Assignment4_Hydro_UPPER

July 30, 2022

1 Stock Price Prediction of UPPER

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 05:07:22.955277: 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 05:07:22.955316: 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]:
```
upper_df = pd.read_csv("data/UPPER.csv")
upper_df.shape
```

[3]:
```
(661, 8)
```

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

[4]:

	S.N.	Date	Total Transactions	Total Traded Shares	\
0	1	2021-12-29	412	57495.0	
1	2	2021-12-28	624	64439.0	
2	3	2021-12-27	759	125794.0	
3	4	2021-12-26	797	145788.0	

```
4     5  2021-12-23                1113             151904.0

   Total Traded Amount  Max. Price  Min. Price  Close Price
0           35461484.7       633.0       612.0        618.0
1           39617082.6       625.0       606.0        624.0
2           79522847.0       650.0       616.1        617.0
3           93525227.6       650.0       630.0        642.0
4           98281767.0       675.0       636.0        637.0
```

1.3 Renaming the Columns

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

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

[6]:
```
   SN        Date  TTrans       TTS         TTA  MaxPrice  MinPrice  \
0   1  2021-12-29     412   57495.0  35461484.7     633.0     612.0
1   2  2021-12-28     624   64439.0  39617082.6     625.0     606.0
2   3  2021-12-27     759  125794.0  79522847.0     650.0     616.1
3   4  2021-12-26     797  145788.0  93525227.6     650.0     630.0
4   5  2021-12-23    1113  151904.0  98281767.0     675.0     636.0

   ClosePrice
0       618.0
1       624.0
2       617.0
3       642.0
4       637.0
```

[7]:
```
upper_df.shape
```

[7]:
```
(661, 8)
```

Converting the Date into Panda's Date Time

[8]:
```
upper_df['Date'] = pd.to_datetime(upper_df['Date'])
```

1.4 Sorting the Date by Date in Ascending Order

[9]:
```
upper_df=upper_df.sort_values(by='Date')
```

1.5 Setting Features and Target Column

[10]:
```
features = ['Date','ClosePrice']
```

[11]:
```
X = upper_df[features]
```

[12]:
```
X.set_index("Date",inplace=True)
```

1.6 Splitting the Data Into Training, Validation and Test Set

[13]:
```
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

[14]:
```
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

[15]:
```
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

[16]:
```
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

[17]:
```
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 05:08:02.197798: 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 05:08:02.197876: W
tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit:
UNKNOWN ERROR (303)
2022-07-30 05:08:02.197916: 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 05:08:02.198324: 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.
```

[18]:
```
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

[19]:
```
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'])
```

[20]:
```
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

