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

[6]:
```
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]:
```
adbl_df.shape
```

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

Converting the Date into Panda's Date Time

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

1.4 Sorting the Date by Date in Ascending Order

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

1.5 Setting Features and Target Column

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

[11]:
```
X = adbl_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 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)}")
```

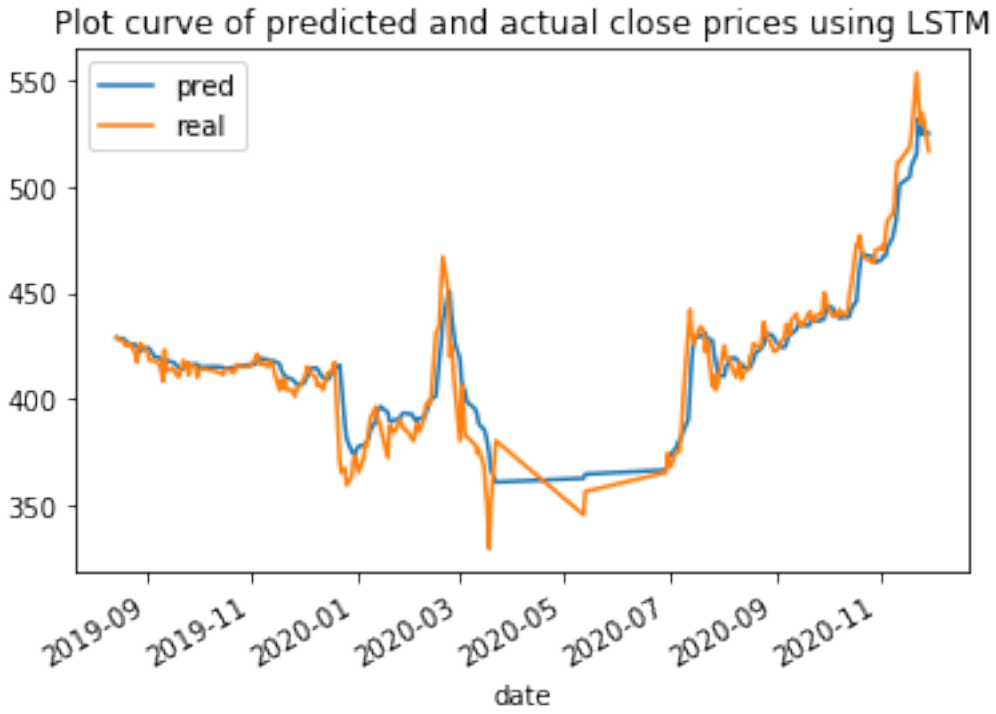
```
### RNN Model ###
RMSE 9.715363292220541
MAE 6.335043649824839
```

[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}"  )
```

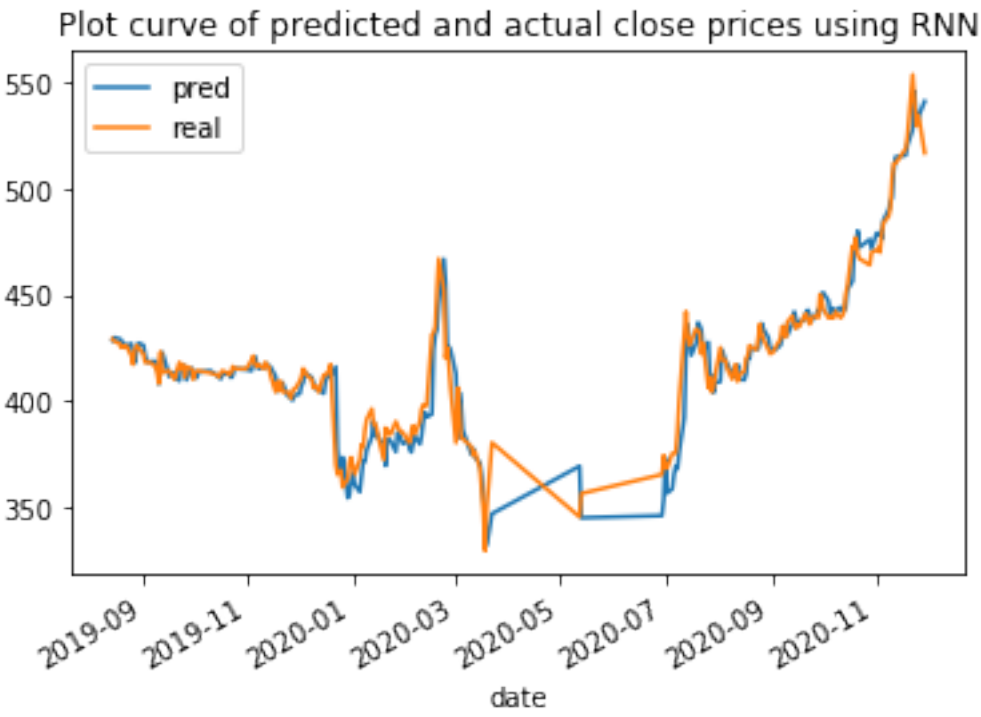
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

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

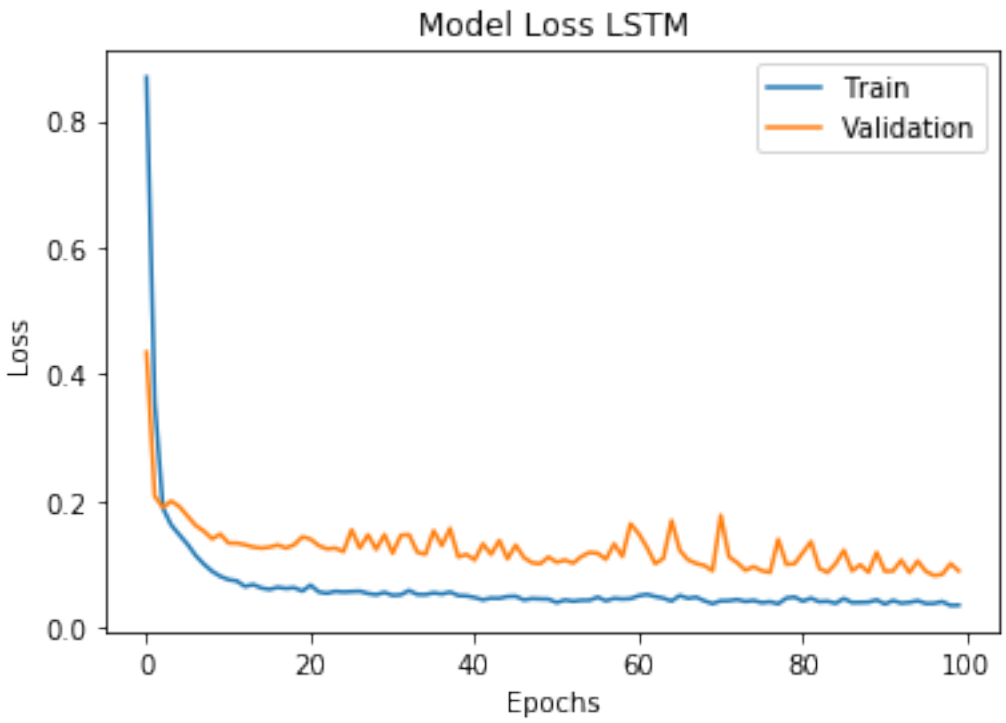
[27]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



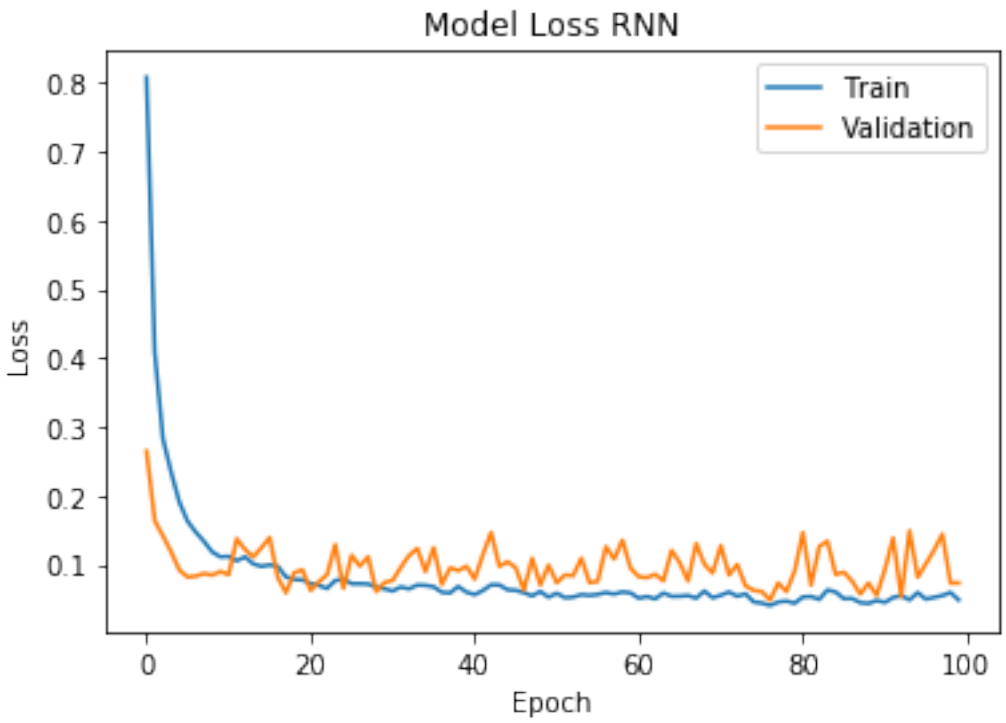
[28]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



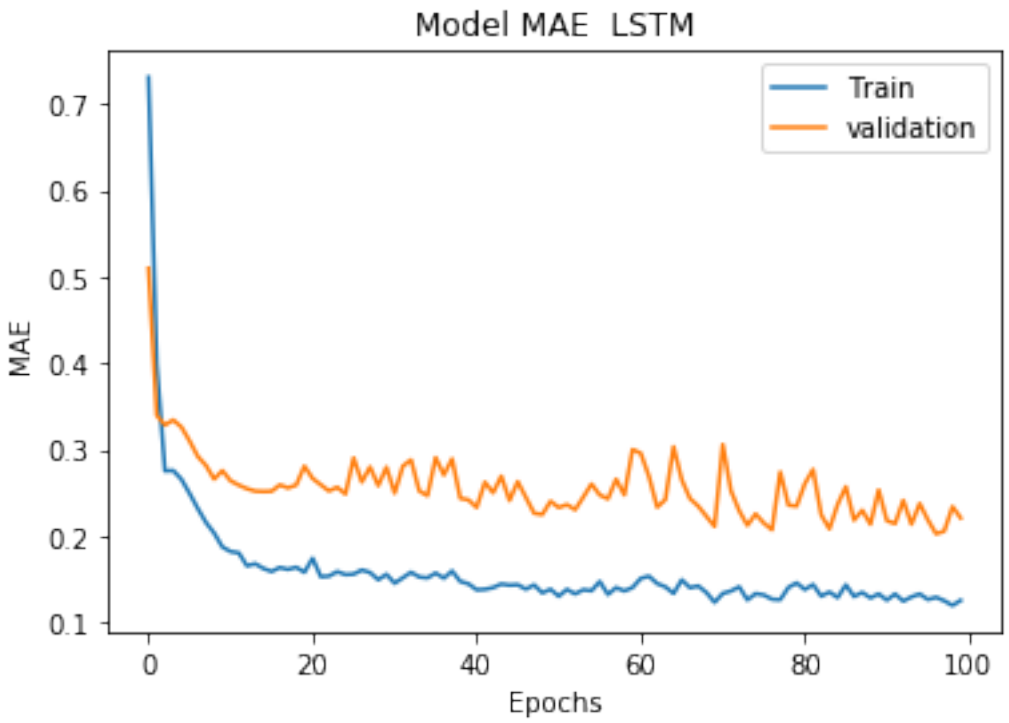
[29]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



