Assignment4_Bank_ADBL

July 30, 2022

1 Stock Price Prediction of ADBL

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 04:57:34.192669: 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 04:57:34.192711: 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]:
```
adbl_df = pd.read_csv("data/ADBL.csv")
adbl_df.shape
```

[3]:
```
(2572, 8)
```

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

[4]:
```
   S.N.        Date  Total Transactions  Total Traded Shares  \
0     1  2021-12-29                1013             200533.0
1     2  2021-12-28                 659              91046.0
2     3  2021-12-27                 816              88858.0
3     4  2021-12-26                1002             130801.0
```

```
4     5  2021-12-23                714             77234.0

   Total Traded Amount  Max. Price  Min. Price  Close Price
0           98526860.8       499.0       488.0        492.0
1           44737396.5       498.0       488.0        494.0
2           44186856.3       510.0       491.0        493.0
3           65306688.9       502.0       495.0        500.0
4           38044401.2       497.0       481.0        496.0
```

1.3 Renaming the Columns

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

[6]:
```python
adbl_df.head()
```

[6]:
```
   SN        Date  TTrans       TTS         TTA  MaxPrice  MinPrice  \
0   1  2021-12-29    1013  200533.0  98526860.8     499.0     488.0
1   2  2021-12-28     659   91046.0  44737396.5     498.0     488.0
2   3  2021-12-27     816   88858.0  44186856.3     510.0     491.0
3   4  2021-12-26    1002  130801.0  65306688.9     502.0     495.0
4   5  2021-12-23     714   77234.0  38044401.2     497.0     481.0

   ClosePrice
0       492.0
1       494.0
2       493.0
3       500.0
4       496.0
```