[21]:
```
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
11/11 - 2s - loss: 1.0581 - mae: 0.7296 - val_loss: 0.8030 - val_mae: 0.7116 -
2s/epoch - 194ms/step
Epoch 2/100
11/11 - 0s - loss: 0.7770 - mae: 0.5907 - val_loss: 0.7831 - val_mae: 0.7048 -
102ms/epoch - 9ms/step
Epoch 3/100
11/11 - 0s - loss: 0.5997 - mae: 0.5100 - val_loss: 0.8085 - val_mae: 0.7212 -
91ms/epoch - 8ms/step
Epoch 4/100
11/11 - 0s - loss: 0.5247 - mae: 0.4738 - val_loss: 0.8471 - val_mae: 0.7474 -
93ms/epoch - 8ms/step
Epoch 5/100
11/11 - 0s - loss: 0.4536 - mae: 0.4396 - val_loss: 0.8658 - val_mae: 0.7632 -
96ms/epoch - 9ms/step
Epoch 6/100
```

```
11/11 - 0s - loss: 0.3882 - mae: 0.4132 - val_loss: 0.8800 - val_mae: 0.7723 -
89ms/epoch - 8ms/step
Epoch 7/100
11/11 - 0s - loss: 0.3468 - mae: 0.3891 - val_loss: 0.8360 - val_mae: 0.7549 -
89ms/epoch - 8ms/step
Epoch 8/100
11/11 - 0s - loss: 0.3001 - mae: 0.3619 - val_loss: 0.8284 - val_mae: 0.7521 -
90ms/epoch - 8ms/step
Epoch 9/100
11/11 - 0s - loss: 0.2959 - mae: 0.3563 - val_loss: 0.8273 - val_mae: 0.7552 -
84ms/epoch - 8ms/step
Epoch 10/100
11/11 - 0s - loss: 0.2629 - mae: 0.3417 - val_loss: 0.7992 - val_mae: 0.7422 -
92ms/epoch - 8ms/step
Epoch 11/100
11/11 - 0s - loss: 0.2600 - mae: 0.3323 - val_loss: 0.7187 - val_mae: 0.6997 -
92ms/epoch - 8ms/step
Epoch 12/100
11/11 - 0s - loss: 0.2171 - mae: 0.3109 - val_loss: 0.6928 - val_mae: 0.6860 -
95ms/epoch - 9ms/step
Epoch 13/100
11/11 - 0s - loss: 0.2009 - mae: 0.2926 - val_loss: 0.6177 - val_mae: 0.6395 -
94ms/epoch - 9ms/step
Epoch 14/100
11/11 - 0s - loss: 0.1988 - mae: 0.2949 - val_loss: 0.6451 - val_mae: 0.6585 -
95ms/epoch - 9ms/step
Epoch 15/100
11/11 - 0s - loss: 0.1895 - mae: 0.2868 - val_loss: 0.5852 - val_mae: 0.6267 -
92ms/epoch - 8ms/step
Epoch 16/100
11/11 - 0s - loss: 0.1659 - mae: 0.2678 - val_loss: 0.5250 - val_mae: 0.5782 -
93ms/epoch - 8ms/step
Epoch 17/100
11/11 - 0s - loss: 0.1701 - mae: 0.2732 - val_loss: 0.5123 - val_mae: 0.5753 -
95ms/epoch - 9ms/step
Epoch 18/100
11/11 - 0s - loss: 0.1532 - mae: 0.2573 - val_loss: 0.4941 - val_mae: 0.5616 -
90ms/epoch - 8ms/step
Epoch 19/100
11/11 - 0s - loss: 0.1328 - mae: 0.2458 - val_loss: 0.4509 - val_mae: 0.5298 -
88ms/epoch - 8ms/step
Epoch 20/100
11/11 - 0s - loss: 0.1391 - mae: 0.2470 - val_loss: 0.4468 - val_mae: 0.5245 -
94ms/epoch - 9ms/step
Epoch 21/100
11/11 - 0s - loss: 0.1309 - mae: 0.2383 - val_loss: 0.4164 - val_mae: 0.5060 -
103ms/epoch - 9ms/step
Epoch 22/100
```

```
11/11 - 0s - loss: 0.1260 - mae: 0.2348 - val_loss: 0.4353 - val_mae: 0.5040 -
97ms/epoch - 9ms/step
Epoch 23/100
11/11 - 0s - loss: 0.1306 - mae: 0.2448 - val_loss: 0.4141 - val_mae: 0.5066 -
101ms/epoch - 9ms/step
Epoch 24/100
11/11 - 0s - loss: 0.1188 - mae: 0.2375 - val_loss: 0.4274 - val_mae: 0.5051 -
96ms/epoch - 9ms/step
Epoch 25/100
11/11 - 0s - loss: 0.1269 - mae: 0.2312 - val_loss: 0.4011 - val_mae: 0.4784 -
95ms/epoch - 9ms/step
Epoch 26/100
11/11 - 0s - loss: 0.1264 - mae: 0.2361 - val_loss: 0.3975 - val_mae: 0.4984 -
117ms/epoch - 11ms/step
Epoch 27/100
11/11 - 0s - loss: 0.1193 - mae: 0.2404 - val_loss: 0.4254 - val_mae: 0.4947 -
124ms/epoch - 11ms/step
Epoch 28/100
11/11 - 0s - loss: 0.1242 - mae: 0.2490 - val_loss: 0.3771 - val_mae: 0.4856 -
113ms/epoch - 10ms/step
Epoch 29/100
11/11 - 0s - loss: 0.1250 - mae: 0.2587 - val_loss: 0.4005 - val_mae: 0.4891 -
92ms/epoch - 8ms/step
Epoch 30/100
11/11 - 0s - loss: 0.1080 - mae: 0.2220 - val_loss: 0.3885 - val_mae: 0.4866 -
104ms/epoch - 9ms/step
Epoch 31/100
11/11 - 0s - loss: 0.1119 - mae: 0.2213 - val_loss: 0.3681 - val_mae: 0.4676 -
133ms/epoch - 12ms/step
Epoch 32/100
11/11 - 0s - loss: 0.1234 - mae: 0.2335 - val_loss: 0.4236 - val_mae: 0.5196 -
109ms/epoch - 10ms/step
Epoch 33/100
11/11 - 0s - loss: 0.1102 - mae: 0.2227 - val_loss: 0.3569 - val_mae: 0.4644 -
122ms/epoch - 11ms/step
Epoch 34/100
11/11 - 0s - loss: 0.1320 - mae: 0.2430 - val_loss: 0.4090 - val_mae: 0.5098 -
127ms/epoch - 12ms/step
Epoch 35/100
11/11 - 0s - loss: 0.1354 - mae: 0.2412 - val_loss: 0.3380 - val_mae: 0.4508 -
121ms/epoch - 11ms/step
Epoch 36/100
11/11 - 0s - loss: 0.1439 - mae: 0.2511 - val_loss: 0.4496 - val_mae: 0.5365 -
116ms/epoch - 11ms/step
Epoch 37/100
11/11 - 0s - loss: 0.1547 - mae: 0.2510 - val_loss: 0.3309 - val_mae: 0.4535 -
105ms/epoch - 10ms/step
Epoch 38/100
```

```
11/11 - 0s - loss: 0.1396 - mae: 0.2476 - val_loss: 0.5213 - val_mae: 0.5870 -
108ms/epoch - 10ms/step
Epoch 39/100
11/11 - 0s - loss: 0.1189 - mae: 0.2205 - val_loss: 0.3732 - val_mae: 0.4972 -
146ms/epoch - 13ms/step
Epoch 40/100
11/11 - 0s - loss: 0.1062 - mae: 0.2154 - val_loss: 0.4696 - val_mae: 0.5580 -
120ms/epoch - 11ms/step
Epoch 41/100
11/11 - 0s - loss: 0.0941 - mae: 0.1992 - val_loss: 0.3509 - val_mae: 0.4772 -
112ms/epoch - 10ms/step
Epoch 42/100
11/11 - 0s - loss: 0.0896 - mae: 0.1971 - val_loss: 0.3914 - val_mae: 0.5005 -
117ms/epoch - 11ms/step
Epoch 43/100
11/11 - 0s - loss: 0.1023 - mae: 0.2016 - val_loss: 0.3855 - val_mae: 0.4952 -
110ms/epoch - 10ms/step
Epoch 44/100
11/11 - 0s - loss: 0.0857 - mae: 0.1893 - val_loss: 0.3578 - val_mae: 0.4720 -
111ms/epoch - 10ms/step
Epoch 45/100
11/11 - 0s - loss: 0.0842 - mae: 0.1921 - val_loss: 0.3763 - val_mae: 0.4867 -
110ms/epoch - 10ms/step
Epoch 46/100
11/11 - 0s - loss: 0.0824 - mae: 0.1904 - val_loss: 0.3458 - val_mae: 0.4585 -
85ms/epoch - 8ms/step
Epoch 47/100
11/11 - 0s - loss: 0.0778 - mae: 0.1833 - val_loss: 0.3618 - val_mae: 0.4737 -
90ms/epoch - 8ms/step
Epoch 48/100
11/11 - 0s - loss: 0.0789 - mae: 0.1851 - val_loss: 0.3404 - val_mae: 0.4574 -
90ms/epoch - 8ms/step
Epoch 49/100
11/11 - 0s - loss: 0.0911 - mae: 0.1969 - val_loss: 0.3896 - val_mae: 0.5009 -
106ms/epoch - 10ms/step
Epoch 50/100
11/11 - 0s - loss: 0.0874 - mae: 0.1912 - val_loss: 0.3317 - val_mae: 0.4544 -