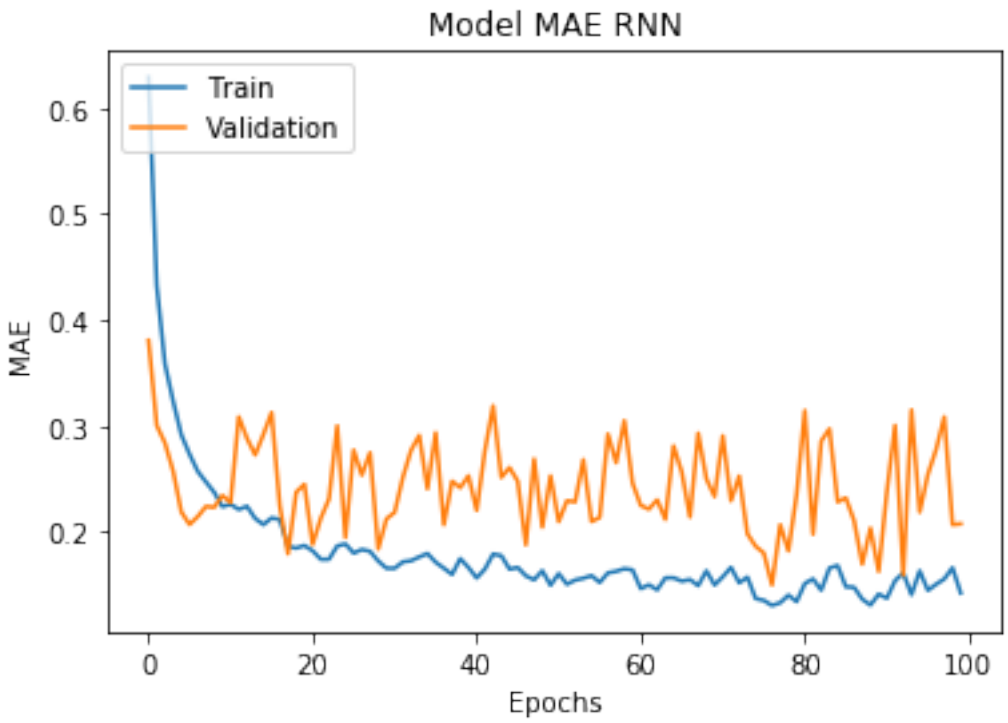
[30]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[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

Assignment4_Bank_NABIL

July 30, 2022

1 Stock Price Prediction of NABIL BANK

1.1 Import the Required Libraries

[106]:
```
import warnings
warnings.filterwarnings('ignore')
```

[107]:
```
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
from matplotlib import pyplot as plt
```

1.2 Load Data

[108]:
```
nabil_df = pd.read_csv("data/nabil.csv")
nabil_df.shape
```

[108]: (2392, 8)

[109]:
```
nabil_df.head()
```

[109]:

	Symbol	Date	Open	High	Low	Close	Percent Change	Volume
0	NABIL	2022-07-12	796.0	804.9	787.0	796.9	0.62	42720
1	NABIL	2022-07-11	831.0	831.0	792.0	792.0	-4.12	69864
2	NABIL	2022-07-08	845.0	847.0	825.5	826.0	-2.25	30318
3	NABIL	2022-07-07	864.0	886.0	844.0	845.0	-2.20	51271
4	NABIL	2022-07-06	885.0	889.0	837.0	864.0	-0.92	58061

1.3 Removing the Unwanted Columns

[110]:
```
nabil_df.drop(columns=['Symbol','Percent Change','Volume'],inplace=True)
```

[111]:
```
nabil_df.head()
```

[111]:

	Date	Open	High	Low	Close
0	2022-07-12	796.0	804.9	787.0	796.9
1	2022-07-11	831.0	831.0	792.0	792.0
2	2022-07-08	845.0	847.0	825.5	826.0
3	2022-07-07	864.0	886.0	844.0	845.0
4	2022-07-06	885.0	889.0	837.0	864.0

[112]:
```
nabil_df.shape
```

[112]: (2392, 5)

Converting the Date into Panda's Date Time

[113]:
```
nabil_df['Date'] = pd.to_datetime(nabil_df['Date'])
```

1.4 Sorting the Date by Date in Ascending Order

[114]:
```
nabil_df=nabil_df.sort_values(by='Date')
```

1.5 Setting Features and Target Column

[115]:
```
features = ['Date','Close']
```

[116]:
```
X = nabil_df[features]
```

[117]:
```
X.set_index("Date",inplace=True)
```

1.6 Splitting the Data Into Training, Validation and Test Set

[118]:
```
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

[119]:
```
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

[120]:
```
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

[121]:
```
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

[122]:
```
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'])
```

[123]:
```
RNN_Model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_6 (SimpleRNN)    (None, 5, 50)             2600

 dropout_10 (Dropout)        (None, 5, 50)             0

 simple_rnn_7 (SimpleRNN)    (None, 50)                5050

 dropout_11 (Dropout)        (None, 50)                0

 dense_5 (Dense)             (None, 50)                2550

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

1.11 Building LSTM Model

[124]:
```
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'])
```

[125]:
```
LSTM_Model.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 5, 50)             10400