[7]:
```python
adbl_df.shape
```

[7]:
```
(2572, 8)
```

Converting the Date into Panda's Date Time

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

1.4 Sorting the Date by Date in Ascending Order

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

1.5 Setting Features and Target Column

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

[11]:
```python
X = adbl_df[features]
```

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

1.6 Splitting the Data Into Training, Validation and Test Set

[13]:
```python
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]:
```python
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]:
```python
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]:
```python
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]:
```python
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 04:57:36.425982: 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 04:57:36.426040: W
tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit:
UNKNOWN ERROR (303)
2022-07-30 04:57:36.426075: 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 04:57:36.426481: 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
42/42 - 2s - loss: 0.8075 - mae: 0.6293 - val_loss: 0.2648 - val_mae: 0.3807 -
2s/epoch - 49ms/step
Epoch 2/100
42/42 - 0s - loss: 0.4086 - mae: 0.4325 - val_loss: 0.1636 - val_mae: 0.3003 -
275ms/epoch - 7ms/step
Epoch 3/100
42/42 - 0s - loss: 0.2826 - mae: 0.3585 - val_loss: 0.1417 - val_mae: 0.2838 -
247ms/epoch - 6ms/step
Epoch 4/100
42/42 - 0s - loss: 0.2341 - mae: 0.3232 - val_loss: 0.1184 - val_mae: 0.2571 -
288ms/epoch - 7ms/step
Epoch 5/100
42/42 - 0s - loss: 0.1906 - mae: 0.2914 - val_loss: 0.0920 - val_mae: 0.2185 -
343ms/epoch - 8ms/step
Epoch 6/100
```

```
42/42 - 0s - loss: 0.1630 - mae: 0.2732 - val_loss: 0.0820 - val_mae: 0.2068 -
238ms/epoch - 6ms/step
Epoch 7/100
42/42 - 0s - loss: 0.1471 - mae: 0.2570 - val_loss: 0.0833 - val_mae: 0.2143 -
254ms/epoch - 6ms/step
Epoch 8/100
42/42 - 0s - loss: 0.1337 - mae: 0.2469 - val_loss: 0.0871 - val_mae: 0.2238 -
273ms/epoch - 6ms/step
Epoch 9/100
42/42 - 0s - loss: 0.1184 - mae: 0.2373 - val_loss: 0.0848 - val_mae: 0.2230 -
286ms/epoch - 7ms/step
Epoch 10/100
42/42 - 0s - loss: 0.1112 - mae: 0.2237 - val_loss: 0.0901 - val_mae: 0.2341 -
220ms/epoch - 5ms/step
Epoch 11/100
42/42 - 0s - loss: 0.1120 - mae: 0.2258 - val_loss: 0.0854 - val_mae: 0.2274 -
216ms/epoch - 5ms/step
Epoch 12/100
42/42 - 0s - loss: 0.1056 - mae: 0.2210 - val_loss: 0.1376 - val_mae: 0.3084 -
267ms/epoch - 6ms/step
Epoch 13/100
42/42 - 0s - loss: 0.1112 - mae: 0.2239 - val_loss: 0.1219 - val_mae: 0.2881 -
256ms/epoch - 6ms/step
Epoch 14/100
42/42 - 0s - loss: 0.1011 - mae: 0.2124 - val_loss: 0.1112 - val_mae: 0.2725 -
303ms/epoch - 7ms/step
Epoch 15/100
42/42 - 0s - loss: 0.0975 - mae: 0.2067 - val_loss: 0.1242 - val_mae: 0.2918 -
222ms/epoch - 5ms/step
Epoch 16/100
42/42 - 0s - loss: 0.0995 - mae: 0.2130 - val_loss: 0.1393 - val_mae: 0.3125 -
282ms/epoch - 7ms/step
Epoch 17/100
42/42 - 0s - loss: 0.0974 - mae: 0.2114 - val_loss: 0.0813 - val_mae: 0.2244 -
252ms/epoch - 6ms/step
Epoch 18/100
42/42 - 0s - loss: 0.0821 - mae: 0.1862 - val_loss: 0.0591 - val_mae: 0.1791 -
361ms/epoch - 9ms/step
Epoch 19/100
42/42 - 0s - loss: 0.0785 - mae: 0.1845 - val_loss: 0.0882 - val_mae: 0.2373 -
223ms/epoch - 5ms/step
Epoch 20/100
42/42 - 0s - loss: 0.0779 - mae: 0.1869 - val_loss: 0.0923 - val_mae: 0.2447 -
279ms/epoch - 7ms/step
Epoch 21/100
42/42 - 0s - loss: 0.0725 - mae: 0.1819 - val_loss: 0.0624 - val_mae: 0.1879 -
219ms/epoch - 5ms/step
Epoch 22/100
```

```
42/42 - 0s - loss: 0.0697 - mae: 0.1739 - val_loss: 0.0751 - val_mae: 0.2128 -
219ms/epoch - 5ms/step
Epoch 23/100
42/42 - 0s - loss: 0.0654 - mae: 0.1740 - val_loss: 0.0852 - val_mae: 0.2309 -
288ms/epoch - 7ms/step
Epoch 24/100
42/42 - 0s - loss: 0.0768 - mae: 0.1865 - val_loss: 0.1291 - val_mae: 0.3000 -
228ms/epoch - 5ms/step
Epoch 25/100
42/42 - 0s - loss: 0.0789 - mae: 0.1884 - val_loss: 0.0656 - val_mae: 0.1946 -
321ms/epoch - 8ms/step
Epoch 26/100
42/42 - 0s - loss: 0.0724 - mae: 0.1798 - val_loss: 0.1135 - val_mae: 0.2770 -
238ms/epoch - 6ms/step
Epoch 27/100
42/42 - 0s - loss: 0.0723 - mae: 0.1831 - val_loss: 0.0977 - val_mae: 0.2534 -
272ms/epoch - 6ms/step
Epoch 28/100
42/42 - 0s - loss: 0.0719 - mae: 0.1814 - val_loss: 0.1113 - val_mae: 0.2746 -
256ms/epoch - 6ms/step
Epoch 29/100
42/42 - 0s - loss: 0.0672 - mae: 0.1723 - val_loss: 0.0609 - val_mae: 0.1837 -
305ms/epoch - 7ms/step
Epoch 30/100
42/42 - 0s - loss: 0.0649 - mae: 0.1655 - val_loss: 0.0740 - val_mae: 0.2121 -
263ms/epoch - 6ms/step
Epoch 31/100
42/42 - 0s - loss: 0.0617 - mae: 0.1654 - val_loss: 0.0771 - val_mae: 0.2183 -
289ms/epoch - 7ms/step
Epoch 32/100
42/42 - 0s - loss: 0.0676 - mae: 0.1715 - val_loss: 0.0963 - val_mae: 0.2513 -
225ms/epoch - 5ms/step
Epoch 33/100
42/42 - 0s - loss: 0.0648 - mae: 0.1726 - val_loss: 0.1132 - val_mae: 0.2769 -
273ms/epoch - 6ms/step
Epoch 34/100
42/42 - 0s - loss: 0.0703 - mae: 0.1761 - val_loss: 0.1236 - val_mae: 0.2909 -
224ms/epoch - 5ms/step
Epoch 35/100
42/42 - 0s - loss: 0.0699 - mae: 0.1793 - val_loss: 0.0898 - val_mae: 0.2405 -
227ms/epoch - 5ms/step
Epoch 36/100
42/42 - 0s - loss: 0.0679 - mae: 0.1711 - val_loss: 0.1248 - val_mae: 0.2935 -
228ms/epoch - 5ms/step
Epoch 37/100
42/42 - 0s - loss: 0.0598 - mae: 0.1655 - val_loss: 0.0714 - val_mae: 0.2065 -
300ms/epoch - 7ms/step
Epoch 38/100
```

```
42/42 - 0s - loss: 0.0589 - mae: 0.1596 - val_loss: 0.0953 - val_mae: 0.2477 -
224ms/epoch - 5ms/step
Epoch 39/100
42/42 - 0s - loss: 0.0683 - mae: 0.1746 - val_loss: 0.0912 - val_mae: 0.2415 -
232ms/epoch - 6ms/step
Epoch 40/100
42/42 - 0s - loss: 0.0600 - mae: 0.1661 - val_loss: 0.0976 - val_mae: 0.2528 -
268ms/epoch - 6ms/step
Epoch 41/100
42/42 - 0s - loss: 0.0564 - mae: 0.1563 - val_loss: 0.0790 - val_mae: 0.2201 -
318ms/epoch - 8ms/step
Epoch 42/100
42/42 - 0s - loss: 0.0624 - mae: 0.1646 - val_loss: 0.1145 - val_mae: 0.2764 -
263ms/epoch - 6ms/step
Epoch 43/100
42/42 - 0s - loss: 0.0707 - mae: 0.1788 - val_loss: 0.1467 - val_mae: 0.3189 -