95ms/epoch - 9ms/step
Epoch 51/100
11/11 - 0s - loss: 0.1013 - mae: 0.2037 - val_loss: 0.4125 - val_mae: 0.5130 -
81ms/epoch - 7ms/step
Epoch 52/100
11/11 - 0s - loss: 0.0876 - mae: 0.1855 - val_loss: 0.3269 - val_mae: 0.4531 -
90ms/epoch - 8ms/step
Epoch 53/100
11/11 - 0s - loss: 0.0854 - mae: 0.1903 - val_loss: 0.3998 - val_mae: 0.5075 -
98ms/epoch - 9ms/step
Epoch 54/100
```

```
11/11 - 0s - loss: 0.0745 - mae: 0.1771 - val_loss: 0.3262 - val_mae: 0.4517 -
93ms/epoch - 8ms/step
Epoch 55/100
11/11 - 0s - loss: 0.0736 - mae: 0.1773 - val_loss: 0.3653 - val_mae: 0.4762 -
85ms/epoch - 8ms/step
Epoch 56/100
11/11 - 0s - loss: 0.0636 - mae: 0.1682 - val_loss: 0.3380 - val_mae: 0.4550 -
87ms/epoch - 8ms/step
Epoch 57/100
11/11 - 0s - loss: 0.0708 - mae: 0.1726 - val_loss: 0.3465 - val_mae: 0.4612 -
85ms/epoch - 8ms/step
Epoch 58/100
11/11 - 0s - loss: 0.0682 - mae: 0.1706 - val_loss: 0.3229 - val_mae: 0.4402 -
86ms/epoch - 8ms/step
Epoch 59/100
11/11 - 0s - loss: 0.0711 - mae: 0.1701 - val_loss: 0.3470 - val_mae: 0.4594 -
79ms/epoch - 7ms/step
Epoch 60/100
11/11 - 0s - loss: 0.0733 - mae: 0.1744 - val_loss: 0.3471 - val_mae: 0.4633 -
87ms/epoch - 8ms/step
Epoch 61/100
11/11 - 0s - loss: 0.0593 - mae: 0.1616 - val_loss: 0.3281 - val_mae: 0.4486 -
87ms/epoch - 8ms/step
Epoch 62/100
11/11 - 0s - loss: 0.0823 - mae: 0.1858 - val_loss: 0.3201 - val_mae: 0.4333 -
94ms/epoch - 9ms/step
Epoch 63/100
11/11 - 0s - loss: 0.1063 - mae: 0.2164 - val_loss: 0.3832 - val_mae: 0.5037 -
99ms/epoch - 9ms/step
Epoch 64/100
11/11 - 0s - loss: 0.1289 - mae: 0.2329 - val_loss: 0.3013 - val_mae: 0.4398 -
100ms/epoch - 9ms/step
Epoch 65/100
11/11 - 0s - loss: 0.1195 - mae: 0.2250 - val_loss: 0.4458 - val_mae: 0.5473 -
117ms/epoch - 11ms/step
Epoch 66/100
11/11 - 0s - loss: 0.1128 - mae: 0.2051 - val_loss: 0.3362 - val_mae: 0.4743 -
90ms/epoch - 8ms/step
Epoch 67/100
11/11 - 0s - loss: 0.1113 - mae: 0.2149 - val_loss: 0.4517 - val_mae: 0.5581 -
85ms/epoch - 8ms/step
Epoch 68/100
11/11 - 0s - loss: 0.0866 - mae: 0.1869 - val_loss: 0.3294 - val_mae: 0.4718 -
85ms/epoch - 8ms/step
Epoch 69/100
11/11 - 0s - loss: 0.0792 - mae: 0.1856 - val_loss: 0.4148 - val_mae: 0.5290 -
99ms/epoch - 9ms/step
Epoch 70/100
```

```
11/11 - 0s - loss: 0.0716 - mae: 0.1708 - val_loss: 0.3309 - val_mae: 0.4666 -
105ms/epoch - 10ms/step
Epoch 71/100
11/11 - 0s - loss: 0.0623 - mae: 0.1656 - val_loss: 0.3593 - val_mae: 0.4864 -
109ms/epoch - 10ms/step
Epoch 72/100
11/11 - 0s - loss: 0.0703 - mae: 0.1694 - val_loss: 0.3663 - val_mae: 0.4914 -
119ms/epoch - 11ms/step
Epoch 73/100
11/11 - 0s - loss: 0.0671 - mae: 0.1655 - val_loss: 0.3409 - val_mae: 0.4708 -
116ms/epoch - 11ms/step
Epoch 74/100
11/11 - 0s - loss: 0.0660 - mae: 0.1693 - val_loss: 0.3501 - val_mae: 0.4748 -
105ms/epoch - 10ms/step
Epoch 75/100
11/11 - 0s - loss: 0.0656 - mae: 0.1623 - val_loss: 0.3305 - val_mae: 0.4583 -
105ms/epoch - 10ms/step
Epoch 76/100
11/11 - 0s - loss: 0.0713 - mae: 0.1673 - val_loss: 0.3496 - val_mae: 0.4726 -
126ms/epoch - 11ms/step
Epoch 77/100
11/11 - 0s - loss: 0.0640 - mae: 0.1630 - val_loss: 0.3164 - val_mae: 0.4438 -
116ms/epoch - 11ms/step
Epoch 78/100
11/11 - 0s - loss: 0.0575 - mae: 0.1605 - val_loss: 0.3570 - val_mae: 0.4759 -
106ms/epoch - 10ms/step
Epoch 79/100
11/11 - 0s - loss: 0.0629 - mae: 0.1600 - val_loss: 0.3254 - val_mae: 0.4536 -
102ms/epoch - 9ms/step
Epoch 80/100
11/11 - 0s - loss: 0.0613 - mae: 0.1630 - val_loss: 0.3580 - val_mae: 0.4808 -
115ms/epoch - 10ms/step
Epoch 81/100
11/11 - 0s - loss: 0.0693 - mae: 0.1648 - val_loss: 0.3145 - val_mae: 0.4449 -
124ms/epoch - 11ms/step
Epoch 82/100
11/11 - 0s - loss: 0.0654 - mae: 0.1623 - val_loss: 0.3562 - val_mae: 0.4791 -
118ms/epoch - 11ms/step
Epoch 83/100
11/11 - 0s - loss: 0.0566 - mae: 0.1528 - val_loss: 0.2984 - val_mae: 0.4339 -
111ms/epoch - 10ms/step
Epoch 84/100
11/11 - 0s - loss: 0.0569 - mae: 0.1554 - val_loss: 0.3567 - val_mae: 0.4809 -
80ms/epoch - 7ms/step
Epoch 85/100
11/11 - 0s - loss: 0.0636 - mae: 0.1592 - val_loss: 0.3042 - val_mae: 0.4392 -
87ms/epoch - 8ms/step
Epoch 86/100
```

```
11/11 - 0s - loss: 0.0588 - mae: 0.1577 - val_loss: 0.3668 - val_mae: 0.4869 -
87ms/epoch - 8ms/step
Epoch 87/100
11/11 - 0s - loss: 0.0592 - mae: 0.1576 - val_loss: 0.3127 - val_mae: 0.4489 -
83ms/epoch - 8ms/step
Epoch 88/100
11/11 - 0s - loss: 0.0591 - mae: 0.1611 - val_loss: 0.3550 - val_mae: 0.4778 -
89ms/epoch - 8ms/step
Epoch 89/100
11/11 - 0s - loss: 0.0528 - mae: 0.1496 - val_loss: 0.3319 - val_mae: 0.4592 -
79ms/epoch - 7ms/step
Epoch 90/100
11/11 - 0s - loss: 0.0509 - mae: 0.1467 - val_loss: 0.3214 - val_mae: 0.4481 -
95ms/epoch - 9ms/step
Epoch 91/100
11/11 - 0s - loss: 0.0570 - mae: 0.1570 - val_loss: 0.3451 - val_mae: 0.4683 -
90ms/epoch - 8ms/step
Epoch 92/100
11/11 - 0s - loss: 0.0572 - mae: 0.1502 - val_loss: 0.3104 - val_mae: 0.4409 -
83ms/epoch - 8ms/step
Epoch 93/100
11/11 - 0s - loss: 0.0494 - mae: 0.1460 - val_loss: 0.3316 - val_mae: 0.4515 -
82ms/epoch - 7ms/step
Epoch 94/100
11/11 - 0s - loss: 0.0488 - mae: 0.1496 - val_loss: 0.3380 - val_mae: 0.4601 -
76ms/epoch - 7ms/step
Epoch 95/100
11/11 - 0s - loss: 0.0520 - mae: 0.1473 - val_loss: 0.3157 - val_mae: 0.4433 -
80ms/epoch - 7ms/step
Epoch 96/100
11/11 - 0s - loss: 0.0599 - mae: 0.1532 - val_loss: 0.3211 - val_mae: 0.4437 -
89ms/epoch - 8ms/step
Epoch 97/100
11/11 - 0s - loss: 0.0538 - mae: 0.1476 - val_loss: 0.3259 - val_mae: 0.4509 -
86ms/epoch - 8ms/step
Epoch 98/100
11/11 - 0s - loss: 0.0586 - mae: 0.1569 - val_loss: 0.3156 - val_mae: 0.4417 -
84ms/epoch - 8ms/step
Epoch 99/100
11/11 - 0s - loss: 0.0569 - mae: 0.1573 - val_loss: 0.3622 - val_mae: 0.4809 -
87ms/epoch - 8ms/step
Epoch 100/100
11/11 - 0s - loss: 0.0525 - mae: 0.1505 - val_loss: 0.3075 - val_mae: 0.4339 -
78ms/epoch - 7ms/step
```