 dropout_12 (Dropout)        (None, 5, 50)             0

 lstm_5 (LSTM)               (None, 50)                20200

 dropout_13 (Dropout)        (None, 50)                0

 dense_6 (Dense)             (None, 50)                2550

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

1.12 Fitting the RNN Model

[126]:
```
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
39/39 - 2s - loss: 0.7526 - mae: 0.6569 - val_loss: 0.1520 - val_mae: 0.2931 -
2s/epoch - 55ms/step
Epoch 2/100
39/39 - 0s - loss: 0.3223 - mae: 0.4272 - val_loss: 0.0657 - val_mae: 0.1782 -
225ms/epoch - 6ms/step
Epoch 3/100
39/39 - 0s - loss: 0.2306 - mae: 0.3600 - val_loss: 0.0489 - val_mae: 0.1455 -
219ms/epoch - 6ms/step
Epoch 4/100
```

```
39/39 - 0s - loss: 0.1781 - mae: 0.3145 - val_loss: 0.0384 - val_mae: 0.1217 -
216ms/epoch - 6ms/step
Epoch 5/100
39/39 - 0s - loss: 0.1486 - mae: 0.2865 - val_loss: 0.0362 - val_mae: 0.1226 -
217ms/epoch - 6ms/step
Epoch 6/100
39/39 - 0s - loss: 0.1268 - mae: 0.2644 - val_loss: 0.0324 - val_mae: 0.1174 -
219ms/epoch - 6ms/step
Epoch 7/100
39/39 - 0s - loss: 0.1124 - mae: 0.2476 - val_loss: 0.0261 - val_mae: 0.1038 -
220ms/epoch - 6ms/step
Epoch 8/100
39/39 - 0s - loss: 0.0997 - mae: 0.2322 - val_loss: 0.0291 - val_mae: 0.1225 -
218ms/epoch - 6ms/step
Epoch 9/100
39/39 - 0s - loss: 0.0924 - mae: 0.2240 - val_loss: 0.0205 - val_mae: 0.0900 -
219ms/epoch - 6ms/step
Epoch 10/100
39/39 - 0s - loss: 0.0889 - mae: 0.2178 - val_loss: 0.0266 - val_mae: 0.1182 -
224ms/epoch - 6ms/step
Epoch 11/100
39/39 - 0s - loss: 0.0866 - mae: 0.2173 - val_loss: 0.0279 - val_mae: 0.1219 -
221ms/epoch - 6ms/step
Epoch 12/100
39/39 - 0s - loss: 0.0914 - mae: 0.2269 - val_loss: 0.0294 - val_mae: 0.1273 -
225ms/epoch - 6ms/step
Epoch 13/100
39/39 - 0s - loss: 0.0876 - mae: 0.2241 - val_loss: 0.0288 - val_mae: 0.1226 -
214ms/epoch - 5ms/step
Epoch 14/100
39/39 - 0s - loss: 0.0742 - mae: 0.2039 - val_loss: 0.0209 - val_mae: 0.0941 -
220ms/epoch - 6ms/step
Epoch 15/100
39/39 - 0s - loss: 0.0655 - mae: 0.1886 - val_loss: 0.0211 - val_mae: 0.0988 -
225ms/epoch - 6ms/step
Epoch 16/100
39/39 - 0s - loss: 0.0682 - mae: 0.1934 - val_loss: 0.0257 - val_mae: 0.1122 -
216ms/epoch - 6ms/step
Epoch 17/100
39/39 - 0s - loss: 0.0653 - mae: 0.1894 - val_loss: 0.0218 - val_mae: 0.0931 -
217ms/epoch - 6ms/step
Epoch 18/100
39/39 - 0s - loss: 0.0629 - mae: 0.1848 - val_loss: 0.0227 - val_mae: 0.1009 -
218ms/epoch - 6ms/step
Epoch 19/100
39/39 - 0s - loss: 0.0566 - mae: 0.1740 - val_loss: 0.0163 - val_mae: 0.0773 -
217ms/epoch - 6ms/step
Epoch 20/100
```

```
39/39 - 0s - loss: 0.0522 - mae: 0.1669 - val_loss: 0.0173 - val_mae: 0.0835 -
228ms/epoch - 6ms/step
Epoch 21/100
39/39 - 0s - loss: 0.0552 - mae: 0.1735 - val_loss: 0.0228 - val_mae: 0.1076 -
216ms/epoch - 6ms/step
Epoch 22/100
39/39 - 0s - loss: 0.0520 - mae: 0.1670 - val_loss: 0.0194 - val_mae: 0.0986 -
218ms/epoch - 6ms/step
Epoch 23/100
39/39 - 0s - loss: 0.0506 - mae: 0.1629 - val_loss: 0.0144 - val_mae: 0.0736 -
236ms/epoch - 6ms/step
Epoch 24/100
39/39 - 0s - loss: 0.0622 - mae: 0.1864 - val_loss: 0.0247 - val_mae: 0.1204 -
224ms/epoch - 6ms/step
Epoch 25/100
39/39 - 0s - loss: 0.0628 - mae: 0.1899 - val_loss: 0.0227 - val_mae: 0.1072 -
218ms/epoch - 6ms/step
Epoch 26/100
39/39 - 0s - loss: 0.0642 - mae: 0.1927 - val_loss: 0.0225 - val_mae: 0.1101 -
215ms/epoch - 6ms/step
Epoch 27/100
39/39 - 0s - loss: 0.0596 - mae: 0.1853 - val_loss: 0.0297 - val_mae: 0.1329 -
219ms/epoch - 6ms/step
Epoch 28/100
39/39 - 0s - loss: 0.0547 - mae: 0.1741 - val_loss: 0.0181 - val_mae: 0.0863 -
222ms/epoch - 6ms/step
Epoch 29/100
39/39 - 0s - loss: 0.0510 - mae: 0.1658 - val_loss: 0.0171 - val_mae: 0.0835 -
218ms/epoch - 6ms/step
Epoch 30/100
39/39 - 0s - loss: 0.0491 - mae: 0.1622 - val_loss: 0.0249 - val_mae: 0.1211 -
240ms/epoch - 6ms/step
Epoch 31/100
39/39 - 0s - loss: 0.0462 - mae: 0.1579 - val_loss: 0.0176 - val_mae: 0.0905 -
228ms/epoch - 6ms/step
Epoch 32/100
39/39 - 0s - loss: 0.0436 - mae: 0.1527 - val_loss: 0.0169 - val_mae: 0.0872 -
217ms/epoch - 6ms/step
Epoch 33/100
39/39 - 0s - loss: 0.0495 - mae: 0.1618 - val_loss: 0.0171 - val_mae: 0.0877 -
225ms/epoch - 6ms/step
Epoch 34/100
39/39 - 0s - loss: 0.0504 - mae: 0.1641 - val_loss: 0.0176 - val_mae: 0.0906 -
221ms/epoch - 6ms/step
Epoch 35/100
39/39 - 0s - loss: 0.0434 - mae: 0.1525 - val_loss: 0.0175 - val_mae: 0.0899 -
215ms/epoch - 6ms/step
Epoch 36/100
```

```
39/39 - 0s - loss: 0.0443 - mae: 0.1539 - val_loss: 0.0171 - val_mae: 0.0883 -
224ms/epoch - 6ms/step
Epoch 37/100
39/39 - 0s - loss: 0.0406 - mae: 0.1470 - val_loss: 0.0180 - val_mae: 0.0955 -
222ms/epoch - 6ms/step
Epoch 38/100
39/39 - 0s - loss: 0.0403 - mae: 0.1467 - val_loss: 0.0205 - val_mae: 0.1063 -
216ms/epoch - 6ms/step
Epoch 39/100
39/39 - 0s - loss: 0.0483 - mae: 0.1622 - val_loss: 0.0192 - val_mae: 0.0980 -
226ms/epoch - 6ms/step
Epoch 40/100
39/39 - 0s - loss: 0.0441 - mae: 0.1547 - val_loss: 0.0169 - val_mae: 0.0877 -
222ms/epoch - 6ms/step
Epoch 41/100
39/39 - 0s - loss: 0.0453 - mae: 0.1570 - val_loss: 0.0183 - val_mae: 0.0936 -
223ms/epoch - 6ms/step
Epoch 42/100
39/39 - 0s - loss: 0.0450 - mae: 0.1597 - val_loss: 0.0183 - val_mae: 0.0947 -
369ms/epoch - 9ms/step
Epoch 43/100
39/39 - 0s - loss: 0.0434 - mae: 0.1527 - val_loss: 0.0175 - val_mae: 0.0913 -
396ms/epoch - 10ms/step
Epoch 44/100
39/39 - 0s - loss: 0.0458 - mae: 0.1570 - val_loss: 0.0204 - val_mae: 0.1038 -
215ms/epoch - 6ms/step
Epoch 45/100
39/39 - 0s - loss: 0.0383 - mae: 0.1448 - val_loss: 0.0179 - val_mae: 0.0942 -
215ms/epoch - 6ms/step
Epoch 46/100
39/39 - 0s - loss: 0.0395 - mae: 0.1472 - val_loss: 0.0164 - val_mae: 0.0864 -
218ms/epoch - 6ms/step
Epoch 47/100
39/39 - 0s - loss: 0.0430 - mae: 0.1523 - val_loss: 0.0161 - val_mae: 0.0856 -
224ms/epoch - 6ms/step
Epoch 48/100
39/39 - 0s - loss: 0.0482 - mae: 0.1612 - val_loss: 0.0274 - val_mae: 0.1291 -
221ms/epoch - 6ms/step
Epoch 49/100
39/39 - 0s - loss: 0.0438 - mae: 0.1571 - val_loss: 0.0164 - val_mae: 0.0846 -
219ms/epoch - 6ms/step
Epoch 50/100
39/39 - 0s - loss: 0.0386 - mae: 0.1450 - val_loss: 0.0225 - val_mae: 0.1156 -
218ms/epoch - 6ms/step
Epoch 51/100
39/39 - 0s - loss: 0.0407 - mae: 0.1479 - val_loss: 0.0161 - val_mae: 0.0801 -
220ms/epoch - 6ms/step
Epoch 52/100
```

```
39/39 - 0s - loss: 0.0366 - mae: 0.1399 - val_loss: 0.0174 - val_mae: 0.0874 -
217ms/epoch - 6ms/step
Epoch 53/100
39/39 - 0s - loss: 0.0377 - mae: 0.1422 - val_loss: 0.0172 - val_mae: 0.0880 -
220ms/epoch - 6ms/step
Epoch 54/100
39/39 - 0s - loss: 0.0365 - mae: 0.1395 - val_loss: 0.0174 - val_mae: 0.0935 -
225ms/epoch - 6ms/step
Epoch 55/100
39/39 - 0s - loss: 0.0393 - mae: 0.1464 - val_loss: 0.0144 - val_mae: 0.0754 -
208ms/epoch - 5ms/step
Epoch 56/100
39/39 - 0s - loss: 0.0416 - mae: 0.1494 - val_loss: 0.0181 - val_mae: 0.0935 -
214ms/epoch - 5ms/step
Epoch 57/100
39/39 - 0s - loss: 0.0463 - mae: 0.1592 - val_loss: 0.0208 - val_mae: 0.1019 -
210ms/epoch - 5ms/step
Epoch 58/100
39/39 - 0s - loss: 0.0492 - mae: 0.1661 - val_loss: 0.0222 - val_mae: 0.1104 -
226ms/epoch - 6ms/step
Epoch 59/100
39/39 - 0s - loss: 0.0407 - mae: 0.1518 - val_loss: 0.0166 - val_mae: 0.0848 -
267ms/epoch - 7ms/step
Epoch 60/100
39/39 - 0s - loss: 0.0393 - mae: 0.1425 - val_loss: 0.0220 - val_mae: 0.1121 -
232ms/epoch - 6ms/step
Epoch 61/100
39/39 - 0s - loss: 0.0393 - mae: 0.1454 - val_loss: 0.0153 - val_mae: 0.0792 -
215ms/epoch - 6ms/step
Epoch 62/100
39/39 - 0s - loss: 0.0343 - mae: 0.1354 - val_loss: 0.0176 - val_mae: 0.0919 -
214ms/epoch - 5ms/step
Epoch 63/100
39/39 - 0s - loss: 0.0352 - mae: 0.1367 - val_loss: 0.0154 - val_mae: 0.0762 -
209ms/epoch - 5ms/step
Epoch 64/100
39/39 - 0s - loss: 0.0364 - mae: 0.1385 - val_loss: 0.0140 - val_mae: 0.0734 -
227ms/epoch - 6ms/step
Epoch 65/100
39/39 - 0s - loss: 0.0377 - mae: 0.1410 - val_loss: 0.0202 - val_mae: 0.1026 -
213ms/epoch - 5ms/step
Epoch 66/100
39/39 - 0s - loss: 0.0381 - mae: 0.1462 - val_loss: 0.0230 - val_mae: 0.1176 -
207ms/epoch - 5ms/step
Epoch 67/100
39/39 - 0s - loss: 0.0364 - mae: 0.1398 - val_loss: 0.0146 - val_mae: 0.0709 -
209ms/epoch - 5ms/step
Epoch 68/100
```

```
39/39 - 0s - loss: 0.0391 - mae: 0.1415 - val_loss: 0.0159 - val_mae: 0.0846 -
212ms/epoch - 5ms/step
Epoch 69/100
39/39 - 0s - loss: 0.0379 - mae: 0.1423 - val_loss: 0.0193 - val_mae: 0.1003 -
210ms/epoch - 5ms/step
Epoch 70/100
39/39 - 0s - loss: 0.0411 - mae: 0.1520 - val_loss: 0.0184 - val_mae: 0.0956 -
216ms/epoch - 6ms/step
Epoch 71/100
39/39 - 0s - loss: 0.0380 - mae: 0.1449 - val_loss: 0.0189 - val_mae: 0.1001 -
209ms/epoch - 5ms/step
Epoch 72/100
39/39 - 0s - loss: 0.0427 - mae: 0.1514 - val_loss: 0.0183 - val_mae: 0.0936 -
216ms/epoch - 6ms/step
Epoch 73/100
39/39 - 0s - loss: 0.0412 - mae: 0.1482 - val_loss: 0.0217 - val_mae: 0.1115 -
213ms/epoch - 5ms/step
Epoch 74/100
39/39 - 0s - loss: 0.0381 - mae: 0.1416 - val_loss: 0.0187 - val_mae: 0.0982 -
210ms/epoch - 5ms/step
Epoch 75/100
39/39 - 0s - loss: 0.0370 - mae: 0.1428 - val_loss: 0.0213 - val_mae: 0.1099 -
214ms/epoch - 5ms/step
Epoch 76/100
39/39 - 0s - loss: 0.0405 - mae: 0.1488 - val_loss: 0.0226 - val_mae: 0.1163 -
208ms/epoch - 5ms/step
Epoch 77/100
39/39 - 0s - loss: 0.0392 - mae: 0.1458 - val_loss: 0.0212 - val_mae: 0.1103 -
212ms/epoch - 5ms/step
Epoch 78/100
39/39 - 0s - loss: 0.0404 - mae: 0.1465 - val_loss: 0.0177 - val_mae: 0.0944 -
214ms/epoch - 5ms/step
Epoch 79/100
39/39 - 0s - loss: 0.0402 - mae: 0.1482 - val_loss: 0.0198 - val_mae: 0.1018 -
208ms/epoch - 5ms/step
Epoch 80/100
39/39 - 0s - loss: 0.0413 - mae: 0.1476 - val_loss: 0.0223 - val_mae: 0.1140 -
208ms/epoch - 5ms/step
Epoch 81/100
39/39 - 0s - loss: 0.0391 - mae: 0.1459 - val_loss: 0.0225 - val_mae: 0.1160 -
219ms/epoch - 6ms/step
Epoch 82/100
39/39 - 0s - loss: 0.0418 - mae: 0.1506 - val_loss: 0.0192 - val_mae: 0.1016 -
214ms/epoch - 5ms/step
Epoch 83/100
39/39 - 0s - loss: 0.0365 - mae: 0.1418 - val_loss: 0.0181 - val_mae: 0.0961 -
209ms/epoch - 5ms/step
Epoch 84/100
```

```
39/39 - 0s - loss: 0.0414 - mae: 0.1485 - val_loss: 0.0168 - val_mae: 0.0891 -
205ms/epoch - 5ms/step
Epoch 85/100
39/39 - 0s - loss: 0.0372 - mae: 0.1420 - val_loss: 0.0207 - val_mae: 0.1081 -
212ms/epoch - 5ms/step
Epoch 86/100
39/39 - 0s - loss: 0.0373 - mae: 0.1418 - val_loss: 0.0151 - val_mae: 0.0791 -
211ms/epoch - 5ms/step
Epoch 87/100
39/39 - 0s - loss: 0.0431 - mae: 0.1538 - val_loss: 0.0195 - val_mae: 0.0999 -
229ms/epoch - 6ms/step
Epoch 88/100
39/39 - 0s - loss: 0.0411 - mae: 0.1518 - val_loss: 0.0219 - val_mae: 0.1124 -
207ms/epoch - 5ms/step
Epoch 89/100
39/39 - 0s - loss: 0.0389 - mae: 0.1469 - val_loss: 0.0214 - val_mae: 0.1098 -
207ms/epoch - 5ms/step
Epoch 90/100
39/39 - 0s - loss: 0.0349 - mae: 0.1374 - val_loss: 0.0255 - val_mae: 0.1277 -
213ms/epoch - 5ms/step
Epoch 91/100
39/39 - 0s - loss: 0.0350 - mae: 0.1382 - val_loss: 0.0225 - val_mae: 0.1144 -
209ms/epoch - 5ms/step
Epoch 92/100
39/39 - 0s - loss: 0.0370 - mae: 0.1404 - val_loss: 0.0258 - val_mae: 0.1291 -
204ms/epoch - 5ms/step
Epoch 93/100
39/39 - 0s - loss: 0.0365 - mae: 0.1386 - val_loss: 0.0246 - val_mae: 0.1231 -
211ms/epoch - 5ms/step
Epoch 94/100
39/39 - 0s - loss: 0.0374 - mae: 0.1412 - val_loss: 0.0213 - val_mae: 0.1107 -
210ms/epoch - 5ms/step
Epoch 95/100
39/39 - 0s - loss: 0.0359 - mae: 0.1405 - val_loss: 0.0205 - val_mae: 0.1023 -
206ms/epoch - 5ms/step
Epoch 96/100
39/39 - 0s - loss: 0.0359 - mae: 0.1371 - val_loss: 0.0141 - val_mae: 0.0745 -
209ms/epoch - 5ms/step
Epoch 97/100
39/39 - 0s - loss: 0.0373 - mae: 0.1391 - val_loss: 0.0205 - val_mae: 0.1089 -
208ms/epoch - 5ms/step
Epoch 98/100
39/39 - 0s - loss: 0.0363 - mae: 0.1395 - val_loss: 0.0182 - val_mae: 0.0974 -
207ms/epoch - 5ms/step
Epoch 99/100
39/39 - 0s - loss: 0.0348 - mae: 0.1384 - val_loss: 0.0150 - val_mae: 0.0805 -
212ms/epoch - 5ms/step
Epoch 100/100
```

```
39/39 - 0s - loss: 0.0378 - mae: 0.1447 - val_loss: 0.0199 - val_mae: 0.1019 -
212ms/epoch - 5ms/step
```