237ms/epoch - 6ms/step
Epoch 44/100
42/42 - 0s - loss: 0.0705 - mae: 0.1774 - val_loss: 0.0975 - val_mae: 0.2512 -
307ms/epoch - 7ms/step
Epoch 45/100
42/42 - 0s - loss: 0.0636 - mae: 0.1647 - val_loss: 0.1041 - val_mae: 0.2605 -
297ms/epoch - 7ms/step
Epoch 46/100
42/42 - 0s - loss: 0.0627 - mae: 0.1662 - val_loss: 0.0955 - val_mae: 0.2477 -
227ms/epoch - 5ms/step
Epoch 47/100
42/42 - 0s - loss: 0.0594 - mae: 0.1583 - val_loss: 0.0628 - val_mae: 0.1875 -
283ms/epoch - 7ms/step
Epoch 48/100
42/42 - 0s - loss: 0.0546 - mae: 0.1546 - val_loss: 0.1093 - val_mae: 0.2687 -
228ms/epoch - 5ms/step
Epoch 49/100
42/42 - 0s - loss: 0.0610 - mae: 0.1631 - val_loss: 0.0703 - val_mae: 0.2043 -
271ms/epoch - 6ms/step
Epoch 50/100
42/42 - 0s - loss: 0.0527 - mae: 0.1494 - val_loss: 0.0993 - val_mae: 0.2528 -
295ms/epoch - 7ms/step
Epoch 51/100
42/42 - 0s - loss: 0.0585 - mae: 0.1604 - val_loss: 0.0736 - val_mae: 0.2095 -
297ms/epoch - 7ms/step
Epoch 52/100
42/42 - 0s - loss: 0.0520 - mae: 0.1503 - val_loss: 0.0850 - val_mae: 0.2292 -
237ms/epoch - 6ms/step
Epoch 53/100
42/42 - 0s - loss: 0.0529 - mae: 0.1543 - val_loss: 0.0839 - val_mae: 0.2282 -
219ms/epoch - 5ms/step
Epoch 54/100
```

```
42/42 - 0s - loss: 0.0564 - mae: 0.1560 - val_loss: 0.1093 - val_mae: 0.2680 -
254ms/epoch - 6ms/step
Epoch 55/100
42/42 - 0s - loss: 0.0553 - mae: 0.1585 - val_loss: 0.0738 - val_mae: 0.2096 -
245ms/epoch - 6ms/step
Epoch 56/100
42/42 - 0s - loss: 0.0567 - mae: 0.1520 - val_loss: 0.0758 - val_mae: 0.2134 -
217ms/epoch - 5ms/step
Epoch 57/100
42/42 - 0s - loss: 0.0594 - mae: 0.1611 - val_loss: 0.1260 - val_mae: 0.2922 -
256ms/epoch - 6ms/step
Epoch 58/100
42/42 - 0s - loss: 0.0572 - mae: 0.1625 - val_loss: 0.1079 - val_mae: 0.2651 -
284ms/epoch - 7ms/step
Epoch 59/100
42/42 - 0s - loss: 0.0604 - mae: 0.1650 - val_loss: 0.1356 - val_mae: 0.3049 -
349ms/epoch - 8ms/step
Epoch 60/100
42/42 - 0s - loss: 0.0589 - mae: 0.1639 - val_loss: 0.0948 - val_mae: 0.2458 -
324ms/epoch - 8ms/step
Epoch 61/100
42/42 - 0s - loss: 0.0512 - mae: 0.1462 - val_loss: 0.0825 - val_mae: 0.2248 -
276ms/epoch - 7ms/step
Epoch 62/100
42/42 - 0s - loss: 0.0537 - mae: 0.1498 - val_loss: 0.0810 - val_mae: 0.2213 -
263ms/epoch - 6ms/step
Epoch 63/100
42/42 - 0s - loss: 0.0502 - mae: 0.1451 - val_loss: 0.0860 - val_mae: 0.2301 -
238ms/epoch - 6ms/step
Epoch 64/100
42/42 - 0s - loss: 0.0588 - mae: 0.1564 - val_loss: 0.0764 - val_mae: 0.2116 -
345ms/epoch - 8ms/step
Epoch 65/100
42/42 - 0s - loss: 0.0539 - mae: 0.1564 - val_loss: 0.1206 - val_mae: 0.2810 -
230ms/epoch - 5ms/step
Epoch 66/100
42/42 - 0s - loss: 0.0541 - mae: 0.1533 - val_loss: 0.1026 - val_mae: 0.2563 -
225ms/epoch - 5ms/step
Epoch 67/100
42/42 - 0s - loss: 0.0551 - mae: 0.1548 - val_loss: 0.0763 - val_mae: 0.2138 -
310ms/epoch - 7ms/step
Epoch 68/100
42/42 - 0s - loss: 0.0513 - mae: 0.1490 - val_loss: 0.1305 - val_mae: 0.2930 -
399ms/epoch - 10ms/step
Epoch 69/100
42/42 - 0s - loss: 0.0616 - mae: 0.1630 - val_loss: 0.1006 - val_mae: 0.2500 -
313ms/epoch - 7ms/step
Epoch 70/100
```

```
42/42 - 0s - loss: 0.0519 - mae: 0.1496 - val_loss: 0.0879 - val_mae: 0.2332 -
218ms/epoch - 5ms/step
Epoch 71/100
42/42 - 0s - loss: 0.0555 - mae: 0.1574 - val_loss: 0.1275 - val_mae: 0.2906 -
222ms/epoch - 5ms/step
Epoch 72/100
42/42 - 0s - loss: 0.0602 - mae: 0.1663 - val_loss: 0.0850 - val_mae: 0.2292 -
321ms/epoch - 8ms/step
Epoch 73/100
42/42 - 0s - loss: 0.0541 - mae: 0.1517 - val_loss: 0.1002 - val_mae: 0.2526 -
217ms/epoch - 5ms/step
Epoch 74/100
42/42 - 0s - loss: 0.0577 - mae: 0.1570 - val_loss: 0.0695 - val_mae: 0.1979 -
380ms/epoch - 9ms/step
Epoch 75/100
42/42 - 0s - loss: 0.0459 - mae: 0.1369 - val_loss: 0.0626 - val_mae: 0.1862 -
269ms/epoch - 6ms/step
Epoch 76/100
42/42 - 0s - loss: 0.0442 - mae: 0.1351 - val_loss: 0.0609 - val_mae: 0.1798 -
220ms/epoch - 5ms/step
Epoch 77/100
42/42 - 0s - loss: 0.0407 - mae: 0.1303 - val_loss: 0.0485 - val_mae: 0.1497 -
271ms/epoch - 6ms/step
Epoch 78/100
42/42 - 0s - loss: 0.0457 - mae: 0.1329 - val_loss: 0.0734 - val_mae: 0.2065 -
254ms/epoch - 6ms/step
Epoch 79/100
42/42 - 0s - loss: 0.0471 - mae: 0.1400 - val_loss: 0.0607 - val_mae: 0.1818 -
215ms/epoch - 5ms/step
Epoch 80/100
42/42 - 0s - loss: 0.0439 - mae: 0.1340 - val_loss: 0.0906 - val_mae: 0.2352 -
210ms/epoch - 5ms/step
Epoch 81/100
42/42 - 0s - loss: 0.0530 - mae: 0.1511 - val_loss: 0.1466 - val_mae: 0.3144 -
256ms/epoch - 6ms/step
Epoch 82/100
42/42 - 0s - loss: 0.0537 - mae: 0.1553 - val_loss: 0.0701 - val_mae: 0.1976 -
252ms/epoch - 6ms/step
Epoch 83/100
42/42 - 0s - loss: 0.0496 - mae: 0.1447 - val_loss: 0.1262 - val_mae: 0.2855 -
259ms/epoch - 6ms/step
Epoch 84/100
42/42 - 0s - loss: 0.0631 - mae: 0.1656 - val_loss: 0.1340 - val_mae: 0.2971 -
214ms/epoch - 5ms/step
Epoch 85/100
42/42 - 0s - loss: 0.0603 - mae: 0.1680 - val_loss: 0.0857 - val_mae: 0.2281 -
210ms/epoch - 5ms/step
Epoch 86/100
```

```
42/42 - 0s - loss: 0.0505 - mae: 0.1480 - val_loss: 0.0890 - val_mae: 0.2319 -
212ms/epoch - 5ms/step
Epoch 87/100
42/42 - 0s - loss: 0.0509 - mae: 0.1474 - val_loss: 0.0762 - val_mae: 0.2104 -
251ms/epoch - 6ms/step
Epoch 88/100
42/42 - 0s - loss: 0.0448 - mae: 0.1363 - val_loss: 0.0572 - val_mae: 0.1693 -
221ms/epoch - 5ms/step
Epoch 89/100
42/42 - 0s - loss: 0.0437 - mae: 0.1307 - val_loss: 0.0732 - val_mae: 0.2034 -
271ms/epoch - 6ms/step
Epoch 90/100
42/42 - 0s - loss: 0.0480 - mae: 0.1410 - val_loss: 0.0553 - val_mae: 0.1624 -
262ms/epoch - 6ms/step
Epoch 91/100
42/42 - 0s - loss: 0.0451 - mae: 0.1371 - val_loss: 0.0930 - val_mae: 0.2376 -
224ms/epoch - 5ms/step
Epoch 92/100
42/42 - 0s - loss: 0.0522 - mae: 0.1537 - val_loss: 0.1386 - val_mae: 0.3004 -
242ms/epoch - 6ms/step
Epoch 93/100
42/42 - 0s - loss: 0.0549 - mae: 0.1609 - val_loss: 0.0532 - val_mae: 0.1588 -
217ms/epoch - 5ms/step
Epoch 94/100
42/42 - 0s - loss: 0.0487 - mae: 0.1405 - val_loss: 0.1493 - val_mae: 0.3150 -
250ms/epoch - 6ms/step
Epoch 95/100
42/42 - 0s - loss: 0.0595 - mae: 0.1630 - val_loss: 0.0817 - val_mae: 0.2185 -
257ms/epoch - 6ms/step
Epoch 96/100
42/42 - 0s - loss: 0.0499 - mae: 0.1446 - val_loss: 0.1013 - val_mae: 0.2541 -