[22]:
```
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
11/11 - 5s - loss: 0.9745 - mae: 0.7424 - val_loss: 0.8045 - val_mae: 0.7187 -
5s/epoch - 438ms/step
Epoch 2/100
11/11 - 0s - loss: 0.8979 - mae: 0.7042 - val_loss: 0.7719 - val_mae: 0.6990 -
134ms/epoch - 12ms/step
Epoch 3/100
11/11 - 0s - loss: 0.7660 - mae: 0.6468 - val_loss: 0.7220 - val_mae: 0.6646 -
131ms/epoch - 12ms/step
Epoch 4/100
11/11 - 0s - loss: 0.5519 - mae: 0.5347 - val_loss: 0.6877 - val_mae: 0.6438 -
133ms/epoch - 12ms/step
Epoch 5/100
11/11 - 0s - loss: 0.3475 - mae: 0.3901 - val_loss: 0.7907 - val_mae: 0.7140 -
146ms/epoch - 13ms/step
Epoch 6/100
11/11 - 0s - loss: 0.2454 - mae: 0.3166 - val_loss: 0.9823 - val_mae: 0.8088 -
142ms/epoch - 13ms/step
Epoch 7/100
11/11 - 0s - loss: 0.1991 - mae: 0.2853 - val_loss: 0.9741 - val_mae: 0.8095 -
139ms/epoch - 13ms/step
Epoch 8/100
11/11 - 0s - loss: 0.1639 - mae: 0.2598 - val_loss: 0.8834 - val_mae: 0.7691 -
128ms/epoch - 12ms/step
Epoch 9/100
11/11 - 0s - loss: 0.1558 - mae: 0.2479 - val_loss: 0.8234 - val_mae: 0.7395 -
125ms/epoch - 11ms/step
Epoch 10/100
11/11 - 0s - loss: 0.1262 - mae: 0.2292 - val_loss: 0.8732 - val_mae: 0.7653 -
118ms/epoch - 11ms/step
Epoch 11/100
11/11 - 0s - loss: 0.1167 - mae: 0.2189 - val_loss: 0.8485 - val_mae: 0.7532 -
132ms/epoch - 12ms/step
Epoch 12/100
11/11 - 0s - loss: 0.1192 - mae: 0.2240 - val_loss: 0.8063 - val_mae: 0.7322 -
137ms/epoch - 12ms/step
Epoch 13/100
11/11 - 0s - loss: 0.1079 - mae: 0.2146 - val_loss: 0.8165 - val_mae: 0.7381 -
124ms/epoch - 11ms/step
Epoch 14/100
11/11 - 0s - loss: 0.1048 - mae: 0.2113 - val_loss: 0.8381 - val_mae: 0.7495 -
139ms/epoch - 13ms/step
Epoch 15/100
11/11 - 0s - loss: 0.1030 - mae: 0.2065 - val_loss: 0.7998 - val_mae: 0.7302 -
131ms/epoch - 12ms/step
```

```
Epoch 16/100
11/11 - 0s - loss: 0.1034 - mae: 0.2036 - val_loss: 0.8292 - val_mae: 0.7460 -
133ms/epoch - 12ms/step
Epoch 17/100
11/11 - 0s - loss: 0.0949 - mae: 0.1992 - val_loss: 0.7956 - val_mae: 0.7288 -
123ms/epoch - 11ms/step
Epoch 18/100
11/11 - 0s - loss: 0.0947 - mae: 0.1976 - val_loss: 0.8299 - val_mae: 0.7474 -
133ms/epoch - 12ms/step
Epoch 19/100
11/11 - 0s - loss: 0.0950 - mae: 0.1950 - val_loss: 0.8451 - val_mae: 0.7552 -
132ms/epoch - 12ms/step
Epoch 20/100
11/11 - 0s - loss: 0.0974 - mae: 0.1999 - val_loss: 0.7952 - val_mae: 0.7295 -
138ms/epoch - 13ms/step
Epoch 21/100
11/11 - 0s - loss: 0.0880 - mae: 0.1926 - val_loss: 0.8250 - val_mae: 0.7457 -
142ms/epoch - 13ms/step
Epoch 22/100
11/11 - 0s - loss: 0.0851 - mae: 0.1870 - val_loss: 0.7640 - val_mae: 0.7122 -
142ms/epoch - 13ms/step
Epoch 23/100
11/11 - 0s - loss: 0.0928 - mae: 0.1919 - val_loss: 0.7593 - val_mae: 0.7093 -
128ms/epoch - 12ms/step
Epoch 24/100
11/11 - 0s - loss: 0.0929 - mae: 0.1945 - val_loss: 0.8716 - val_mae: 0.7711 -
136ms/epoch - 12ms/step
Epoch 25/100
11/11 - 0s - loss: 0.0872 - mae: 0.1831 - val_loss: 0.7537 - val_mae: 0.7051 -
139ms/epoch - 13ms/step
Epoch 26/100
11/11 - 0s - loss: 0.0820 - mae: 0.1834 - val_loss: 0.8656 - val_mae: 0.7683 -
130ms/epoch - 12ms/step
Epoch 27/100
11/11 - 0s - loss: 0.0864 - mae: 0.1828 - val_loss: 0.8519 - val_mae: 0.7613 -
134ms/epoch - 12ms/step
Epoch 28/100
11/11 - 0s - loss: 0.0888 - mae: 0.1832 - val_loss: 0.7570 - val_mae: 0.7077 -
131ms/epoch - 12ms/step
Epoch 29/100
11/11 - 0s - loss: 0.0756 - mae: 0.1771 - val_loss: 0.8836 - val_mae: 0.7783 -
261ms/epoch - 24ms/step
Epoch 30/100
11/11 - 0s - loss: 0.0749 - mae: 0.1725 - val_loss: 0.8094 - val_mae: 0.7393 -
173ms/epoch - 16ms/step
Epoch 31/100
11/11 - 0s - loss: 0.0753 - mae: 0.1743 - val_loss: 0.8309 - val_mae: 0.7509 -
133ms/epoch - 12ms/step
```

```
Epoch 32/100
11/11 - 0s - loss: 0.0760 - mae: 0.1729 - val_loss: 0.7856 - val_mae: 0.7264 -
125ms/epoch - 11ms/step
Epoch 33/100
11/11 - 0s - loss: 0.0713 - mae: 0.1706 - val_loss: 0.8476 - val_mae: 0.7600 -
127ms/epoch - 12ms/step
Epoch 34/100
