447 - val_mae: 0.1428 -
322ms/epoch - 8ms/step
Epoch 82/100
38/38 - 0s - loss: 0.0420 - mae: 0.1446 - val_loss: 0.0467 - val_mae: 0.1437 -
366ms/epoch - 10ms/step
Epoch 83/100
38/38 - 0s - loss: 0.0404 - mae: 0.1472 - val_loss: 0.0555 - val_mae: 0.1625 -
381ms/epoch - 10ms/step
Epoch 84/100
38/38 - 0s - loss: 0.0440 - mae: 0.1529 - val_loss: 0.0536 - val_mae: 0.1693 -
318ms/epoch - 8ms/step
Epoch 85/100
38/38 - 0s - loss: 0.0431 - mae: 0.1482 - val_loss: 0.0464 - val_mae: 0.1518 -
317ms/epoch - 8ms/step
Epoch 86/100
38/38 - 0s - loss: 0.0395 - mae: 0.1447 - val_loss: 0.0482 - val_mae: 0.1542 -
317ms/epoch - 8ms/step
Epoch 87/100
38/38 - 0s - loss: 0.0373 - mae: 0.1387 - val_loss: 0.0483 - val_mae: 0.1530 -
315ms/epoch - 8ms/step
Epoch 88/100
38/38 - 0s - loss: 0.0407 - mae: 0.1418 - val_loss: 0.0480 - val_mae: 0.1524 -
315ms/epoch - 8ms/step