208ms/epoch - 5ms/step
Epoch 97/100
42/42 - 0s - loss: 0.0525 - mae: 0.1503 - val_loss: 0.1210 - val_mae: 0.2775 -
254ms/epoch - 6ms/step
Epoch 98/100
42/42 - 0s - loss: 0.0551 - mae: 0.1554 - val_loss: 0.1442 - val_mae: 0.3085 -
209ms/epoch - 5ms/step
Epoch 99/100
42/42 - 0s - loss: 0.0596 - mae: 0.1659 - val_loss: 0.0732 - val_mae: 0.2066 -
210ms/epoch - 5ms/step
Epoch 100/100
42/42 - 0s - loss: 0.0488 - mae: 0.1421 - val_loss: 0.0734 - val_mae: 0.2074 -
211ms/epoch - 5ms/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
42/42 - 5s - loss: 0.8692 - mae: 0.7307 - val_loss: 0.4351 - val_mae: 0.5097 -
5s/epoch - 125ms/step
Epoch 2/100
42/42 - 0s - loss: 0.3558 - mae: 0.3989 - val_loss: 0.2074 - val_mae: 0.3397 -
372ms/epoch - 9ms/step
Epoch 3/100
42/42 - 0s - loss: 0.1888 - mae: 0.2754 - val_loss: 0.1902 - val_mae: 0.3285 -
371ms/epoch - 9ms/step
Epoch 4/100
42/42 - 0s - loss: 0.1625 - mae: 0.2759 - val_loss: 0.2000 - val_mae: 0.3343 -
371ms/epoch - 9ms/step
Epoch 5/100
42/42 - 0s - loss: 0.1468 - mae: 0.2659 - val_loss: 0.1914 - val_mae: 0.3263 -
370ms/epoch - 9ms/step
Epoch 6/100
42/42 - 0s - loss: 0.1317 - mae: 0.2495 - val_loss: 0.1763 - val_mae: 0.3101 -
378ms/epoch - 9ms/step
Epoch 7/100
42/42 - 0s - loss: 0.1140 - mae: 0.2321 - val_loss: 0.1614 - val_mae: 0.2922 -
368ms/epoch - 9ms/step
Epoch 8/100
42/42 - 0s - loss: 0.1004 - mae: 0.2158 - val_loss: 0.1525 - val_mae: 0.2817 -
431ms/epoch - 10ms/step
Epoch 9/100
42/42 - 0s - loss: 0.0891 - mae: 0.2035 - val_loss: 0.1404 - val_mae: 0.2659 -
400ms/epoch - 10ms/step
Epoch 10/100
42/42 - 0s - loss: 0.0812 - mae: 0.1875 - val_loss: 0.1480 - val_mae: 0.2758 -
485ms/epoch - 12ms/step
Epoch 11/100
42/42 - 1s - loss: 0.0760 - mae: 0.1822 - val_loss: 0.1336 - val_mae: 0.2640 -
572ms/epoch - 14ms/step
Epoch 12/100
42/42 - 0s - loss: 0.0740 - mae: 0.1807 - val_loss: 0.1337 - val_mae: 0.2590 -
384ms/epoch - 9ms/step
Epoch 13/100
42/42 - 0s - loss: 0.0652 - mae: 0.1657 - val_loss: 0.1311 - val_mae: 0.2549 -
385ms/epoch - 9ms/step
Epoch 14/100
42/42 - 0s - loss: 0.0683 - mae: 0.1681 - val_loss: 0.1275 - val_mae: 0.2521 -
385ms/epoch - 9ms/step
Epoch 15/100
42/42 - 0s - loss: 0.0628 - mae: 0.1627 - val_loss: 0.1258 - val_mae: 0.2516 -
372ms/epoch - 9ms/step
```

```
Epoch 16/100
42/42 - 0s - loss: 0.0606 - mae: 0.1591 - val_loss: 0.1273 - val_mae: 0.2519 -
389ms/epoch - 9ms/step
Epoch 17/100
42/42 - 1s - loss: 0.0640 - mae: 0.1637 - val_loss: 0.1308 - val_mae: 0.2593 -
517ms/epoch - 12ms/step
Epoch 18/100
42/42 - 0s - loss: 0.0621 - mae: 0.1616 - val_loss: 0.1259 - val_mae: 0.2558 -
386ms/epoch - 9ms/step
Epoch 19/100
42/42 - 0s - loss: 0.0634 - mae: 0.1640 - val_loss: 0.1311 - val_mae: 0.2591 -
370ms/epoch - 9ms/step
Epoch 20/100
42/42 - 0s - loss: 0.0579 - mae: 0.1583 - val_loss: 0.1437 - val_mae: 0.2809 -
382ms/epoch - 9ms/step
Epoch 21/100
42/42 - 0s - loss: 0.0674 - mae: 0.1744 - val_loss: 0.1399 - val_mae: 0.2668 -
375ms/epoch - 9ms/step
Epoch 22/100
42/42 - 0s - loss: 0.0565 - mae: 0.1531 - val_loss: 0.1297 - val_mae: 0.2594 -
386ms/epoch - 9ms/step
Epoch 23/100
42/42 - 0s - loss: 0.0545 - mae: 0.1538 - val_loss: 0.1245 - val_mae: 0.2522 -
380ms/epoch - 9ms/step
Epoch 24/100
42/42 - 0s - loss: 0.0576 - mae: 0.1590 - val_loss: 0.1259 - val_mae: 0.2566 -
371ms/epoch - 9ms/step
Epoch 25/100
42/42 - 0s - loss: 0.0565 - mae: 0.1556 - val_loss: 0.1202 - val_mae: 0.2487 -
383ms/epoch - 9ms/step
Epoch 26/100
42/42 - 0s - loss: 0.0572 - mae: 0.1561 - val_loss: 0.1546 - val_mae: 0.2905 -
382ms/epoch - 9ms/step
Epoch 27/100
42/42 - 0s - loss: 0.0577 - mae: 0.1609 - val_loss: 0.1260 - val_mae: 0.2626 -
386ms/epoch - 9ms/step
Epoch 28/100
42/42 - 0s - loss: 0.0541 - mae: 0.1580 - val_loss: 0.1469 - val_mae: 0.2800 -
370ms/epoch - 9ms/step
Epoch 29/100
42/42 - 0s - loss: 0.0522 - mae: 0.1496 - val_loss: 0.1236 - val_mae: 0.2584 -
378ms/epoch - 9ms/step
Epoch 30/100
42/42 - 0s - loss: 0.0559 - mae: 0.1559 - val_loss: 0.1472 - val_mae: 0.2797 -
376ms/epoch - 9ms/step
Epoch 31/100
42/42 - 0s - loss: 0.0512 - mae: 0.1455 - val_loss: 0.1177 - val_mae: 0.2497 -
371ms/epoch - 9ms/step
```

```
Epoch 32/100
42/42 - 0s - loss: 0.0521 - mae: 0.1519 - val_loss: 0.1460 - val_mae: 0.2809 -
376ms/epoch - 9ms/step
Epoch 33/100
42/42 - 0s - loss: 0.0589 - mae: 0.1582 - val_loss: 0.1470 - val_mae: 0.2882 -
386ms/epoch - 9ms/step
Epoch 34/100
42/42 - 0s - loss: 0.0526 - mae: 0.1528 - val_loss: 0.1194 - val_mae: 0.2523 -
375ms/epoch - 9ms/step
Epoch 35/100
42/42 - 0s - loss: 0.0524 - mae: 0.1517 - val_loss: 0.1161 - val_mae: 0.2468 -
377ms/epoch - 9ms/step
Epoch 36/100
42/42 - 0s - loss: 0.0552 - mae: 0.1573 - val_loss: 0.1533 - val_mae: 0.2909 -
388ms/epoch - 9ms/step
Epoch 37/100
42/42 - 0s - loss: 0.0533 - mae: 0.1513 - val_loss: 0.1298 - val_mae: 0.2707 -
371ms/epoch - 9ms/step
Epoch 38/100
42/42 - 0s - loss: 0.0565 - mae: 0.1598 - val_loss: 0.1568 - val_mae: 0.2893 -
372ms/epoch - 9ms/step
Epoch 39/100
42/42 - 0s - loss: 0.0509 - mae: 0.1476 - val_loss: 0.1118 - val_mae: 0.2438 -
384ms/epoch - 9ms/step
Epoch 40/100
42/42 - 1s - loss: 0.0504 - mae: 0.1449 - val_loss: 0.1160 - val_mae: 0.2419 -
651ms/epoch - 16ms/step
Epoch 41/100
42/42 - 0s - loss: 0.0479 - mae: 0.1380 - val_loss: 0.1070 - val_mae: 0.2332 -
376ms/epoch - 9ms/step
Epoch 42/100
42/42 - 0s - loss: 0.0441 - mae: 0.1385 - val_loss: 0.1326 - val_mae: 0.2624 -
493ms/epoch - 12ms/step
Epoch 43/100
42/42 - 0s - loss: 0.0469 - mae: 0.1405 - val_loss: 0.1158 - val_mae: 0.2501 -
424ms/epoch - 10ms/step
Epoch 44/100
42/42 - 1s - loss: 0.0465 - mae: 0.1447 - val_loss: 0.1379 - val_mae: 0.2694 -
627ms/epoch - 15ms/step
Epoch 45/100
42/42 - 0s - loss: 0.0494 - mae: 0.1433 - val_loss: 0.1088 - val_mae: 0.2412 -
420ms/epoch - 10ms/step
Epoch 46/100
42/42 - 0s - loss: 0.0498 - mae: 0.1439 - val_loss: 0.1307 - val_mae: 0.2631 -
411ms/epoch - 10ms/step
Epoch 47/100
42/42 - 0s - loss: 0.0439 - mae: 0.1387 - val_loss: 0.1108 - val_mae: 0.2448 -
371ms/epoch - 9ms/step
```

```
Epoch 48/100
42/42 - 0s - loss: 0.0462 - mae: 0.1437 - val_loss: 0.1026 - val_mae: 0.2265 -
369ms/epoch - 9ms/step
Epoch 49/100