11/11 - 0s - loss: 0.0730 - mae: 0.1697 - val_loss: 0.7688 - val_mae: 0.7169 -
129ms/epoch - 12ms/step
Epoch 35/100
11/11 - 0s - loss: 0.0708 - mae: 0.1676 - val_loss: 0.7859 - val_mae: 0.7269 -
124ms/epoch - 11ms/step
Epoch 36/100
11/11 - 0s - loss: 0.0647 - mae: 0.1655 - val_loss: 0.8193 - val_mae: 0.7453 -
118ms/epoch - 11ms/step
Epoch 37/100
11/11 - 0s - loss: 0.0698 - mae: 0.1685 - val_loss: 0.7718 - val_mae: 0.7185 -
140ms/epoch - 13ms/step
Epoch 38/100
11/11 - 0s - loss: 0.0748 - mae: 0.1720 - val_loss: 0.8236 - val_mae: 0.7476 -
128ms/epoch - 12ms/step
Epoch 39/100
11/11 - 0s - loss: 0.0777 - mae: 0.1723 - val_loss: 0.7613 - val_mae: 0.7119 -
130ms/epoch - 12ms/step
Epoch 40/100
11/11 - 0s - loss: 0.0730 - mae: 0.1691 - val_loss: 0.8619 - val_mae: 0.7684 -
134ms/epoch - 12ms/step
Epoch 41/100
11/11 - 0s - loss: 0.0771 - mae: 0.1726 - val_loss: 0.7545 - val_mae: 0.7073 -
135ms/epoch - 12ms/step
Epoch 42/100
11/11 - 0s - loss: 0.0693 - mae: 0.1688 - val_loss: 0.8688 - val_mae: 0.7720 -
129ms/epoch - 12ms/step
Epoch 43/100
11/11 - 0s - loss: 0.0719 - mae: 0.1683 - val_loss: 0.7553 - val_mae: 0.7077 -
135ms/epoch - 12ms/step
Epoch 44/100
11/11 - 0s - loss: 0.0652 - mae: 0.1625 - val_loss: 0.8847 - val_mae: 0.7809 -
124ms/epoch - 11ms/step
Epoch 45/100
11/11 - 0s - loss: 0.0589 - mae: 0.1554 - val_loss: 0.8145 - val_mae: 0.7433 -
125ms/epoch - 11ms/step
Epoch 46/100
11/11 - 0s - loss: 0.0744 - mae: 0.1670 - val_loss: 0.7908 - val_mae: 0.7299 -
131ms/epoch - 12ms/step
Epoch 47/100
11/11 - 0s - loss: 0.0717 - mae: 0.1719 - val_loss: 0.9296 - val_mae: 0.8042 -
132ms/epoch - 12ms/step
```

```
Epoch 48/100
11/11 - 0s - loss: 0.0709 - mae: 0.1644 - val_loss: 0.7483 - val_mae: 0.7052 -
131ms/epoch - 12ms/step
Epoch 49/100
11/11 - 0s - loss: 0.0639 - mae: 0.1594 - val_loss: 0.7788 - val_mae: 0.7235 -
125ms/epoch - 11ms/step
Epoch 50/100
11/11 - 0s - loss: 0.0701 - mae: 0.1617 - val_loss: 0.7959 - val_mae: 0.7330 -
125ms/epoch - 11ms/step
Epoch 51/100
11/11 - 0s - loss: 0.0772 - mae: 0.1724 - val_loss: 0.8539 - val_mae: 0.7654 -
126ms/epoch - 11ms/step
Epoch 52/100
11/11 - 0s - loss: 0.0724 - mae: 0.1656 - val_loss: 0.7138 - val_mae: 0.6815 -
133ms/epoch - 12ms/step
Epoch 53/100
11/11 - 0s - loss: 0.0634 - mae: 0.1604 - val_loss: 0.8689 - val_mae: 0.7730 -
127ms/epoch - 12ms/step
Epoch 54/100
11/11 - 0s - loss: 0.0618 - mae: 0.1579 - val_loss: 0.7762 - val_mae: 0.7217 -
130ms/epoch - 12ms/step
Epoch 55/100
11/11 - 0s - loss: 0.0533 - mae: 0.1500 - val_loss: 0.8463 - val_mae: 0.7613 -
120ms/epoch - 11ms/step
Epoch 56/100
11/11 - 0s - loss: 0.0586 - mae: 0.1521 - val_loss: 0.8047 - val_mae: 0.7387 -
123ms/epoch - 11ms/step
Epoch 57/100
11/11 - 0s - loss: 0.0571 - mae: 0.1506 - val_loss: 0.7833 - val_mae: 0.7266 -
125ms/epoch - 11ms/step
Epoch 58/100
11/11 - 0s - loss: 0.0518 - mae: 0.1475 - val_loss: 0.8085 - val_mae: 0.7403 -
126ms/epoch - 11ms/step
Epoch 59/100
11/11 - 0s - loss: 0.0531 - mae: 0.1501 - val_loss: 0.7992 - val_mae: 0.7357 -
116ms/epoch - 11ms/step
Epoch 60/100
11/11 - 0s - loss: 0.0518 - mae: 0.1496 - val_loss: 0.7722 - val_mae: 0.7204 -
123ms/epoch - 11ms/step
Epoch 61/100
11/11 - 0s - loss: 0.0504 - mae: 0.1442 - val_loss: 0.7916 - val_mae: 0.7312 -
133ms/epoch - 12ms/step
Epoch 62/100
11/11 - 0s - loss: 0.0524 - mae: 0.1468 - val_loss: 0.7450 - val_mae: 0.7034 -
138ms/epoch - 13ms/step
Epoch 63/100
11/11 - 0s - loss: 0.0506 - mae: 0.1453 - val_loss: 0.8217 - val_mae: 0.7484 -
147ms/epoch - 13ms/step
```

```
Epoch 64/100
11/11 - 0s - loss: 0.0601 - mae: 0.1505 - val_loss: 0.7728 - val_mae: 0.7207 -
156ms/epoch - 14ms/step
Epoch 65/100
11/11 - 0s - loss: 0.0502 - mae: 0.1427 - val_loss: 0.8236 - val_mae: 0.7500 -
128ms/epoch - 12ms/step
Epoch 66/100
11/11 - 0s - loss: 0.0579 - mae: 0.1470 - val_loss: 0.7458 - val_mae: 0.7046 -
129ms/epoch - 12ms/step
Epoch 67/100
11/11 - 0s - loss: 0.0542 - mae: 0.1470 - val_loss: 0.8157 - val_mae: 0.7453 -