Epoch 89/100
38/38 - 0s - loss: 0.0399 - mae: 0.1400 - val_loss: 0.0466 - val_mae: 0.1498 -
322ms/epoch - 8ms/step
Epoch 90/100
38/38 - 0s - loss: 0.0425 - mae: 0.1460 - val_loss: 0.0526 - val_mae: 0.1542 -
347ms/epoch - 9ms/step
Epoch 91/100
38/38 - 0s - loss: 0.0421 - mae: 0.1461 - val_loss: 0.0498 - val_mae: 0.1508 -
346ms/epoch - 9ms/step
Epoch 92/100
38/38 - 0s - loss: 0.0423 - mae: 0.1466 - val_loss: 0.0481 - val_mae: 0.1468 -
345ms/epoch - 9ms/step
Epoch 93/100
38/38 - 0s - loss: 0.0409 - mae: 0.1438 - val_loss: 0.0483 - val_mae: 0.1483 -
336ms/epoch - 9ms/step
Epoch 94/100
38/38 - 0s - loss: 0.0389 - mae: 0.1425 - val_loss: 0.0461 - val_mae: 0.1443 -
318ms/epoch - 8ms/step
Epoch 95/100
38/38 - 0s - loss: 0.0401 - mae: 0.1406 - val_loss: 0.0477 - val_mae: 0.1502 -
320ms/epoch - 8ms/step
```

```
Epoch 96/100