[127]:
```
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
39/39 - 5s - loss: 0.8346 - mae: 0.7923 - val_loss: 0.4397 - val_mae: 0.5781 -
5s/epoch - 125ms/step
Epoch 2/100
39/39 - 0s - loss: 0.2562 - mae: 0.3796 - val_loss: 0.0979 - val_mae: 0.2256 -
379ms/epoch - 10ms/step
Epoch 3/100
39/39 - 0s - loss: 0.1387 - mae: 0.2817 - val_loss: 0.0571 - val_mae: 0.1537 -
379ms/epoch - 10ms/step
Epoch 4/100
39/39 - 0s - loss: 0.1133 - mae: 0.2517 - val_loss: 0.0543 - val_mae: 0.1473 -
382ms/epoch - 10ms/step
Epoch 5/100
39/39 - 0s - loss: 0.0959 - mae: 0.2313 - val_loss: 0.0509 - val_mae: 0.1403 -
384ms/epoch - 10ms/step
Epoch 6/100
39/39 - 0s - loss: 0.0812 - mae: 0.2105 - val_loss: 0.0535 - val_mae: 0.1488 -
382ms/epoch - 10ms/step
Epoch 7/100
39/39 - 0s - loss: 0.0794 - mae: 0.2088 - val_loss: 0.0575 - val_mae: 0.1547 -
379ms/epoch - 10ms/step
Epoch 8/100
39/39 - 0s - loss: 0.0739 - mae: 0.2007 - val_loss: 0.0687 - val_mae: 0.1770 -
377ms/epoch - 10ms/step
Epoch 9/100
39/39 - 0s - loss: 0.0716 - mae: 0.1979 - val_loss: 0.0736 - val_mae: 0.1852 -
377ms/epoch - 10ms/step
Epoch 10/100
39/39 - 0s - loss: 0.0735 - mae: 0.2007 - val_loss: 0.1193 - val_mae: 0.2592 -
371ms/epoch - 10ms/step
Epoch 11/100
39/39 - 0s - loss: 0.0767 - mae: 0.2064 - val_loss: 0.1395 - val_mae: 0.2841 -
379ms/epoch - 10ms/step
Epoch 12/100
39/39 - 0s - loss: 0.0813 - mae: 0.2148 - val_loss: 0.1186 - val_mae: 0.2553 -
383ms/epoch - 10ms/step
Epoch 13/100
39/39 - 0s - loss: 0.0770 - mae: 0.2091 - val_loss: 0.0957 - val_mae: 0.2278 -
379ms/epoch - 10ms/step
Epoch 14/100
39/39 - 0s - loss: 0.0701 - mae: 0.1981 - val_loss: 0.0724 - val_mae: 0.1911 -
```

```
375ms/epoch - 10ms/step
Epoch 15/100
39/39 - 0s - loss: 0.0680 - mae: 0.1939 - val_loss: 0.0738 - val_mae: 0.1918 -
380ms/epoch - 10ms/step
Epoch 16/100
39/39 - 0s - loss: 0.0615 - mae: 0.1837 - val_loss: 0.0504 - val_mae: 0.1457 -
382ms/epoch - 10ms/step
Epoch 17/100
39/39 - 0s - loss: 0.0565 - mae: 0.1742 - val_loss: 0.0542 - val_mae: 0.1585 -
381ms/epoch - 10ms/step
Epoch 18/100
39/39 - 0s - loss: 0.0536 - mae: 0.1711 - val_loss: 0.0506 - val_mae: 0.1480 -
380ms/epoch - 10ms/step
Epoch 19/100
39/39 - 0s - loss: 0.0541 - mae: 0.1720 - val_loss: 0.0713 - val_mae: 0.1929 -
379ms/epoch - 10ms/step
Epoch 20/100
39/39 - 0s - loss: 0.0580 - mae: 0.1788 - val_loss: 0.0841 - val_mae: 0.2156 -
382ms/epoch - 10ms/step
Epoch 21/100
39/39 - 0s - loss: 0.0605 - mae: 0.1844 - val_loss: 0.1131 - val_mae: 0.2517 -
382ms/epoch - 10ms/step
Epoch 22/100
39/39 - 0s - loss: 0.0709 - mae: 0.1990 - val_loss: 0.0997 - val_mae: 0.2293 -
378ms/epoch - 10ms/step
Epoch 23/100
39/39 - 0s - loss: 0.0664 - mae: 0.1911 - val_loss: 0.0720 - val_mae: 0.1893 -
378ms/epoch - 10ms/step
Epoch 24/100
39/39 - 0s - loss: 0.0550 - mae: 0.1747 - val_loss: 0.0635 - val_mae: 0.1782 -
377ms/epoch - 10ms/step
Epoch 25/100
39/39 - 0s - loss: 0.0561 - mae: 0.1746 - val_loss: 0.0651 - val_mae: 0.1828 -
380ms/epoch - 10ms/step
Epoch 26/100
39/39 - 1s - loss: 0.0551 - mae: 0.1731 - val_loss: 0.0726 - val_mae: 0.1952 -
688ms/epoch - 18ms/step
Epoch 27/100
39/39 - 0s - loss: 0.0563 - mae: 0.1760 - val_loss: 0.0696 - val_mae: 0.1905 -
392ms/epoch - 10ms/step
Epoch 28/100
39/39 - 0s - loss: 0.0557 - mae: 0.1755 - val_loss: 0.0645 - val_mae: 0.1817 -
373ms/epoch - 10ms/step
Epoch 29/100
39/39 - 0s - loss: 0.0512 - mae: 0.1683 - val_loss: 0.0588 - val_mae: 0.1744 -
375ms/epoch - 10ms/step
Epoch 30/100
39/39 - 0s - loss: 0.0496 - mae: 0.1634 - val_loss: 0.0467 - val_mae: 0.1433 -
```

```
378ms/epoch - 10ms/step
Epoch 31/100
39/39 - 0s - loss: 0.0459 - mae: 0.1592 - val_loss: 0.0450 - val_mae: 0.1362 -
376ms/epoch - 10ms/step
Epoch 32/100
39/39 - 0s - loss: 0.0462 - mae: 0.1584 - val_loss: 0.0537 - val_mae: 0.1627 -
377ms/epoch - 10ms/step
Epoch 33/100
39/39 - 0s - loss: 0.0480 - mae: 0.1605 - val_loss: 0.0597 - val_mae: 0.1784 -
377ms/epoch - 10ms/step
Epoch 34/100
39/39 - 0s - loss: 0.0511 - mae: 0.1660 - val_loss: 0.0568 - val_mae: 0.1634 -
373ms/epoch - 10ms/step
Epoch 35/100
39/39 - 0s - loss: 0.0462 - mae: 0.1611 - val_loss: 0.0587 - val_mae: 0.1739 -
377ms/epoch - 10ms/step
Epoch 36/100
39/39 - 0s - loss: 0.0488 - mae: 0.1620 - val_loss: 0.0548 - val_mae: 0.1631 -
377ms/epoch - 10ms/step
Epoch 37/100
39/39 - 0s - loss: 0.0453 - mae: 0.1560 - val_loss: 0.0589 - val_mae: 0.1690 -
377ms/epoch - 10ms/step
Epoch 38/100
39/39 - 0s - loss: 0.0476 - mae: 0.1591 - val_loss: 0.0533 - val_mae: 0.1577 -
378ms/epoch - 10ms/step
Epoch 39/100
39/39 - 0s - loss: 0.0472 - mae: 0.1587 - val_loss: 0.0621 - val_mae: 0.1712 -
375ms/epoch - 10ms/step
Epoch 40/100
39/39 - 0s - loss: 0.0486 - mae: 0.1609 - val_loss: 0.0516 - val_mae: 0.1536 -
374ms/epoch - 10ms/step
Epoch 41/100
39/39 - 0s - loss: 0.0478 - mae: 0.1594 - val_loss: 0.0547 - val_mae: 0.1673 -
379ms/epoch - 10ms/step
Epoch 42/100
39/39 - 0s - loss: 0.0459 - mae: 0.1576 - val_loss: 0.0553 - val_mae: 0.1675 -
380ms/epoch - 10ms/step
Epoch 43/100
39/39 - 0s - loss: 0.0454 - mae: 0.1573 - val_loss: 0.0464 - val_mae: 0.1448 -
374ms/epoch - 10ms/step
Epoch 44/100
39/39 - 0s - loss: 0.0419 - mae: 0.1500 - val_loss: 0.0472 - val_mae: 0.1430 -
379ms/epoch - 10ms/step
Epoch 45/100
39/39 - 0s - loss: 0.0403 - mae: 0.1476 - val_loss: 0.0664 - val_mae: 0.1862 -
369ms/epoch - 9ms/step
Epoch 46/100
39/39 - 0s - loss: 0.0477 - mae: 0.1627 - val_loss: 0.0633 - val_mae: 0.1807 -
```

```
365ms/epoch - 9ms/step
Epoch 47/100
39/39 - 0s - loss: 0.0490 - mae: 0.1640 - val_loss: 0.0937 - val_mae: 0.2302 -
363ms/epoch - 9ms/step
Epoch 48/100
39/39 - 0s - loss: 0.0516 - mae: 0.1721 - val_loss: 0.0645 - val_mae: 0.1867 -
373ms/epoch - 10ms/step
Epoch 49/100
39/39 - 0s - loss: 0.0500 - mae: 0.1634 - val_loss: 0.0685 - val_mae: 0.1917 -
369ms/epoch - 9ms/step
Epoch 50/100
39/39 - 0s - loss: 0.0478 - mae: 0.1630 - val_loss: 0.0684 - val_mae: 0.1983 -
369ms/epoch - 9ms/step
Epoch 51/100
39/39 - 0s - loss: 0.0495 - mae: 0.1663 - val_loss: 0.0407 - val_mae: 0.1318 -
368ms/epoch - 9ms/step
Epoch 52/100
39/39 - 0s - loss: 0.0407 - mae: 0.1476 - val_loss: 0.0296 - val_mae: 0.1049 -
365ms/epoch - 9ms/step
Epoch 53/100
39/39 - 0s - loss: 0.0382 - mae: 0.1417 - val_loss: 0.0334 - val_mae: 0.1168 -
366ms/epoch - 9ms/step
Epoch 54/100
39/39 - 0s - loss: 0.0390 - mae: 0.1446 - val_loss: 0.0349 - val_mae: 0.1197 -
367ms/epoch - 9ms/step
Epoch 55/100
39/39 - 0s - loss: 0.0380 - mae: 0.1433 - val_loss: 0.0364 - val_mae: 0.1231 -
359ms/epoch - 9ms/step
Epoch 56/100
39/39 - 0s - loss: 0.0380 - mae: 0.1412 - val_loss: 0.0405 - val_mae: 0.1331 -
381ms/epoch - 10ms/step
Epoch 57/100
39/39 - 0s - loss: 0.0385 - mae: 0.1449 - val_loss: 0.0475 - val_mae: 0.1588 -
428ms/epoch - 11ms/step
Epoch 58/100
39/39 - 0s - loss: 0.0417 - mae: 0.1498 - val_loss: 0.0457 - val_mae: 0.1518 -
360ms/epoch - 9ms/step
Epoch 59/100
39/39 - 0s - loss: 0.0407 - mae: 0.1487 - val_loss: 0.0403 - val_mae: 0.1339 -
359ms/epoch - 9ms/step
Epoch 60/100
39/39 - 0s - loss: 0.0402 - mae: 0.1483 - val_loss: 0.0506 - val_mae: 0.1622 -
353ms/epoch - 9ms/step
Epoch 61/100
39/39 - 0s - loss: 0.0444 - mae: 0.1542 - val_loss: 0.0523 - val_mae: 0.1636 -
361ms/epoch - 9ms/step
Epoch 62/100
39/39 - 0s - loss: 0.0398 - mae: 0.1478 - val_loss: 0.0548 - val_mae: 0.1694 -
```

```
364ms/epoch - 9ms/step
Epoch 63/100
39/39 - 0s - loss: 0.0432 - mae: 0.1522 - val_loss: 0.0538 - val_mae: 0.1640 -
367ms/epoch - 9ms/step
Epoch 64/100
39/39 - 0s - loss: 0.0432 - mae: 0.1541 - val_loss: 0.0701 - val_mae: 0.1967 -
365ms/epoch - 9ms/step
Epoch 65/100
39/39 - 0s - loss: 0.0480 - mae: 0.1633 - val_loss: 0.0721 - val_mae: 0.1950 -
372ms/epoch - 10ms/step
Epoch 66/100
39/39 - 0s - loss: 0.0499 - mae: 0.1683 - val_loss: 0.0790 - val_mae: 0.2176 -
368ms/epoch - 9ms/step
Epoch 67/100
39/39 - 0s - loss: 0.0468 - mae: 0.1596 - val_loss: 0.0570 - val_mae: 0.1726 -
379ms/epoch - 10ms/step
Epoch 68/100
39/39 - 0s - loss: 0.0442 - mae: 0.1554 - val_loss: 0.0474 - val_mae: 0.1555 -
363ms/epoch - 9ms/step
Epoch 69/100
39/39 - 0s - loss: 0.0405 - mae: 0.1479 - val_loss: 0.0421 - val_mae: 0.1444 -
373ms/epoch - 10ms/step
Epoch 70/100
39/39 - 0s - loss: 0.0401 - mae: 0.1472 - val_loss: 0.0419 - val_mae: 0.1443 -
367ms/epoch - 9ms/step
Epoch 71/100
39/39 - 0s - loss: 0.0375 - mae: 0.1442 - val_loss: 0.0425 - val_mae: 0.1436 -
370ms/epoch - 9ms/step
Epoch 72/100
39/39 - 0s - loss: 0.0417 - mae: 0.1493 - val_loss: 0.0529 - val_mae: 0.1616 -
366ms/epoch - 9ms/step
Epoch 73/100
39/39 - 0s - loss: 0.0443 - mae: 0.1570 - val_loss: 0.0547 - val_mae: 0.1728 -
366ms/epoch - 9ms/step
Epoch 74/100
39/39 - 0s - loss: 0.0439 - mae: 0.1568 - val_loss: 0.0574 - val_mae: 0.1763 -
355ms/epoch - 9ms/step
Epoch 75/100
39/39 - 0s - loss: 0.0443 - mae: 0.1586 - val_loss: 0.0640 - val_mae: 0.1874 -
355ms/epoch - 9ms/step
Epoch 76/100
39/39 - 0s - loss: 0.0446 - mae: 0.1567 - val_loss: 0.0487 - val_mae: 0.1642 -
356ms/epoch - 9ms/step
Epoch 77/100
39/39 - 0s - loss: 0.0425 - mae: 0.1527 - val_loss: 0.0368 - val_mae: 0.1359 -
357ms/epoch - 9ms/step
Epoch 78/100
39/39 - 0s - loss: 0.0406 - mae: 0.1482 - val_loss: 0.0341 - val_mae: 0.1249 -
```

```
362ms/epoch - 9ms/step
Epoch 79/100
39/39 - 0s - loss: 0.0379 - mae: 0.1421 - val_loss: 0.0355 - val_mae: 0.1270 -
458ms/epoch - 12ms/step
Epoch 80/100
39/39 - 0s - loss: 0.0383 - mae: 0.1407 - val_loss: 0.0322 - val_mae: 0.1205 -
424ms/epoch - 11ms/step
Epoch 81/100
39/39 - 0s - loss: 0.0375 - mae: 0.1400 - val_loss: 0.0303 - val_mae: 0.1161 -
374ms/epoch - 10ms/step
Epoch 82/100
39/39 - 0s - loss: 0.0354 - mae: 0.1384 - val_loss: 0.0231 - val_mae: 0.0895 -
390ms/epoch - 10ms/step
Epoch 83/100
39/39 - 0s - loss: 0.0348 - mae: 0.1356 - val_loss: 0.0267 - val_mae: 0.1071 -
493ms/epoch - 13ms/step
Epoch 84/100
39/39 - 0s - loss: 0.0362 - mae: 0.1378 - val_loss: 0.0239 - val_mae: 0.0934 -
381ms/epoch - 10ms/step
Epoch 85/100
39/39 - 1s - loss: 0.0333 - mae: 0.1338 - val_loss: 0.0256 - val_mae: 0.0981 -
504ms/epoch - 13ms/step
Epoch 86/100
39/39 - 1s - loss: 0.0336 - mae: 0.1336 - val_loss: 0.0336 - val_mae: 0.1277 -
519ms/epoch - 13ms/step
Epoch 87/100
39/39 - 0s - loss: 0.0380 - mae: 0.1440 - val_loss: 0.0428 - val_mae: 0.1475 -
402ms/epoch - 10ms/step
Epoch 88/100
39/39 - 0s - loss: 0.0396 - mae: 0.1476 - val_loss: 0.0506 - val_mae: 0.1692 -
476ms/epoch - 12ms/step
Epoch 89/100
39/39 - 0s - loss: 0.0400 - mae: 0.1489 - val_loss: 0.0512 - val_mae: 0.1654 -
427ms/epoch - 11ms/step
Epoch 90/100
39/39 - 0s - loss: 0.0428 - mae: 0.1516 - val_loss: 0.0712 - val_mae: 0.2132 -
441ms/epoch - 11ms/step
Epoch 91/100
39/39 - 0s - loss: 0.0479 - mae: 0.1631 - val_loss: 0.0649 - val_mae: 0.1897 -
474ms/epoch - 12ms/step
Epoch 92/100
39/39 - 0s - loss: 0.0469 - mae: 0.1623 - val_loss: 0.0362 - val_mae: 0.1333 -
466ms/epoch - 12ms/step
Epoch 93/100
39/39 - 0s - loss: 0.0370 - mae: 0.1418 - val_loss: 0.0294 - val_mae: 0.1103 -
382ms/epoch - 10ms/step
Epoch 94/100
39/39 - 0s - loss: 0.0367 - mae: 0.1388 - val_loss: 0.0424 - val_mae: 0.1489 -
```

```
469ms/epoch - 12ms/step
Epoch 95/100
39/39 - 0s - loss: 0.0390 - mae: 0.1430 - val_loss: 0.0347 - val_mae: 0.1264 -
358ms/epoch - 9ms/step
Epoch 96/100
39/39 - 0s - loss: 0.0402 - mae: 0.1469 - val_loss: 0.0367 - val_mae: 0.1350 -
361ms/epoch - 9ms/step
Epoch 97/100
39/39 - 0s - loss: 0.0365 - mae: 0.1420 - val_loss: 0.0522 - val_mae: 0.1821 -
493ms/epoch - 13ms/step
Epoch 98/100
39/39 - 0s - loss: 0.0403 - mae: 0.1488 - val_loss: 0.0417 - val_mae: 0.1445 -
473ms/epoch - 12ms/step
Epoch 99/100
39/39 - 0s - loss: 0.0397 - mae: 0.1458 - val_loss: 0.0301 - val_mae: 0.1184 -
496ms/epoch - 13ms/step
Epoch 100/100
39/39 - 1s - loss: 0.0370 - mae: 0.1400 - val_loss: 0.0304 - val_mae: 0.1220 -
525ms/epoch - 13ms/step
```