42/42 - 0s - loss: 0.0456 - mae: 0.1344 - val_loss: 0.1013 - val_mae: 0.2250 -
384ms/epoch - 9ms/step
Epoch 50/100
42/42 - 0s - loss: 0.0453 - mae: 0.1386 - val_loss: 0.1118 - val_mae: 0.2403 -
382ms/epoch - 9ms/step
Epoch 51/100
42/42 - 0s - loss: 0.0398 - mae: 0.1305 - val_loss: 0.1034 - val_mae: 0.2331 -
389ms/epoch - 9ms/step
Epoch 52/100
42/42 - 1s - loss: 0.0440 - mae: 0.1384 - val_loss: 0.1070 - val_mae: 0.2365 -
528ms/epoch - 13ms/step
Epoch 53/100
42/42 - 0s - loss: 0.0419 - mae: 0.1332 - val_loss: 0.1019 - val_mae: 0.2306 -
423ms/epoch - 10ms/step
Epoch 54/100
42/42 - 0s - loss: 0.0435 - mae: 0.1381 - val_loss: 0.1125 - val_mae: 0.2453 -
363ms/epoch - 9ms/step
Epoch 55/100
42/42 - 0s - loss: 0.0435 - mae: 0.1368 - val_loss: 0.1190 - val_mae: 0.2604 -
365ms/epoch - 9ms/step
Epoch 56/100
42/42 - 0s - loss: 0.0484 - mae: 0.1476 - val_loss: 0.1176 - val_mae: 0.2476 -
365ms/epoch - 9ms/step
Epoch 57/100
42/42 - 0s - loss: 0.0426 - mae: 0.1326 - val_loss: 0.1081 - val_mae: 0.2432 -
358ms/epoch - 9ms/step
Epoch 58/100
42/42 - 0s - loss: 0.0465 - mae: 0.1405 - val_loss: 0.1326 - val_mae: 0.2658 -
357ms/epoch - 8ms/step
Epoch 59/100
42/42 - 0s - loss: 0.0454 - mae: 0.1366 - val_loss: 0.1125 - val_mae: 0.2477 -
366ms/epoch - 9ms/step
Epoch 60/100
42/42 - 0s - loss: 0.0460 - mae: 0.1406 - val_loss: 0.1643 - val_mae: 0.3002 -
360ms/epoch - 9ms/step
Epoch 61/100
42/42 - 0s - loss: 0.0506 - mae: 0.1511 - val_loss: 0.1484 - val_mae: 0.2959 -
351ms/epoch - 8ms/step
Epoch 62/100
42/42 - 0s - loss: 0.0527 - mae: 0.1537 - val_loss: 0.1269 - val_mae: 0.2691 -
355ms/epoch - 8ms/step
Epoch 63/100
42/42 - 0s - loss: 0.0494 - mae: 0.1452 - val_loss: 0.1013 - val_mae: 0.2335 -
360ms/epoch - 9ms/step
```

```
Epoch 64/100
42/42 - 0s - loss: 0.0468 - mae: 0.1414 - val_loss: 0.1094 - val_mae: 0.2423 -
350ms/epoch - 8ms/step
Epoch 65/100
42/42 - 0s - loss: 0.0422 - mae: 0.1337 - val_loss: 0.1694 - val_mae: 0.3034 -
363ms/epoch - 9ms/step
Epoch 66/100
42/42 - 0s - loss: 0.0510 - mae: 0.1494 - val_loss: 0.1224 - val_mae: 0.2652 -
357ms/epoch - 8ms/step
Epoch 67/100
42/42 - 0s - loss: 0.0465 - mae: 0.1404 - val_loss: 0.1074 - val_mae: 0.2435 -
349ms/epoch - 8ms/step
Epoch 68/100
42/42 - 0s - loss: 0.0484 - mae: 0.1425 - val_loss: 0.1016 - val_mae: 0.2348 -
350ms/epoch - 8ms/step
Epoch 69/100
42/42 - 0s - loss: 0.0426 - mae: 0.1348 - val_loss: 0.0984 - val_mae: 0.2230 -
355ms/epoch - 8ms/step
Epoch 70/100
42/42 - 0s - loss: 0.0381 - mae: 0.1235 - val_loss: 0.0901 - val_mae: 0.2112 -
351ms/epoch - 8ms/step
Epoch 71/100
42/42 - 0s - loss: 0.0427 - mae: 0.1338 - val_loss: 0.1776 - val_mae: 0.3060 -
351ms/epoch - 8ms/step
Epoch 72/100
42/42 - 0s - loss: 0.0428 - mae: 0.1366 - val_loss: 0.1120 - val_mae: 0.2524 -
356ms/epoch - 8ms/step
Epoch 73/100
42/42 - 0s - loss: 0.0447 - mae: 0.1414 - val_loss: 0.1029 - val_mae: 0.2294 -
351ms/epoch - 8ms/step
Epoch 74/100
42/42 - 0s - loss: 0.0415 - mae: 0.1263 - val_loss: 0.0907 - val_mae: 0.2125 -
351ms/epoch - 8ms/step
Epoch 75/100
42/42 - 0s - loss: 0.0432 - mae: 0.1335 - val_loss: 0.0966 - val_mae: 0.2257 -
357ms/epoch - 9ms/step
Epoch 76/100
42/42 - 0s - loss: 0.0396 - mae: 0.1320 - val_loss: 0.0901 - val_mae: 0.2151 -
374ms/epoch - 9ms/step
Epoch 77/100
42/42 - 0s - loss: 0.0412 - mae: 0.1271 - val_loss: 0.0874 - val_mae: 0.2073 -
362ms/epoch - 9ms/step
Epoch 78/100
42/42 - 0s - loss: 0.0379 - mae: 0.1265 - val_loss: 0.1397 - val_mae: 0.2742 -
350ms/epoch - 8ms/step
Epoch 79/100
42/42 - 0s - loss: 0.0470 - mae: 0.1407 - val_loss: 0.1000 - val_mae: 0.2359 -
363ms/epoch - 9ms/step
```

```
Epoch 80/100
42/42 - 0s - loss: 0.0485 - mae: 0.1459 - val_loss: 0.1013 - val_mae: 0.2347 -
349ms/epoch - 8ms/step
Epoch 81/100
42/42 - 1s - loss: 0.0420 - mae: 0.1383 - val_loss: 0.1186 - val_mae: 0.2607 -
516ms/epoch - 12ms/step
Epoch 82/100
42/42 - 0s - loss: 0.0469 - mae: 0.1440 - val_loss: 0.1361 - val_mae: 0.2771 -
352ms/epoch - 8ms/step
Epoch 83/100
42/42 - 0s - loss: 0.0411 - mae: 0.1307 - val_loss: 0.0938 - val_mae: 0.2245 -
379ms/epoch - 9ms/step
Epoch 84/100
42/42 - 0s - loss: 0.0420 - mae: 0.1354 - val_loss: 0.0874 - val_mae: 0.2084 -
411ms/epoch - 10ms/step
Epoch 85/100
42/42 - 0s - loss: 0.0385 - mae: 0.1290 - val_loss: 0.1012 - val_mae: 0.2367 -
385ms/epoch - 9ms/step
Epoch 86/100
42/42 - 0s - loss: 0.0464 - mae: 0.1437 - val_loss: 0.1224 - val_mae: 0.2571 -
373ms/epoch - 9ms/step
Epoch 87/100
42/42 - 0s - loss: 0.0398 - mae: 0.1302 - val_loss: 0.0900 - val_mae: 0.2185 -
384ms/epoch - 9ms/step
Epoch 88/100
42/42 - 0s - loss: 0.0401 - mae: 0.1350 - val_loss: 0.0995 - val_mae: 0.2300 -
370ms/epoch - 9ms/step
Epoch 89/100
42/42 - 0s - loss: 0.0405 - mae: 0.1285 - val_loss: 0.0875 - val_mae: 0.2137 -
373ms/epoch - 9ms/step
Epoch 90/100
42/42 - 0s - loss: 0.0441 - mae: 0.1332 - val_loss: 0.1189 - val_mae: 0.2535 -
351ms/epoch - 8ms/step
Epoch 91/100
42/42 - 0s - loss: 0.0374 - mae: 0.1260 - val_loss: 0.0892 - val_mae: 0.2179 -
368ms/epoch - 9ms/step
Epoch 92/100
42/42 - 0s - loss: 0.0431 - mae: 0.1331 - val_loss: 0.0898 - val_mae: 0.2145 -
364ms/epoch - 9ms/step
Epoch 93/100
42/42 - 0s - loss: 0.0389 - mae: 0.1249 - val_loss: 0.1068 - val_mae: 0.2414 -
372ms/epoch - 9ms/step
Epoch 94/100
42/42 - 0s - loss: 0.0401 - mae: 0.1298 - val_loss: 0.0870 - val_mae: 0.2138 -
374ms/epoch - 9ms/step
Epoch 95/100
42/42 - 0s - loss: 0.0428 - mae: 0.1332 - val_loss: 0.1056 - val_mae: 0.2377 -
416ms/epoch - 10ms/step
```

```
Epoch 96/100
42/42 - 0s - loss: 0.0384 - mae: 0.1267 - val_loss: 0.0891 - val_mae: 0.2188 -
436ms/epoch - 10ms/step
Epoch 97/100
42/42 - 0s - loss: 0.0386 - mae: 0.1293 - val_loss: 0.0826 - val_mae: 0.2028 -
376ms/epoch - 9ms/step
Epoch 98/100
42/42 - 0s - loss: 0.0414 - mae: 0.1253 - val_loss: 0.0843 - val_mae: 0.2059 -
438ms/epoch - 10ms/step
Epoch 99/100
42/42 - 1s - loss: 0.0352 - mae: 0.1198 - val_loss: 0.1013 - val_mae: 0.2341 -
532ms/epoch - 13ms/step
Epoch 100/100
42/42 - 0s - loss: 0.0357 - mae: 0.1258 - val_loss: 0.0899 - val_mae: 0.2210 -
500ms/epoch - 12ms/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 9.715363292220541
MAE 6.335043649824839
```