38/38 - 0s - loss: 0.0395 - mae: 0.1389 - val_loss: 0.0472 - val_mae: 0.1541 -
346ms/epoch - 9ms/step
Epoch 97/100
38/38 - 0s - loss: 0.0384 - mae: 0.1355 - val_loss: 0.0470 - val_mae: 0.1554 -
358ms/epoch - 9ms/step
Epoch 98/100
38/38 - 0s - loss: 0.0391 - mae: 0.1361 - val_loss: 0.0451 - val_mae: 0.1492 -
354ms/epoch - 9ms/step
Epoch 99/100
38/38 - 0s - loss: 0.0382 - mae: 0.1395 - val_loss: 0.0463 - val_mae: 0.1516 -
348ms/epoch - 9ms/step
Epoch 100/100
38/38 - 0s - loss: 0.0393 - mae: 0.1398 - val_loss: 0.0460 - val_mae: 0.1523 -
337ms/epoch - 9ms/step
```

1.13 Make Predictions

[27]:
```
RNN_Predictions = RNN_Model.predict(X_test)
LSTM_predictions = LSTM_Model.predict(X_test)
```

1.14 Inverse Transform the Values

[28]:
```
RNN_act_prd = std_scalar.inverse_transform(RNN_Predictions)
LSTM_act_prd = std_scalar.inverse_transform(LSTM_predictions)
```

1.15 Evalation Metrics (RMSE and MAE)