1.13 Make Predictions

[128]:
```
RNN_Predictions = RNN_Model.predict(X_test)
LSTM_predictions = LSTM_Model.predict(X_test)
```

1.14 Inverse Transform the Values

[129]:
```
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)

[130]:
```
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)}")
```

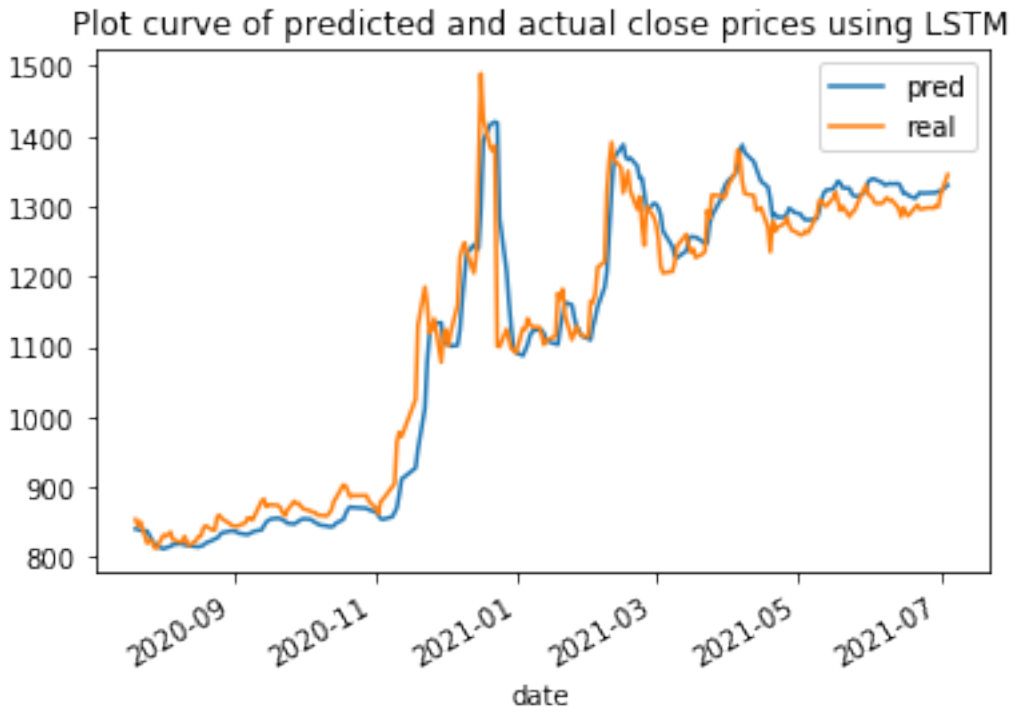
```
### RNN Model ###
RMSE 38.98487522765638
MAE 24.33409862029247
```

[131]:
```
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)}")
```

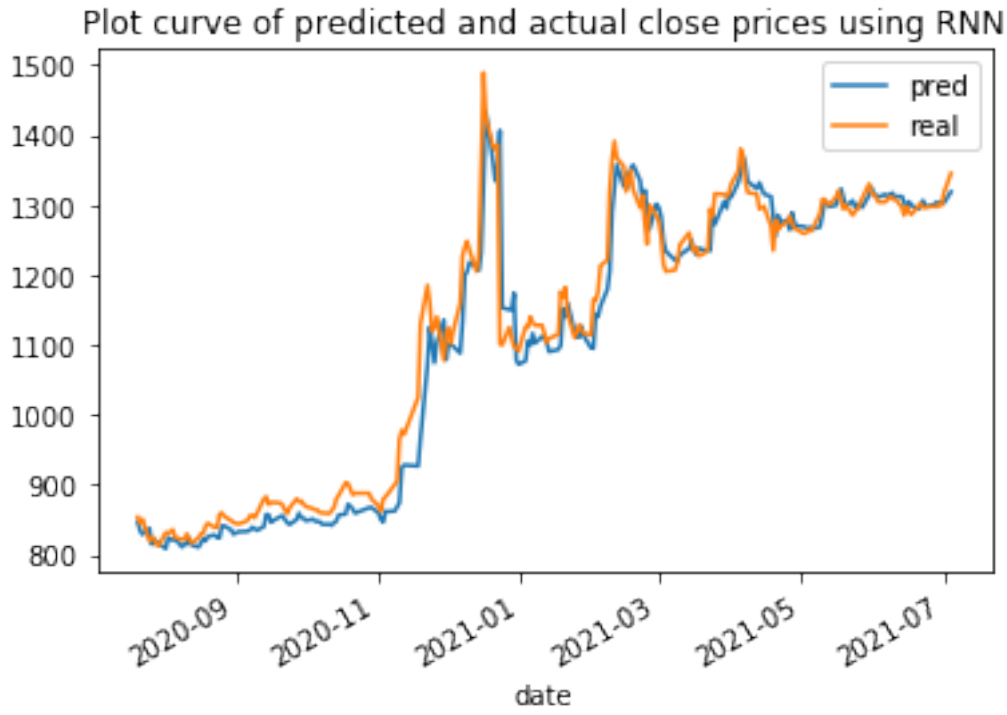
```
### LSTM Model ###
RMSE 46.422508767438416
MAE 30.296848655765892
```

[136]:
```
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()
```

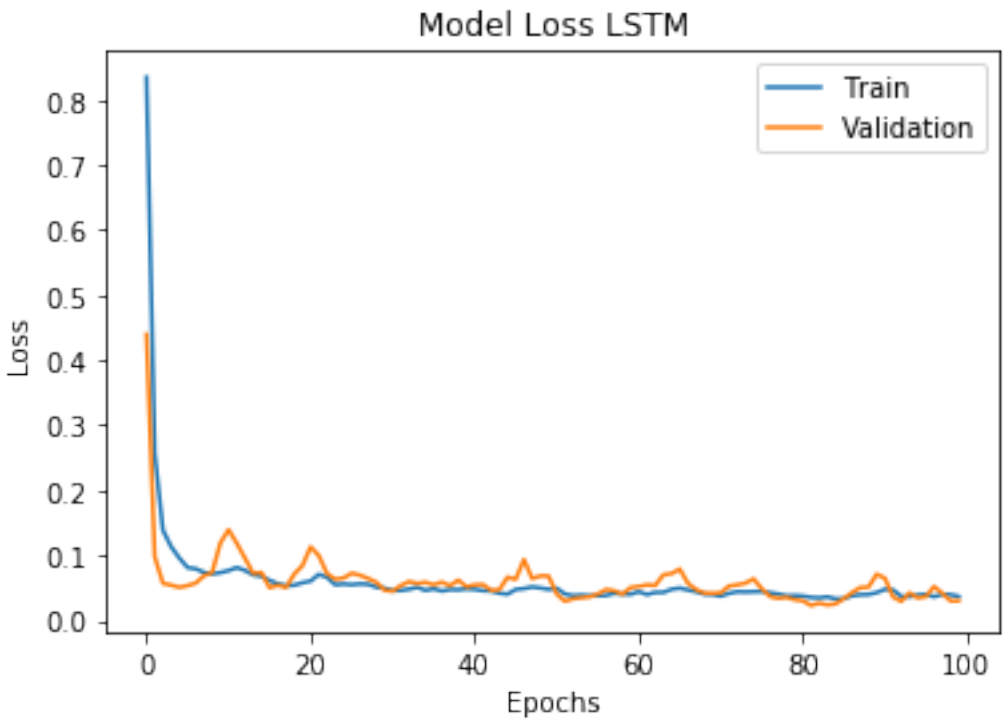


[137]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
```