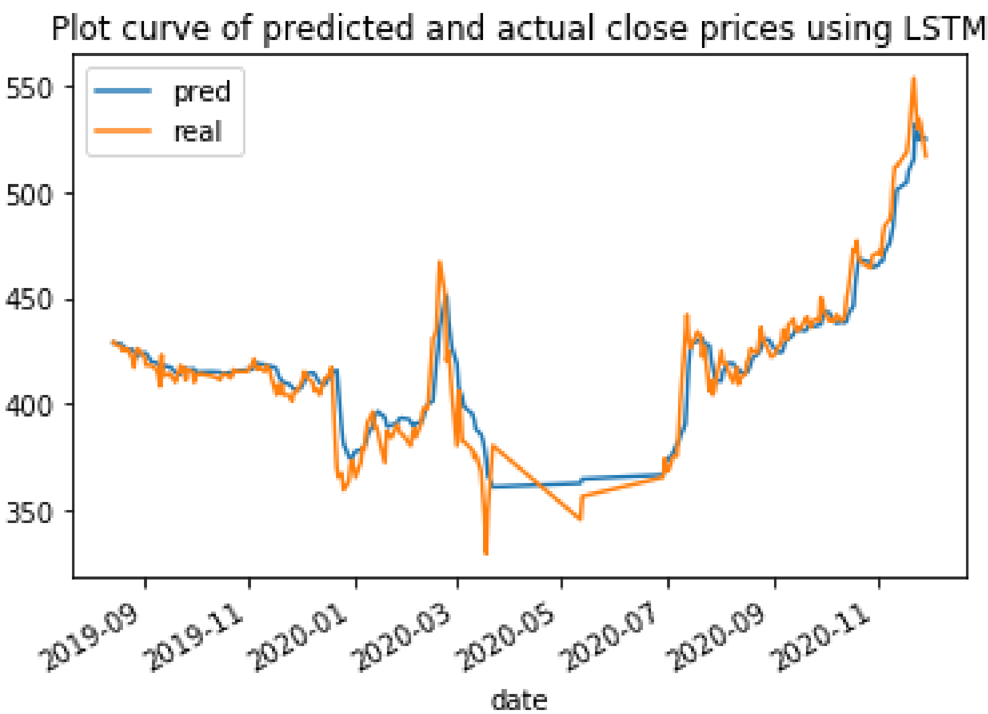
[26]:
```
print("### LSTM Model ###")
Y_test_res_LSTM = std_scalar.inverse_transform(y_test)
pre_LSTM = LSTM_act_prd[:,:1]

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

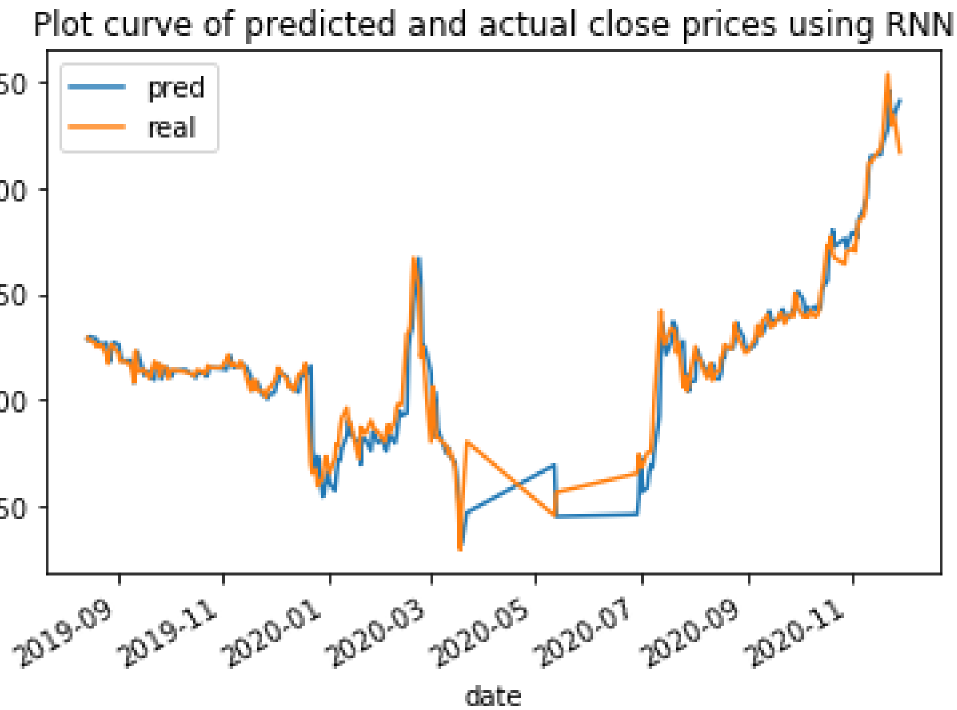
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