[29]:
```
print("### RNN Model ###")
Y_test_res_RNN = std_scalar.inverse_transform(y_test)
pre_RNN = RNN_act_prd[:,:1]

rmse=np.sqrt(np.mean(((pre_RNN- Y_test_res_RNN)**2)))
print(f"RMSE {rmse}"  )

print(f"MAE {mean_absolute_error(Y_test_res_RNN, pre_RNN)}")
```

```
### RNN Model ###
RMSE 76.88429113784437
MAE 46.06268971100514
```

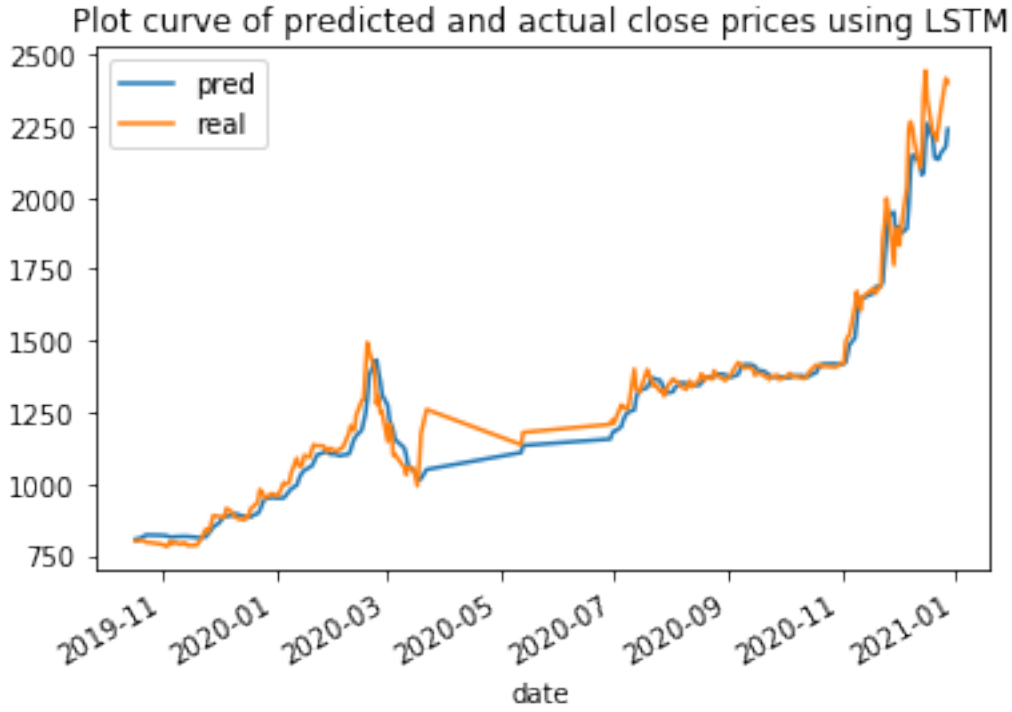
[30]:
```
print("### LSTM Model ###")
Y_test_res_LSTM = std_scalar.inverse_transform(y_test)
pre_LSTM = LSTM_act_prd[:,:1]

rmse=np.sqrt(np.mean(((pre_LSTM- Y_test_res_LSTM)**2)))
print(f"RMSE {rmse}"  )
```

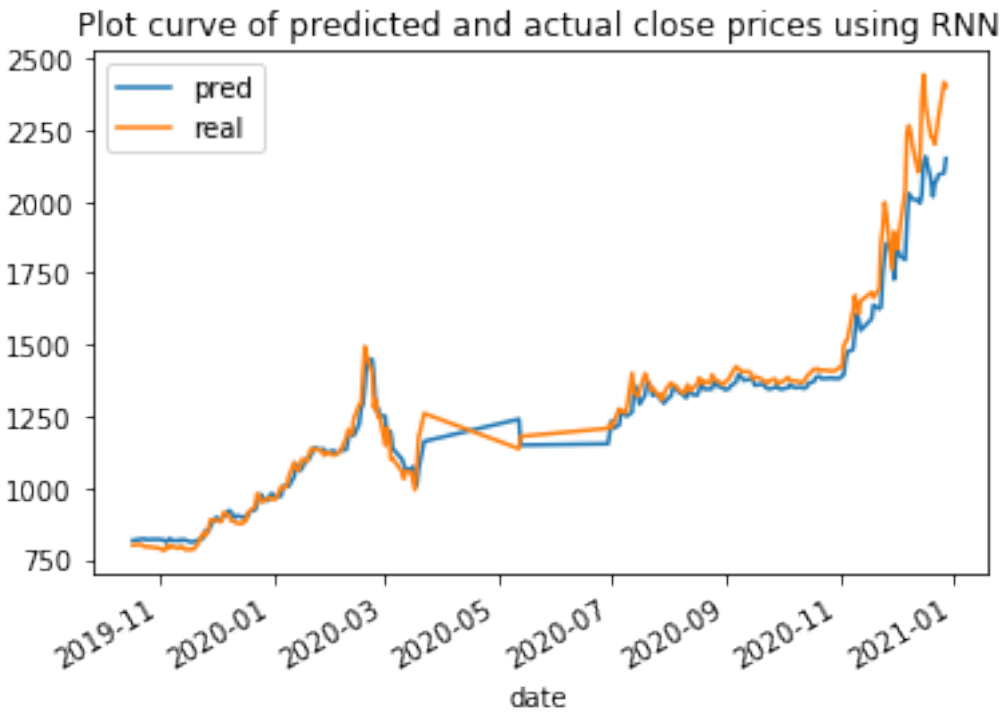
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

```
### LSTM Model ###
RMSE 63.479235264203794
MAE 40.0531029639306
```

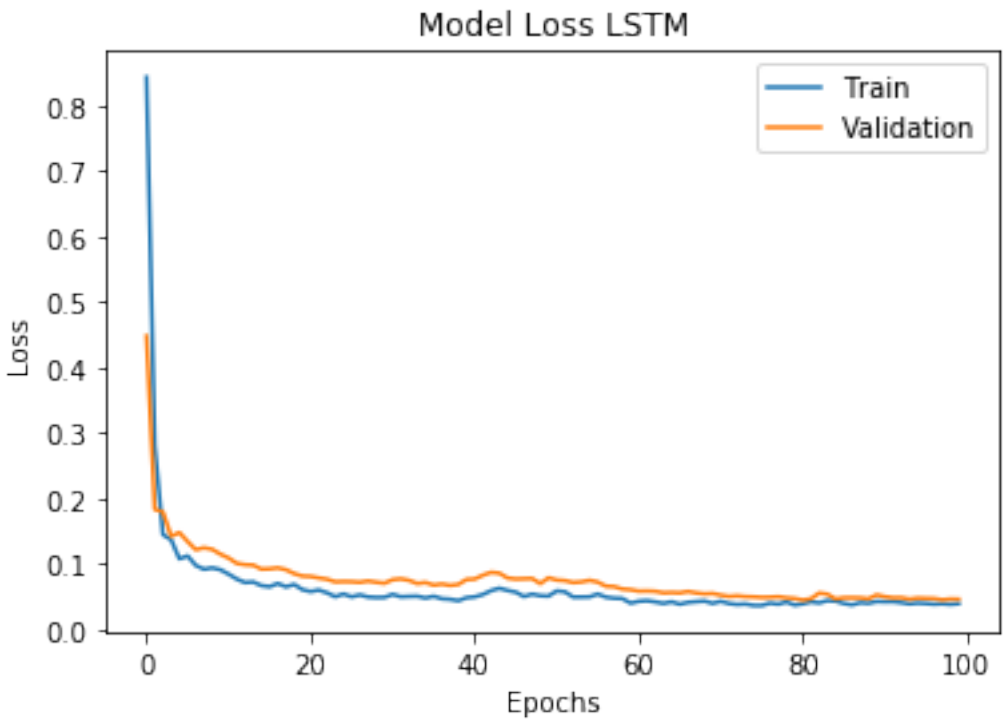
[31]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