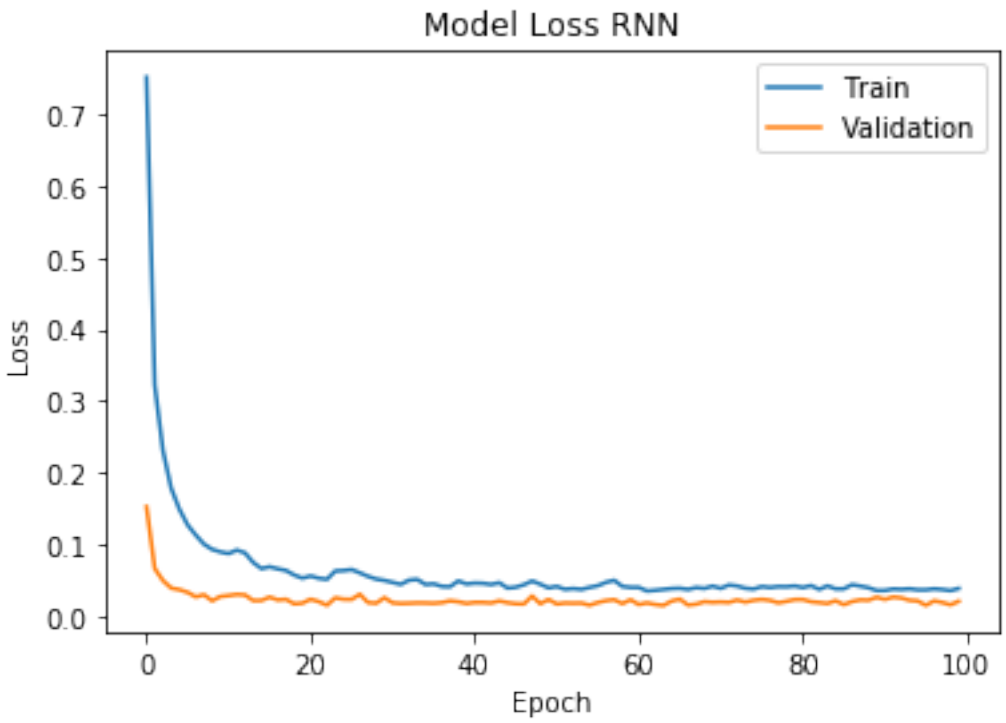
```
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



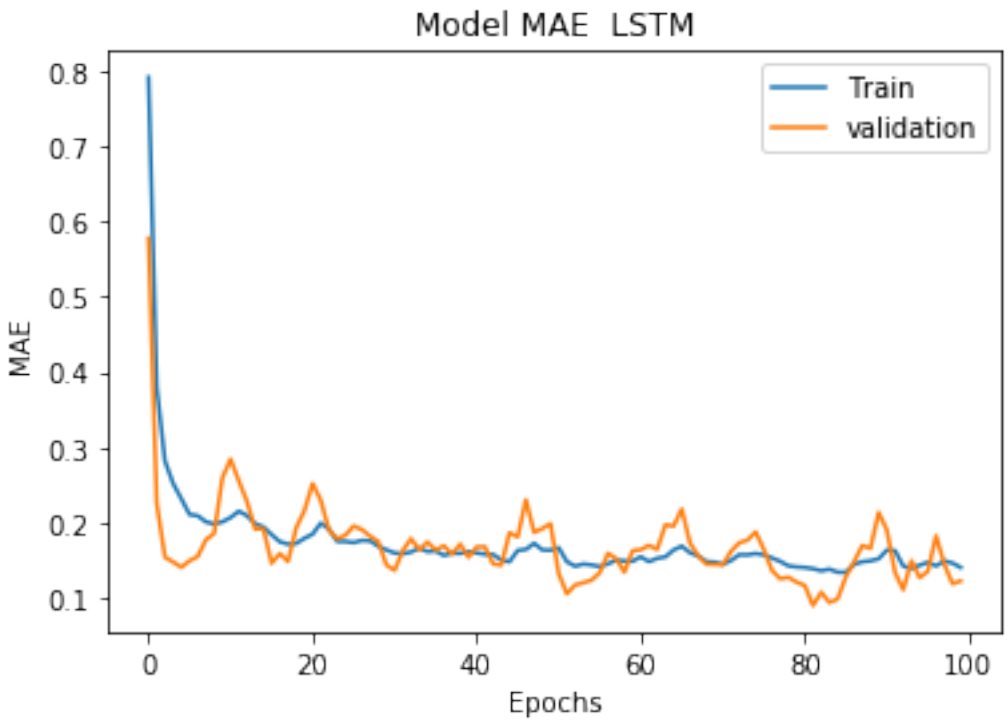
[147]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



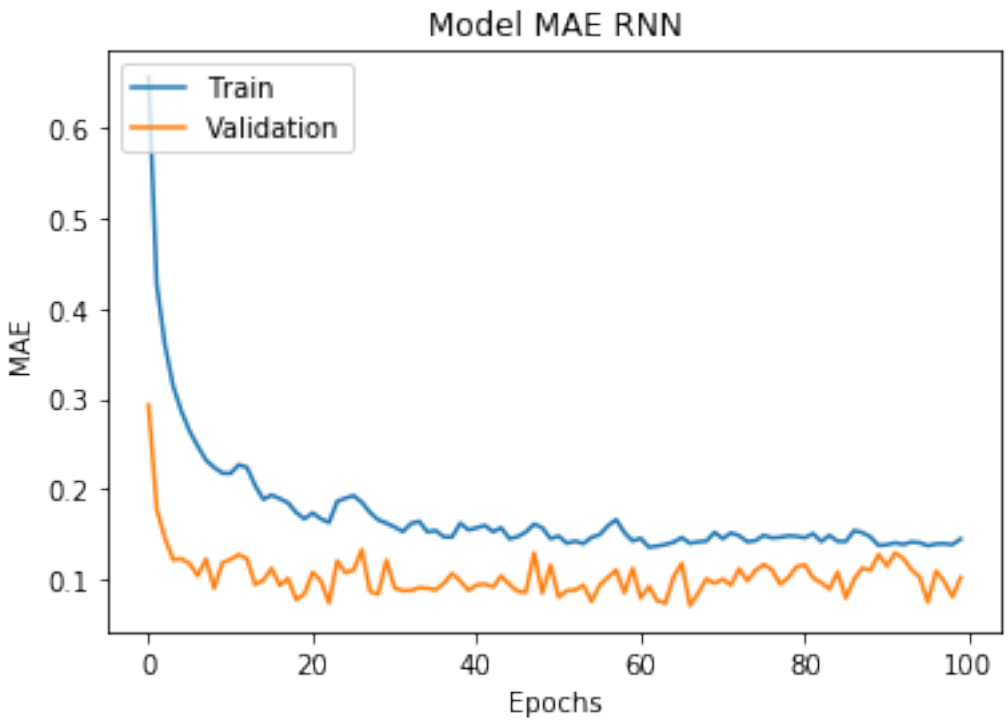
[148]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[149]:
```
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()
```



[150]:
```
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 NABIL Bank LSTM and RNN Models used for Stock Price Prediction
2. The Error is Low for RNN Model

Assignment4_Dev_Bank_LBBL

July 30, 2022

1 Stock Price Prediction of LBLL

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 05:13:12.061456: 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 05:13:12.061495: 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]:
```
lbbl_df = pd.read_csv("data/LBBL.csv")
lbbl_df.shape
```

[3]:
```
(1877, 8)
```

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

[4]:

	S.N.	Date	Total Transactions	Total Traded Shares	\
0	1	2021-12-29	196	37355.0	
1	2	2021-12-28	293	55587.0	
2	3	2021-12-27	407	86428.0	
3	4	2021-12-26	375	79223.0	

```
4     5  2021-12-23                 403              74536.0

   Total Traded Amount  Max. Price  Min. Price  Close Price
0           16232734.3       445.7       427.0        430.2
1           24245227.0       448.0       422.0        437.0
2           37957307.7       461.0       427.0        429.9
3           36089623.0       465.0       446.0        453.0
4           34006761.4       468.0       441.1        451.0
```

1.3 Renaming the Columns

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

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

[6]:

	SN	Date	TTrans	TTS	TTA	MaxPrice	MinPrice	ClosePrice
0	1	2021-12-29	196	37355.0	16232734.3	445.7	427.0	430.2
1	2	2021-12-28	293	55587.0	24245227.0	448.0	422.0	437.0
2	3	2021-12-27	407	86428.0	37957307.7	461.0	427.0	429.9
3	4	2021-12-26	375	79223.0	36089623.0	465.0	446.0	453.0
4	5	2021-12-23	403	74536.0	34006761.4	468.0	441.1	451.0