```
### LSTM Model ###
RMSE 11.515386909556055
MAE 7.437573387509301
```

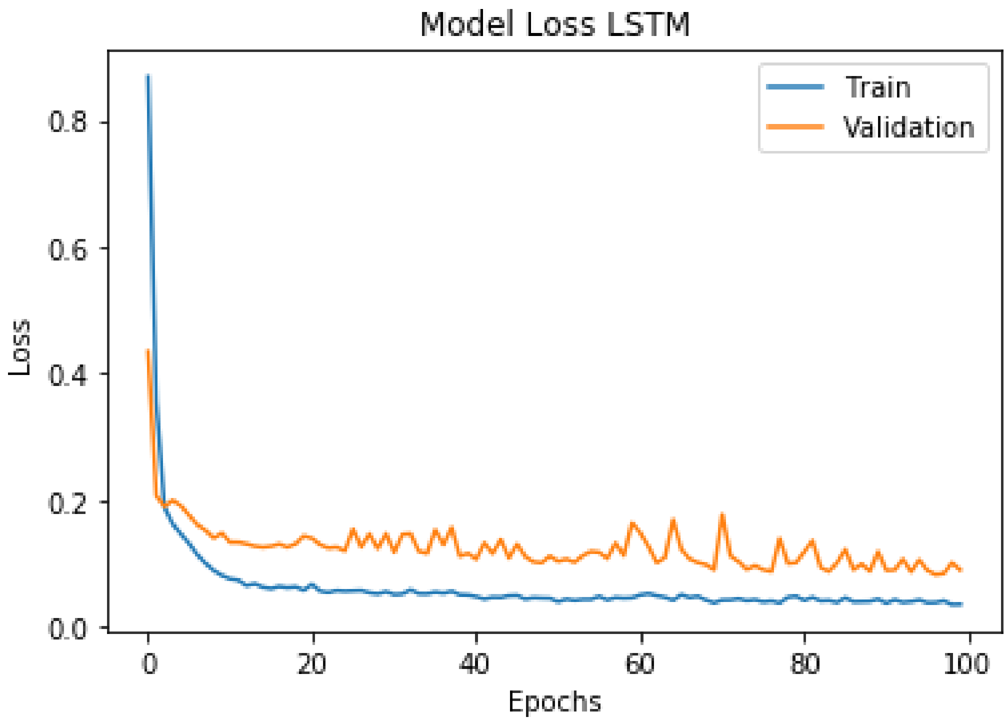
[27]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
↪close prices using LSTM")
plt.show()
```