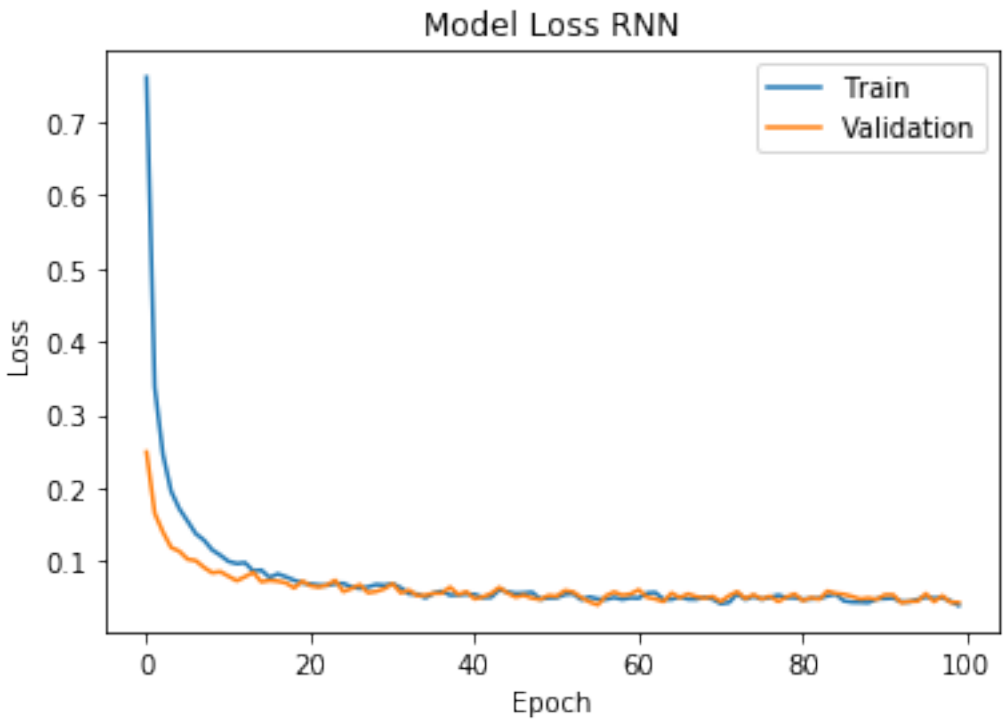
[32]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



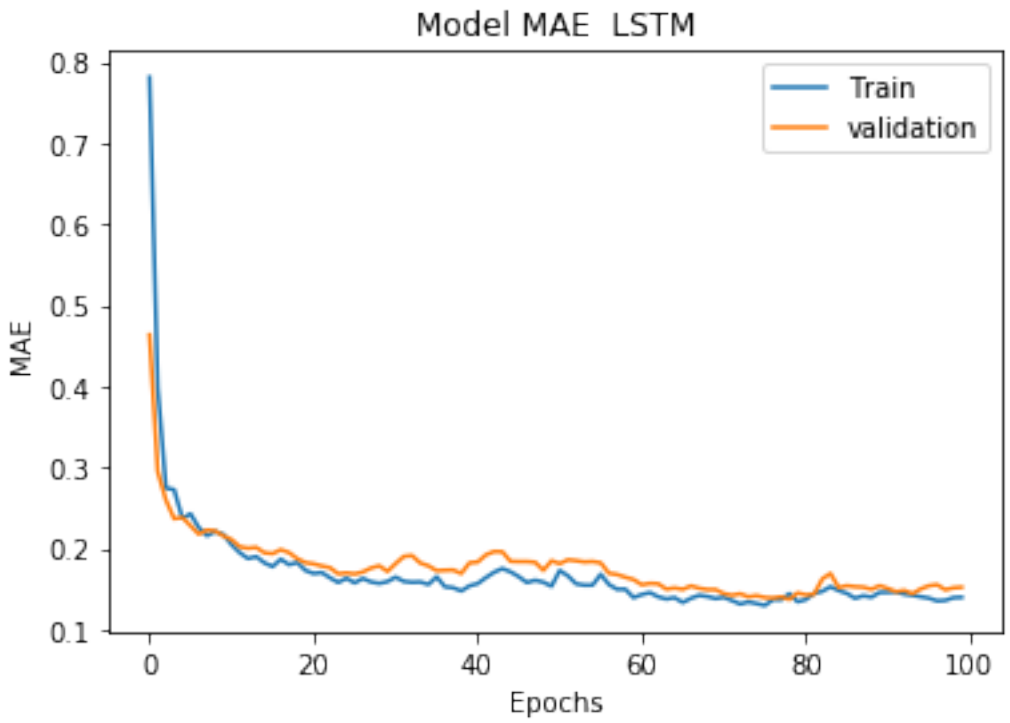
[33]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



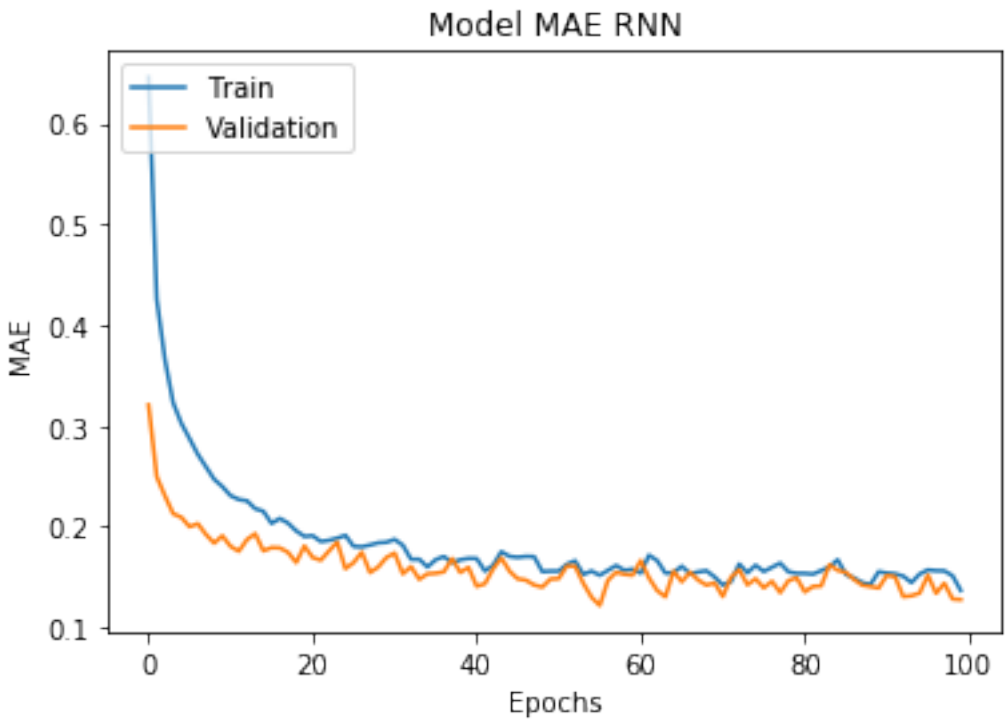
[34]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[35]:
```
plt.plot(LSTM_History.history['mae'])
plt.plot(LSTM_History.history['val_mae'])
plt.title('Model MAE  LSTM')
plt.ylabel('MAE')
plt.xlabel('Epochs')
plt.legend(['Train', 'validation'], loc='upper right')
plt.show()
```



[36]:
```
plt.plot(RNN_History.history['mae'])
plt.plot(RNN_History.history['val_mae'])
plt.title('Model MAE RNN')
plt.ylabel('MAE')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```



1.16 Conclusion

1. For NLIC LSTM and RNN Models used for Stock Price Prediction
2. The Error is Low for LSTM Model