[7]:
```
lbbl_df.shape
```

[7]:
```
(1877, 8)
```

Converting the Date into Panda's Date Time

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

1.4 Sorting the Date by Date in Ascending Order

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

1.5 Setting Features and Target Column

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

[11]:
```
X = lbbl_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 05:13:14.661463: 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 05:13:14.661532: W
tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit:
```

```
UNKNOWN ERROR (303)
2022-07-30 05:13:14.661565: 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 05:13:14.661966: 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
30/30 - 3s - loss: 1.0524 - mae: 0.6914 - val_loss: 0.3640 - val_mae: 0.4768 -
3s/epoch - 86ms/step
Epoch 2/100
30/30 - 0s - loss: 0.6069 - mae: 0.5161 - val_loss: 0.2064 - val_mae: 0.3568 -
179ms/epoch - 6ms/step
Epoch 3/100
30/30 - 0s - loss: 0.4353 - mae: 0.4373 - val_loss: 0.1949 - val_mae: 0.3505 -
177ms/epoch - 6ms/step
Epoch 4/100
30/30 - 0s - loss: 0.3702 - mae: 0.4073 - val_loss: 0.1735 - val_mae: 0.3328 -
169ms/epoch - 6ms/step
Epoch 5/100
30/30 - 0s - loss: 0.3258 - mae: 0.3792 - val_loss: 0.1375 - val_mae: 0.2954 -
171ms/epoch - 6ms/step
Epoch 6/100
30/30 - 0s - loss: 0.2902 - mae: 0.3597 - val_loss: 0.1453 - val_mae: 0.3037 -
171ms/epoch - 6ms/step
Epoch 7/100
30/30 - 0s - loss: 0.2701 - mae: 0.3388 - val_loss: 0.1040 - val_mae: 0.2553 -
165ms/epoch - 5ms/step
Epoch 8/100
30/30 - 0s - loss: 0.2334 - mae: 0.3168 - val_loss: 0.0894 - val_mae: 0.2379 -
170ms/epoch - 6ms/step
```

```
Epoch 9/100
30/30 - 0s - loss: 0.1983 - mae: 0.2946 - val_loss: 0.0790 - val_mae: 0.2253 -
172ms/epoch - 6ms/step
Epoch 10/100
30/30 - 0s - loss: 0.1922 - mae: 0.2869 - val_loss: 0.0850 - val_mae: 0.2349 -
166ms/epoch - 6ms/step
Epoch 11/100
30/30 - 0s - loss: 0.1691 - mae: 0.2774 - val_loss: 0.0833 - val_mae: 0.2326 -
164ms/epoch - 5ms/step
Epoch 12/100
30/30 - 0s - loss: 0.1727 - mae: 0.2735 - val_loss: 0.0739 - val_mae: 0.2189 -
169ms/epoch - 6ms/step
Epoch 13/100
30/30 - 0s - loss: 0.1638 - mae: 0.2670 - val_loss: 0.0773 - val_mae: 0.2260 -
226ms/epoch - 8ms/step
Epoch 14/100
30/30 - 0s - loss: 0.1450 - mae: 0.2551 - val_loss: 0.0657 - val_mae: 0.2083 -
221ms/epoch - 7ms/step
Epoch 15/100
30/30 - 0s - loss: 0.1349 - mae: 0.2469 - val_loss: 0.0590 - val_mae: 0.1971 -
159ms/epoch - 5ms/step
Epoch 16/100
30/30 - 0s - loss: 0.1307 - mae: 0.2401 - val_loss: 0.0489 - val_mae: 0.1807 -
177ms/epoch - 6ms/step
Epoch 17/100
30/30 - 0s - loss: 0.1212 - mae: 0.2302 - val_loss: 0.0463 - val_mae: 0.1757 -
223ms/epoch - 7ms/step
Epoch 18/100
30/30 - 0s - loss: 0.1203 - mae: 0.2302 - val_loss: 0.0524 - val_mae: 0.1857 -
206ms/epoch - 7ms/step
Epoch 19/100
30/30 - 0s - loss: 0.1244 - mae: 0.2293 - val_loss: 0.0520 - val_mae: 0.1853 -
169ms/epoch - 6ms/step
Epoch 20/100
30/30 - 0s - loss: 0.1192 - mae: 0.2285 - val_loss: 0.0527 - val_mae: 0.1861 -
166ms/epoch - 6ms/step
Epoch 21/100
30/30 - 0s - loss: 0.1121 - mae: 0.2251 - val_loss: 0.0529 - val_mae: 0.1865 -
180ms/epoch - 6ms/step
Epoch 22/100
30/30 - 0s - loss: 0.1117 - mae: 0.2250 - val_loss: 0.0563 - val_mae: 0.1918 -
231ms/epoch - 8ms/step
Epoch 23/100
30/30 - 0s - loss: 0.1105 - mae: 0.2225 - val_loss: 0.0590 - val_mae: 0.1969 -
192ms/epoch - 6ms/step
Epoch 24/100
30/30 - 0s - loss: 0.1044 - mae: 0.2188 - val_loss: 0.0500 - val_mae: 0.1809 -
219ms/epoch - 7ms/step
```

```
Epoch 25/100
30/30 - 0s - loss: 0.0985 - mae: 0.2122 - val_loss: 0.0521 - val_mae: 0.1854 -
203ms/epoch - 7ms/step
Epoch 26/100
30/30 - 0s - loss: 0.0982 - mae: 0.2113 - val_loss: 0.0568 - val_mae: 0.1931 -
263ms/epoch - 9ms/step
Epoch 27/100
30/30 - 0s - loss: 0.0970 - mae: 0.2129 - val_loss: 0.0539 - val_mae: 0.1880 -
171ms/epoch - 6ms/step
Epoch 28/100
30/30 - 0s - loss: 0.0970 - mae: 0.2062 - val_loss: 0.0470 - val_mae: 0.1751 -
163ms/epoch - 5ms/step
Epoch 29/100
30/30 - 0s - loss: 0.0928 - mae: 0.2092 - val_loss: 0.0625 - val_mae: 0.2025 -
215ms/epoch - 7ms/step
Epoch 30/100
30/30 - 0s - loss: 0.0989 - mae: 0.2117 - val_loss: 0.0677 - val_mae: 0.2116 -
181ms/epoch - 6ms/step
Epoch 31/100
30/30 - 0s - loss: 0.0975 - mae: 0.2123 - val_loss: 0.0659 - val_mae: 0.2075 -
166ms/epoch - 6ms/step
Epoch 32/100
30/30 - 0s - loss: 0.0869 - mae: 0.1989 - val_loss: 0.0517 - val_mae: 0.1842 -
162ms/epoch - 5ms/step
Epoch 33/100
30/30 - 0s - loss: 0.0873 - mae: 0.1978 - val_loss: 0.0463 - val_mae: 0.1746 -
164ms/epoch - 5ms/step
Epoch 34/100
30/30 - 0s - loss: 0.0843 - mae: 0.1942 - val_loss: 0.0536 - val_mae: 0.1874 -
169ms/epoch - 6ms/step
Epoch 35/100
30/30 - 0s - loss: 0.0892 - mae: 0.2026 - val_loss: 0.0585 - val_mae: 0.1964 -
170ms/epoch - 6ms/step
Epoch 36/100
30/30 - 0s - loss: 0.0933 - mae: 0.2061 - val_loss: 0.0704 - val_mae: 0.2170 -
168ms/epoch - 6ms/step
Epoch 37/100
30/30 - 0s - loss: 0.0881 - mae: 0.2033 - val_loss: 0.0595 - val_mae: 0.1973 -
162ms/epoch - 5ms/step
Epoch 38/100
30/30 - 0s - loss: 0.0796 - mae: 0.1891 - val_loss: 0.0458 - val_mae: 0.1744 -
163ms/epoch - 5ms/step
Epoch 39/100
30/30 - 0s - loss: 0.0816 - mae: 0.1944 - val_loss: 0.0537 - val_mae: 0.1883 -
166ms/epoch - 6ms/step
Epoch 40/100
30/30 - 0s - loss: 0.0894 - mae: 0.1988 - val_loss: 0.0534 - val_mae: 0.1885 -
174ms/epoch - 6ms/step
```

```
Epoch 41/100
30/30 - 0s - loss: 0.0817 - mae: 0.1921 - val_loss: 0.0613 - val_mae: 0.2010 -
165ms/epoch - 6ms/step
Epoch 42/100
30/30 - 0s - loss: 0.0876 - mae: 0.1981 - val_loss: 0.0762 - val_mae: 0.2250 -
168ms/epoch - 6ms/step
Epoch 43/100
30/30 - 0s - loss: 0.0916 - mae: 0.1999 - val_loss: 0.0610 - val_mae: 0.2009 -
189ms/epoch - 6ms/step
Epoch 44/100
30/30 - 0s - loss: 0.0807 - mae: 0.1948 - val_loss: 0.0639 - val_mae: 0.2050 -
217ms/epoch - 7ms/step
Epoch 45/100
30/30 - 0s - loss: 0.0845 - mae: 0.1966 - val_loss: 0.0696 - val_mae: 0.2147 -
167ms/epoch - 6ms/step
Epoch 46/100
30/30 - 0s - loss: 0.0860 - mae: 0.2001 - val_loss: 0.0662 - val_mae: 0.2098 -
187ms/epoch - 6ms/step
Epoch 47/100
30/30 - 0s - loss: 0.0847 - mae: 0.1958 - val_loss: 0.0735 - val_mae: 0.2213 -
205ms/epoch - 7ms/step
Epoch 48/100
30/30 - 0s - loss: 0.0788 - mae: 0.1897 - val_loss: 0.0608 - val_mae: 0.1995 -
195ms/epoch - 7ms/step
Epoch 49/100
30/30 - 0s - loss: 0.0768 - mae: 0.1885 - val_loss: 0.0604 - val_mae: 0.1983 -
200ms/epoch - 7ms/step
Epoch 50/100
30/30 - 0s - loss: 0.0847 - mae: 0.1968 - val_loss: 0.0563 - val_mae: 0.1936 -
219ms/epoch - 7ms/step
Epoch 51/100
30/30 - 0s - loss: 0.0772 - mae: 0.1865 - val_loss: 0.0523 - val_mae: 0.1847 -
189ms/epoch - 6ms/step
Epoch 52/100
30/30 - 0s - loss: 0.0769 - mae: 0.1803 - val_loss: 0.0457 - val_mae: 0.1739 -
163ms/epoch - 5ms/step
Epoch 53/100
30/30 - 0s - loss: 0.0727 - mae: 0.1801 - val_loss: 0.0453 - val_mae: 0.1721 -
201ms/epoch - 7ms/step
Epoch 54/100
30/30 - 0s - loss: 0.0723 - mae: 0.1802 - val_loss: 0.0584 - val_mae: 0.1978 -
192ms/epoch - 6ms/step
Epoch 55/100
30/30 - 0s - loss: 0.0834 - mae: 0.1943 - val_loss: 0.0628 - val_mae: 0.2052 -
165ms/epoch - 5ms/step
Epoch 56/100
30/30 - 0s - loss: 0.0745 - mae: 0.1828 - val_loss: 0.0556 - val_mae: 0.1924 -
199ms/epoch - 7ms/step
```

```
Epoch 57/100
30/30 - 0s - loss: 0.0717 - mae: 0.1824 - val_loss: 0.0618 - val_mae: 0.2041 -
222ms/epoch - 7ms/step
Epoch 58/100
30/30 - 0s - loss: 0.0818 - mae: 0.1918 - val_loss: 0.0628 - val_mae: 0.2041 -
179ms/epoch - 6ms/step
Epoch 59/100
30/30 - 0s - loss: 0.0812 - mae: 0.1897 - val_loss: 0.0666 - val_mae: 0.2118 -
213ms/epoch - 7ms/step
Epoch 60/100
30/30 - 0s - loss: 0.0736 - mae: 0.1830 - val_loss: 0.0577 - val_mae: 0.1948 -
177ms/epoch - 6ms/step
Epoch 61/100
30/30 - 0s - loss: 0.0776 - mae: 0.1824 - val_loss: 0.0580 - val_mae: 0.1962 -
182ms/epoch - 6ms/step
Epoch 62/100
30/30 - 0s - loss: 0.0670 - mae: 0.1729 - val_loss: 0.0559 - val_mae: 0.1912 -
186ms/epoch - 6ms/step
Epoch 63/100
30/30 - 0s - loss: 0.0729 - mae: 0.1843 - val_loss: 0.0645 - val_mae: 0.2077 -
169ms/epoch - 6ms/step
Epoch 64/100
30/30 - 0s - loss: 0.0795 - mae: 0.1944 - val_loss: 0.0623 - val_mae: 0.2041 -
246ms/epoch - 8ms/step
Epoch 65/100
30/30 - 0s - loss: 0.0736 - mae: 0.1801 - val_loss: 0.0564 - val_mae: 0.1947 -
237ms/epoch - 8ms/step
Epoch 66/100