121ms/epoch - 11ms/step
Epoch 68/100
11/11 - 0s - loss: 0.0453 - mae: 0.1401 - val_loss: 0.7519 - val_mae: 0.7082 -
125ms/epoch - 11ms/step
Epoch 69/100
11/11 - 0s - loss: 0.0505 - mae: 0.1435 - val_loss: 0.8158 - val_mae: 0.7448 -
125ms/epoch - 11ms/step
Epoch 70/100
11/11 - 0s - loss: 0.0593 - mae: 0.1513 - val_loss: 0.8491 - val_mae: 0.7655 -
122ms/epoch - 11ms/step
Epoch 71/100
11/11 - 0s - loss: 0.0689 - mae: 0.1613 - val_loss: 0.7228 - val_mae: 0.6888 -
154ms/epoch - 14ms/step
Epoch 72/100
11/11 - 0s - loss: 0.0553 - mae: 0.1520 - val_loss: 0.8945 - val_mae: 0.7879 -
261ms/epoch - 24ms/step
Epoch 73/100
11/11 - 0s - loss: 0.0480 - mae: 0.1418 - val_loss: 0.7587 - val_mae: 0.7124 -
125ms/epoch - 11ms/step
Epoch 74/100
11/11 - 0s - loss: 0.0519 - mae: 0.1448 - val_loss: 0.8144 - val_mae: 0.7441 -
118ms/epoch - 11ms/step
Epoch 75/100
11/11 - 0s - loss: 0.0480 - mae: 0.1413 - val_loss: 0.8081 - val_mae: 0.7414 -
143ms/epoch - 13ms/step
Epoch 76/100
11/11 - 0s - loss: 0.0574 - mae: 0.1476 - val_loss: 0.7644 - val_mae: 0.7164 -
203ms/epoch - 18ms/step
Epoch 77/100
11/11 - 0s - loss: 0.0520 - mae: 0.1461 - val_loss: 0.7901 - val_mae: 0.7315 -
127ms/epoch - 12ms/step
Epoch 78/100
11/11 - 0s - loss: 0.0491 - mae: 0.1439 - val_loss: 0.7616 - val_mae: 0.7143 -
125ms/epoch - 11ms/step
Epoch 79/100
11/11 - 0s - loss: 0.0538 - mae: 0.1475 - val_loss: 0.7891 - val_mae: 0.7301 -
168ms/epoch - 15ms/step
```

```
Epoch 80/100
11/11 - 0s - loss: 0.0471 - mae: 0.1400 - val_loss: 0.8415 - val_mae: 0.7624 -
146ms/epoch - 13ms/step
Epoch 81/100
11/11 - 0s - loss: 0.0516 - mae: 0.1443 - val_loss: 0.7453 - val_mae: 0.7056 -
151ms/epoch - 14ms/step
Epoch 82/100
11/11 - 0s - loss: 0.0554 - mae: 0.1472 - val_loss: 0.8020 - val_mae: 0.7378 -
136ms/epoch - 12ms/step
Epoch 83/100
11/11 - 0s - loss: 0.0474 - mae: 0.1416 - val_loss: 0.7636 - val_mae: 0.7162 -
157ms/epoch - 14ms/step
Epoch 84/100
11/11 - 0s - loss: 0.0487 - mae: 0.1436 - val_loss: 0.7941 - val_mae: 0.7339 -
149ms/epoch - 14ms/step
Epoch 85/100
11/11 - 0s - loss: 0.0425 - mae: 0.1341 - val_loss: 0.7668 - val_mae: 0.7186 -
127ms/epoch - 12ms/step
Epoch 86/100
11/11 - 0s - loss: 0.0404 - mae: 0.1304 - val_loss: 0.7649 - val_mae: 0.7171 -
126ms/epoch - 11ms/step
Epoch 87/100
11/11 - 0s - loss: 0.0423 - mae: 0.1310 - val_loss: 0.7803 - val_mae: 0.7266 -
119ms/epoch - 11ms/step
Epoch 88/100
11/11 - 0s - loss: 0.0465 - mae: 0.1360 - val_loss: 0.7672 - val_mae: 0.7192 -
122ms/epoch - 11ms/step
Epoch 89/100
11/11 - 0s - loss: 0.0507 - mae: 0.1405 - val_loss: 0.7675 - val_mae: 0.7190 -
124ms/epoch - 11ms/step
Epoch 90/100
11/11 - 0s - loss: 0.0434 - mae: 0.1359 - val_loss: 0.8093 - val_mae: 0.7446 -
123ms/epoch - 11ms/step
Epoch 91/100
11/11 - 0s - loss: 0.0443 - mae: 0.1360 - val_loss: 0.7390 - val_mae: 0.7033 -
119ms/epoch - 11ms/step
Epoch 92/100
11/11 - 0s - loss: 0.0592 - mae: 0.1500 - val_loss: 0.7526 - val_mae: 0.7095 -
144ms/epoch - 13ms/step
Epoch 93/100
11/11 - 0s - loss: 0.0473 - mae: 0.1382 - val_loss: 0.8259 - val_mae: 0.7546 -
140ms/epoch - 13ms/step
Epoch 94/100
11/11 - 0s - loss: 0.0529 - mae: 0.1438 - val_loss: 0.7518 - val_mae: 0.7109 -
142ms/epoch - 13ms/step
Epoch 95/100
11/11 - 0s - loss: 0.0430 - mae: 0.1351 - val_loss: 0.7899 - val_mae: 0.7325 -
131ms/epoch - 12ms/step
```

```
Epoch 96/100
11/11 - 0s - loss: 0.0395 - mae: 0.1287 - val_loss: 0.7393 - val_mae: 0.7028 -
123ms/epoch - 11ms/step
Epoch 97/100
11/11 - 0s - loss: 0.0450 - mae: 0.1356 - val_loss: 0.8226 - val_mae: 0.7522 -
121ms/epoch - 11ms/step
Epoch 98/100
11/11 - 0s - loss: 0.0468 - mae: 0.1360 - val_loss: 0.7680 - val_mae: 0.7206 -
122ms/epoch - 11ms/step
Epoch 99/100
11/11 - 0s - loss: 0.0476 - mae: 0.1372 - val_loss: 0.7865 - val_mae: 0.7321 -
128ms/epoch - 12ms/step
Epoch 100/100
11/11 - 0s - loss: 0.0438 - mae: 0.1327 - val_loss: 0.8077 - val_mae: 0.7454 -
122ms/epoch - 11ms/step
```