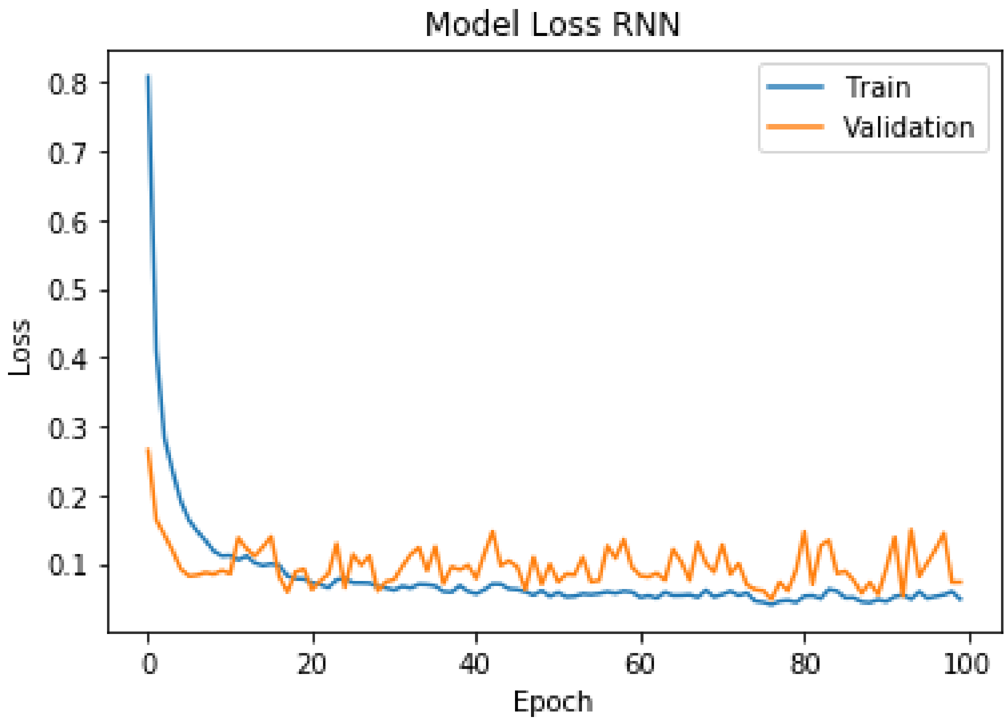
[28]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
↪close prices using RNN")
plt.show()
```



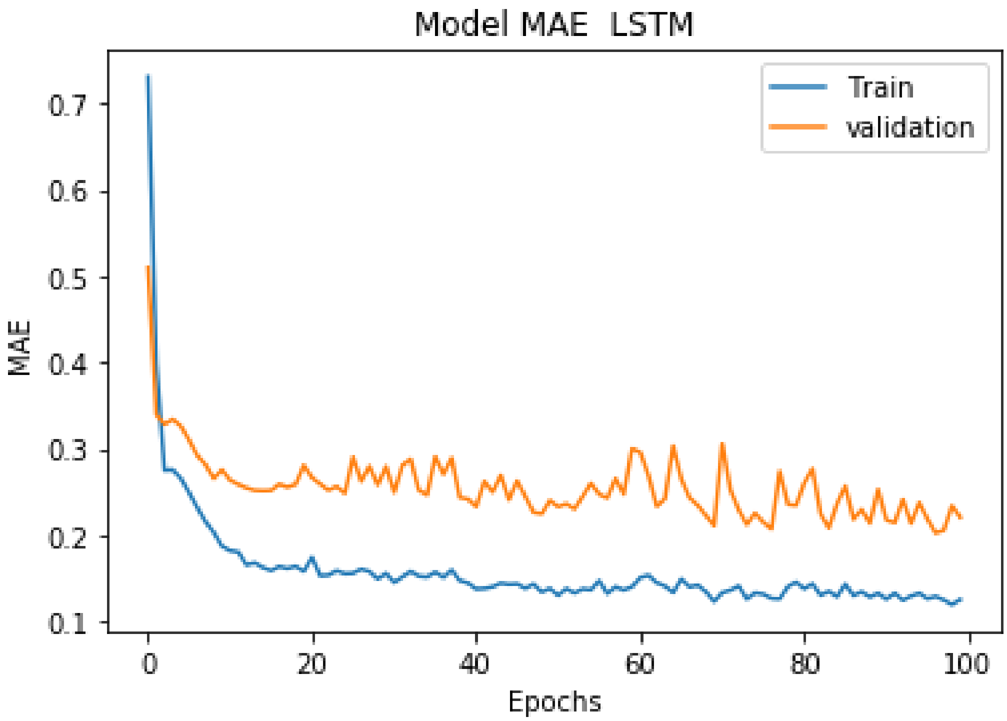
[29]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



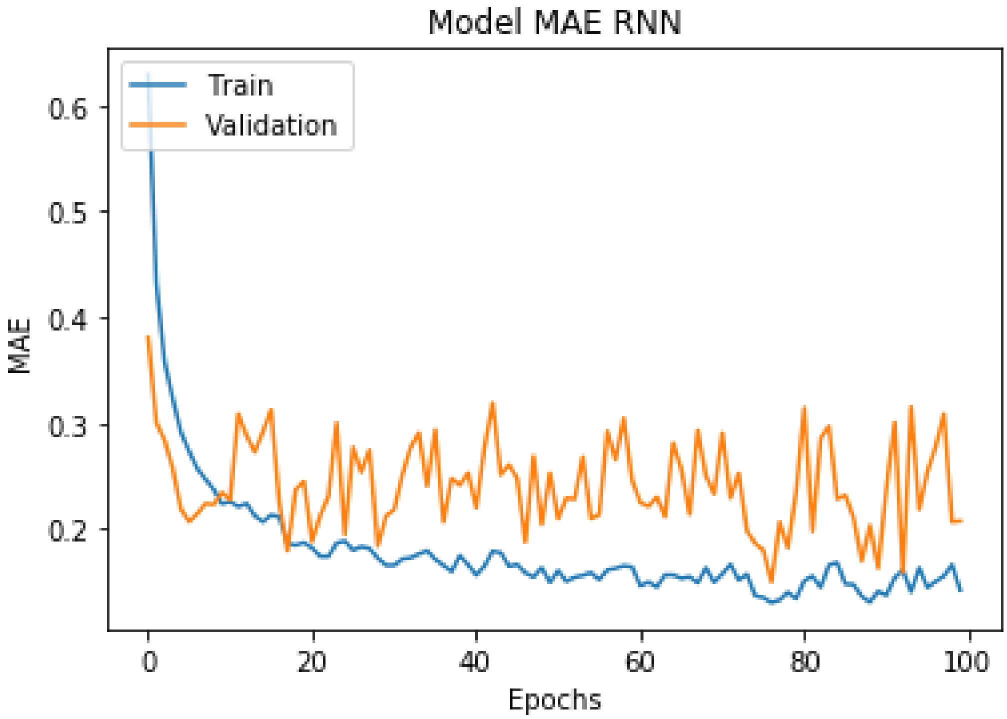
[30]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[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 ADBL Bank LSTM and RNN Models used for Stock Price Prediction
2. The Error is Low for RNN Model