1.13 Make Predictions

[23]:
```
RNN_Predictions = RNN_Model.predict(X_test)
LSTM_predictions = LSTM_Model.predict(X_test)
```

1.14 Inverse Transform the Values

[24]:
```
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)

[25]:
```
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 28.728600151721597
MAE 21.959531690253588
```

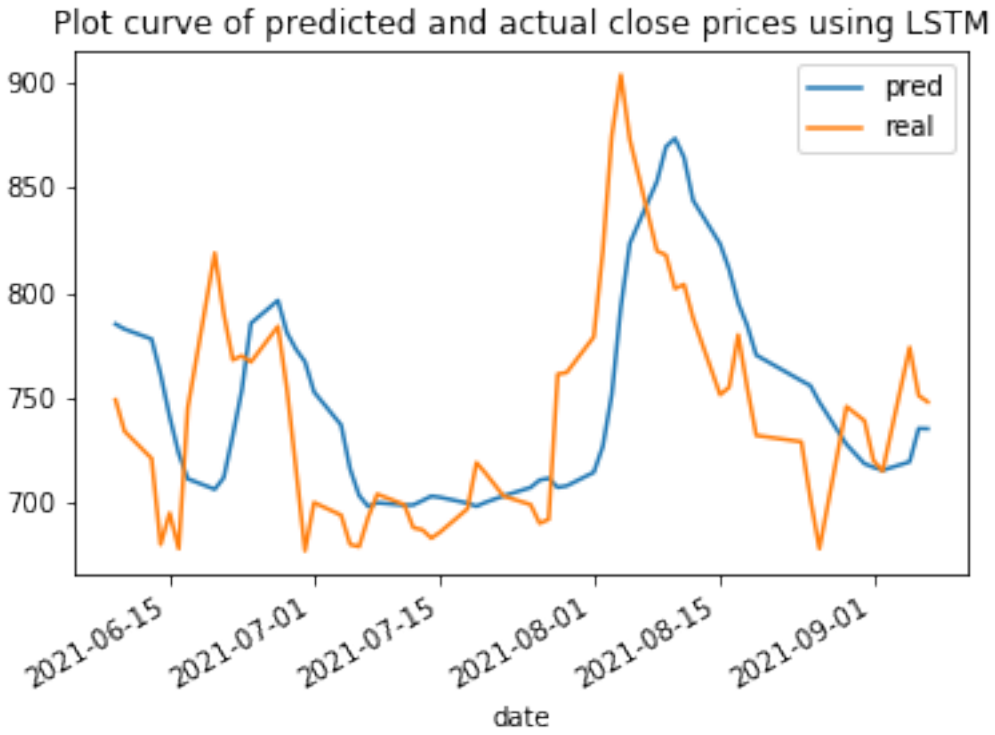
[26]:
```
print("### LSTM Model ###")
Y_test_res_LSTM = std_scalar.inverse_transform(y_test)
pre_LSTM = LSTM_act_prd[:,:1]

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

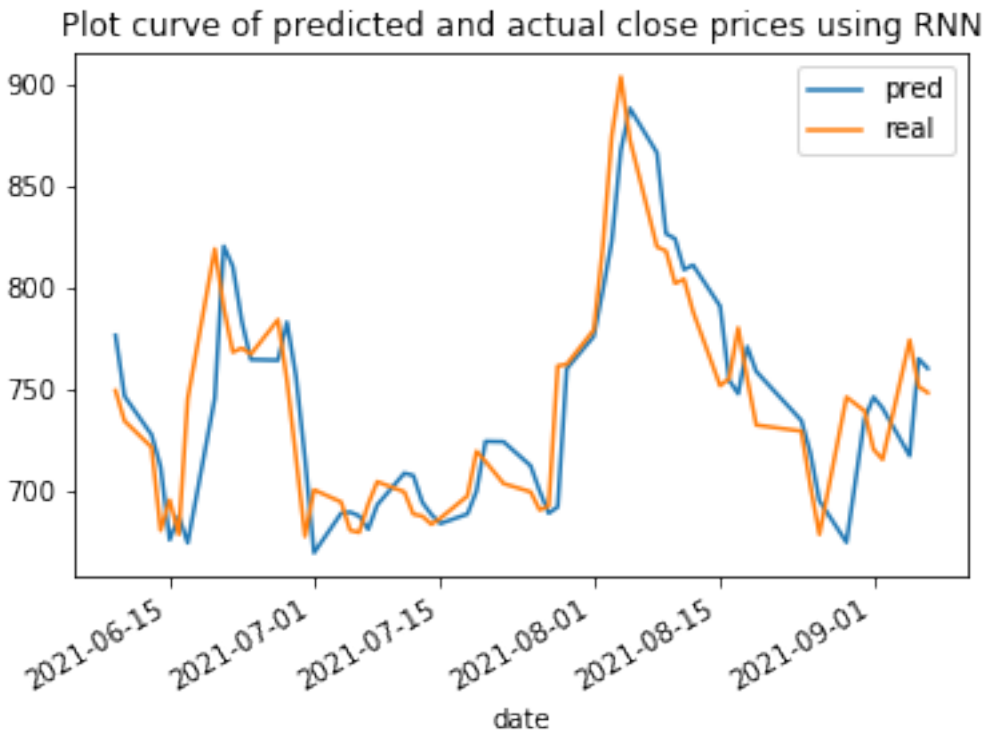
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

```
### LSTM Model ###
RMSE 49.389150590109686
MAE 39.67274450083248
```

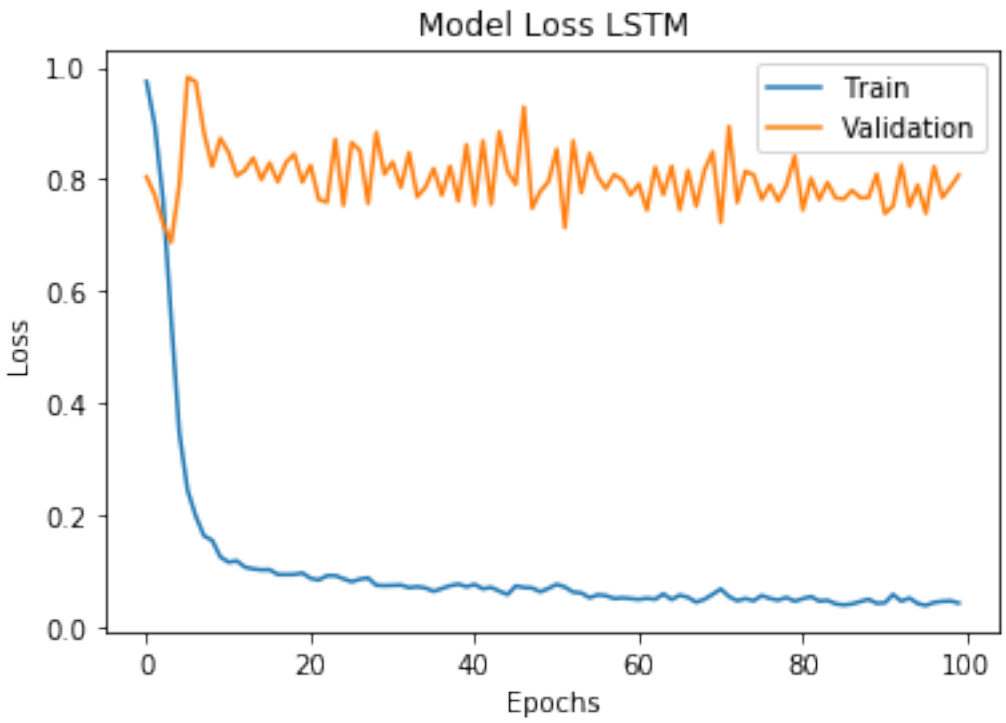
[27]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



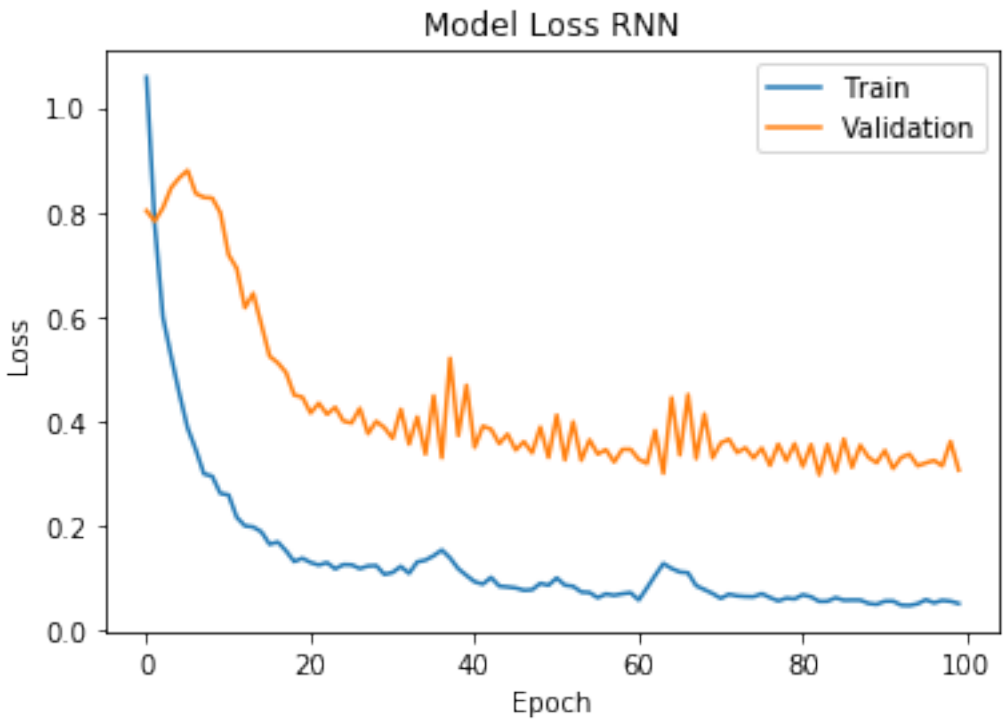
[28]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



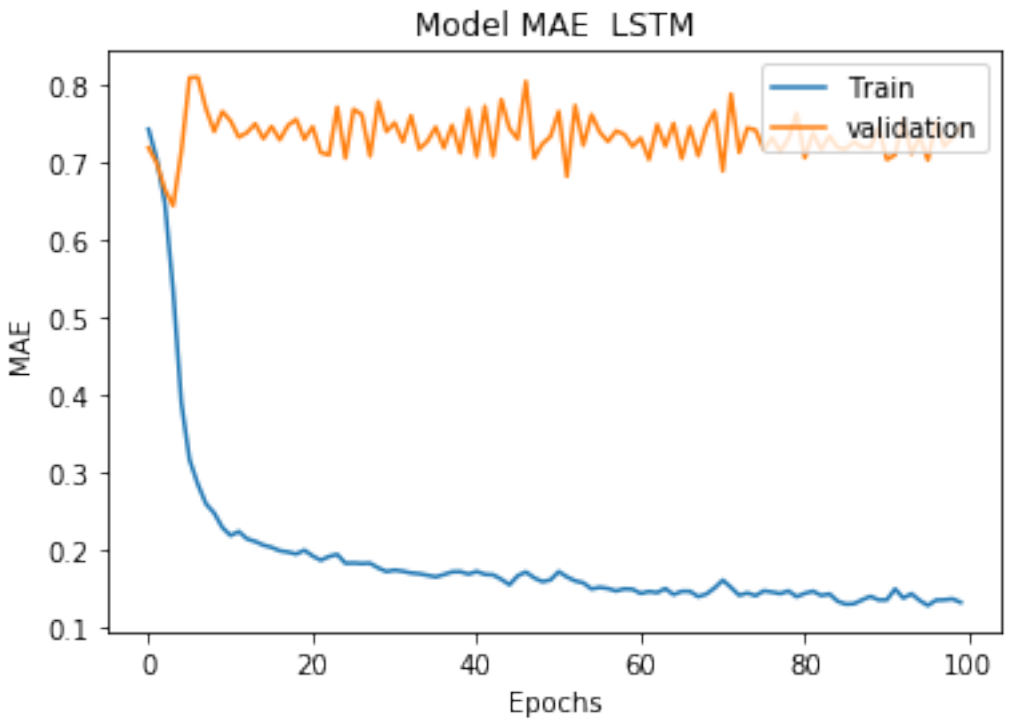
[29]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



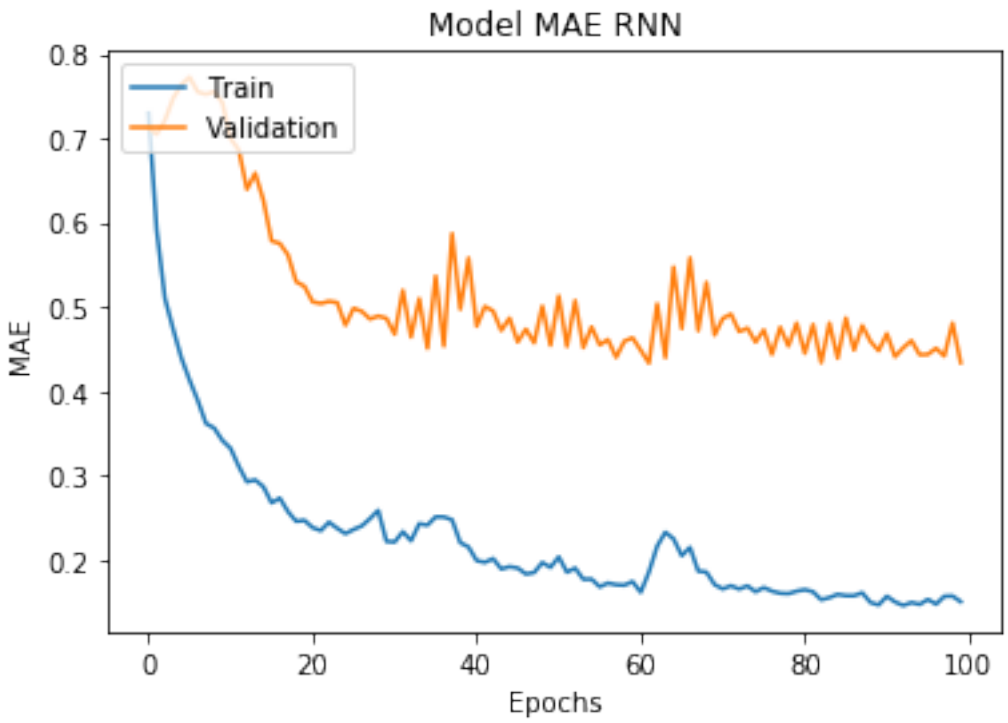
[30]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[31]:
```
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()
```



[32]:
```
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 UPPER LSTM and RNN Models used for Stock Price Prediction
2. The Error is Low for RNN Model