30/30 - 0s - loss: 0.0733 - mae: 0.1832 - val_loss: 0.0497 - val_mae: 0.1829 -
177ms/epoch - 6ms/step
Epoch 67/100
30/30 - 0s - loss: 0.0697 - mae: 0.1774 - val_loss: 0.0554 - val_mae: 0.1925 -
159ms/epoch - 5ms/step
Epoch 68/100
30/30 - 0s - loss: 0.0697 - mae: 0.1768 - val_loss: 0.0457 - val_mae: 0.1745 -
157ms/epoch - 5ms/step
Epoch 69/100
30/30 - 0s - loss: 0.0692 - mae: 0.1756 - val_loss: 0.0494 - val_mae: 0.1821 -
161ms/epoch - 5ms/step
Epoch 70/100
30/30 - 0s - loss: 0.0637 - mae: 0.1680 - val_loss: 0.0420 - val_mae: 0.1670 -
159ms/epoch - 5ms/step
Epoch 71/100
30/30 - 0s - loss: 0.0635 - mae: 0.1701 - val_loss: 0.0376 - val_mae: 0.1578 -
160ms/epoch - 5ms/step
Epoch 72/100
30/30 - 0s - loss: 0.0673 - mae: 0.1686 - val_loss: 0.0378 - val_mae: 0.1581 -
165ms/epoch - 6ms/step
```

```
Epoch 73/100
30/30 - 0s - loss: 0.0658 - mae: 0.1647 - val_loss: 0.0413 - val_mae: 0.1652 -
160ms/epoch - 5ms/step
Epoch 74/100
30/30 - 0s - loss: 0.0649 - mae: 0.1702 - val_loss: 0.0437 - val_mae: 0.1698 -
156ms/epoch - 5ms/step
Epoch 75/100
30/30 - 0s - loss: 0.0614 - mae: 0.1625 - val_loss: 0.0469 - val_mae: 0.1756 -
157ms/epoch - 5ms/step
Epoch 76/100
30/30 - 0s - loss: 0.0664 - mae: 0.1696 - val_loss: 0.0455 - val_mae: 0.1739 -
175ms/epoch - 6ms/step
Epoch 77/100
30/30 - 0s - loss: 0.0612 - mae: 0.1652 - val_loss: 0.0463 - val_mae: 0.1744 -
175ms/epoch - 6ms/step
Epoch 78/100
30/30 - 0s - loss: 0.0634 - mae: 0.1696 - val_loss: 0.0499 - val_mae: 0.1822 -
168ms/epoch - 6ms/step
Epoch 79/100
30/30 - 0s - loss: 0.0652 - mae: 0.1724 - val_loss: 0.0543 - val_mae: 0.1909 -
335ms/epoch - 11ms/step
Epoch 80/100
30/30 - 0s - loss: 0.0658 - mae: 0.1750 - val_loss: 0.0566 - val_mae: 0.1932 -
177ms/epoch - 6ms/step
Epoch 81/100
30/30 - 0s - loss: 0.0704 - mae: 0.1727 - val_loss: 0.0504 - val_mae: 0.1828 -
159ms/epoch - 5ms/step
Epoch 82/100
30/30 - 0s - loss: 0.0616 - mae: 0.1647 - val_loss: 0.0505 - val_mae: 0.1827 -
218ms/epoch - 7ms/step
Epoch 83/100
30/30 - 0s - loss: 0.0678 - mae: 0.1773 - val_loss: 0.0680 - val_mae: 0.2112 -
201ms/epoch - 7ms/step
Epoch 84/100
30/30 - 0s - loss: 0.0636 - mae: 0.1733 - val_loss: 0.0571 - val_mae: 0.1935 -
159ms/epoch - 5ms/step
Epoch 85/100
30/30 - 0s - loss: 0.0632 - mae: 0.1717 - val_loss: 0.0631 - val_mae: 0.2025 -
196ms/epoch - 7ms/step
Epoch 86/100
30/30 - 0s - loss: 0.0683 - mae: 0.1812 - val_loss: 0.0639 - val_mae: 0.2047 -
197ms/epoch - 7ms/step
Epoch 87/100
30/30 - 0s - loss: 0.0728 - mae: 0.1812 - val_loss: 0.0696 - val_mae: 0.2137 -
194ms/epoch - 6ms/step
Epoch 88/100
30/30 - 0s - loss: 0.0723 - mae: 0.1778 - val_loss: 0.0696 - val_mae: 0.2128 -
165ms/epoch - 5ms/step
```

```
Epoch 89/100
30/30 - 0s - loss: 0.0692 - mae: 0.1807 - val_loss: 0.0716 - val_mae: 0.2159 -
297ms/epoch - 10ms/step
Epoch 90/100
30/30 - 0s - loss: 0.0687 - mae: 0.1772 - val_loss: 0.0637 - val_mae: 0.2047 -
384ms/epoch - 13ms/step
Epoch 91/100
30/30 - 0s - loss: 0.0732 - mae: 0.1824 - val_loss: 0.0740 - val_mae: 0.2191 -
240ms/epoch - 8ms/step
Epoch 92/100
30/30 - 0s - loss: 0.0749 - mae: 0.1866 - val_loss: 0.0760 - val_mae: 0.2263 -
334ms/epoch - 11ms/step
Epoch 93/100
30/30 - 0s - loss: 0.0716 - mae: 0.1825 - val_loss: 0.0753 - val_mae: 0.2200 -
178ms/epoch - 6ms/step
Epoch 94/100
30/30 - 0s - loss: 0.0729 - mae: 0.1823 - val_loss: 0.0599 - val_mae: 0.1998 -
314ms/epoch - 10ms/step
Epoch 95/100
30/30 - 0s - loss: 0.0687 - mae: 0.1776 - val_loss: 0.0609 - val_mae: 0.1980 -
178ms/epoch - 6ms/step
Epoch 96/100
30/30 - 0s - loss: 0.0714 - mae: 0.1796 - val_loss: 0.0663 - val_mae: 0.2099 -
164ms/epoch - 5ms/step
Epoch 97/100
30/30 - 0s - loss: 0.0651 - mae: 0.1780 - val_loss: 0.0596 - val_mae: 0.1982 -
159ms/epoch - 5ms/step
Epoch 98/100
30/30 - 0s - loss: 0.0714 - mae: 0.1767 - val_loss: 0.0506 - val_mae: 0.1834 -
167ms/epoch - 6ms/step
Epoch 99/100
30/30 - 0s - loss: 0.0712 - mae: 0.1781 - val_loss: 0.0473 - val_mae: 0.1783 -
170ms/epoch - 6ms/step
Epoch 100/100
30/30 - 0s - loss: 0.0624 - mae: 0.1684 - val_loss: 0.0385 - val_mae: 0.1605 -
161ms/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
30/30 - 6s - loss: 0.9566 - mae: 0.7277 - val_loss: 0.7364 - val_mae: 0.7396 -
6s/epoch - 190ms/step
Epoch 2/100
30/30 - 0s - loss: 0.6934 - mae: 0.5664 - val_loss: 0.2728 - val_mae: 0.4135 -
283ms/epoch - 9ms/step
Epoch 3/100
```

```
30/30 - 0s - loss: 0.3847 - mae: 0.4055 - val_loss: 0.1676 - val_mae: 0.3280 -
358ms/epoch - 12ms/step
Epoch 4/100
30/30 - 0s - loss: 0.2639 - mae: 0.3316 - val_loss: 0.1400 - val_mae: 0.2979 -
388ms/epoch - 13ms/step
Epoch 5/100
30/30 - 0s - loss: 0.2232 - mae: 0.3083 - val_loss: 0.1230 - val_mae: 0.2776 -
316ms/epoch - 11ms/step
Epoch 6/100
30/30 - 0s - loss: 0.1844 - mae: 0.2795 - val_loss: 0.1167 - val_mae: 0.2708 -
286ms/epoch - 10ms/step
Epoch 7/100
30/30 - 0s - loss: 0.1689 - mae: 0.2736 - val_loss: 0.1097 - val_mae: 0.2606 -
295ms/epoch - 10ms/step
Epoch 8/100
30/30 - 0s - loss: 0.1600 - mae: 0.2655 - val_loss: 0.0981 - val_mae: 0.2445 -
343ms/epoch - 11ms/step
Epoch 9/100
30/30 - 0s - loss: 0.1365 - mae: 0.2443 - val_loss: 0.0934 - val_mae: 0.2381 -
366ms/epoch - 12ms/step
Epoch 10/100
30/30 - 0s - loss: 0.1257 - mae: 0.2340 - val_loss: 0.0867 - val_mae: 0.2268 -
378ms/epoch - 13ms/step
Epoch 11/100
30/30 - 0s - loss: 0.1140 - mae: 0.2227 - val_loss: 0.0817 - val_mae: 0.2180 -
317ms/epoch - 11ms/step
Epoch 12/100
30/30 - 0s - loss: 0.1069 - mae: 0.2169 - val_loss: 0.0781 - val_mae: 0.2120 -
375ms/epoch - 13ms/step
Epoch 13/100
30/30 - 0s - loss: 0.0977 - mae: 0.2060 - val_loss: 0.0739 - val_mae: 0.2001 -
370ms/epoch - 12ms/step
Epoch 14/100
30/30 - 0s - loss: 0.0933 - mae: 0.2001 - val_loss: 0.0730 - val_mae: 0.2022 -
323ms/epoch - 11ms/step
Epoch 15/100
30/30 - 0s - loss: 0.0913 - mae: 0.1979 - val_loss: 0.0738 - val_mae: 0.2064 -
358ms/epoch - 12ms/step
Epoch 16/100
30/30 - 0s - loss: 0.0849 - mae: 0.1942 - val_loss: 0.0679 - val_mae: 0.1856 -
324ms/epoch - 11ms/step
Epoch 17/100
30/30 - 0s - loss: 0.0898 - mae: 0.1962 - val_loss: 0.0667 - val_mae: 0.1849 -
316ms/epoch - 11ms/step
Epoch 18/100
30/30 - 0s - loss: 0.0855 - mae: 0.1957 - val_loss: 0.0776 - val_mae: 0.2174 -
288ms/epoch - 10ms/step
Epoch 19/100
```

```
30/30 - 0s - loss: 0.0916 - mae: 0.1992 - val_loss: 0.0688 - val_mae: 0.1862 -
293ms/epoch - 10ms/step
Epoch 20/100
30/30 - 0s - loss: 0.0838 - mae: 0.1913 - val_loss: 0.0667 - val_mae: 0.1930 -
291ms/epoch - 10ms/step
Epoch 21/100
30/30 - 0s - loss: 0.0768 - mae: 0.1853 - val_loss: 0.0643 - val_mae: 0.1803 -
286ms/epoch - 10ms/step
Epoch 22/100
30/30 - 0s - loss: 0.0791 - mae: 0.1846 - val_loss: 0.0690 - val_mae: 0.2009 -
290ms/epoch - 10ms/step
Epoch 23/100
30/30 - 0s - loss: 0.0862 - mae: 0.1916 - val_loss: 0.0730 - val_mae: 0.2089 -
420ms/epoch - 14ms/step
Epoch 24/100
30/30 - 0s - loss: 0.0893 - mae: 0.2005 - val_loss: 0.0928 - val_mae: 0.2465 -
300ms/epoch - 10ms/step
Epoch 25/100
30/30 - 0s - loss: 0.0929 - mae: 0.2035 - val_loss: 0.0631 - val_mae: 0.1766 -
303ms/epoch - 10ms/step
Epoch 26/100
30/30 - 0s - loss: 0.0782 - mae: 0.1821 - val_loss: 0.0629 - val_mae: 0.1778 -
428ms/epoch - 14ms/step
Epoch 27/100
30/30 - 0s - loss: 0.0745 - mae: 0.1800 - val_loss: 0.0650 - val_mae: 0.1950 -
439ms/epoch - 15ms/step
Epoch 28/100
30/30 - 0s - loss: 0.0754 - mae: 0.1826 - val_loss: 0.0653 - val_mae: 0.1949 -
354ms/epoch - 12ms/step
Epoch 29/100
30/30 - 0s - loss: 0.0782 - mae: 0.1845 - val_loss: 0.0592 - val_mae: 0.1687 -
368ms/epoch - 12ms/step
Epoch 30/100
30/30 - 0s - loss: 0.0737 - mae: 0.1773 - val_loss: 0.0714 - val_mae: 0.2101 -
301ms/epoch - 10ms/step
Epoch 31/100
30/30 - 0s - loss: 0.0827 - mae: 0.1859 - val_loss: 0.0630 - val_mae: 0.1840 -
324ms/epoch - 11ms/step
Epoch 32/100
30/30 - 0s - loss: 0.0803 - mae: 0.1897 - val_loss: 0.0900 - val_mae: 0.2393 -
498ms/epoch - 17ms/step
Epoch 33/100
30/30 - 1s - loss: 0.0858 - mae: 0.1965 - val_loss: 0.0585 - val_mae: 0.1677 -
572ms/epoch - 19ms/step
Epoch 34/100
30/30 - 0s - loss: 0.0744 - mae: 0.1809 - val_loss: 0.0813 - val_mae: 0.2269 -
401ms/epoch - 13ms/step
Epoch 35/100
```

```
30/30 - 0s - loss: 0.0853 - mae: 0.1942 - val_loss: 0.0677 - val_mae: 0.1984 -
356ms/epoch - 12ms/step
Epoch 36/100
30/30 - 0s - loss: 0.0784 - mae: 0.1869 - val_loss: 0.0847 - val_mae: 0.2310 -
343ms/epoch - 11ms/step
Epoch 37/100
30/30 - 0s - loss: 0.0867 - mae: 0.1978 - val_loss: 0.0602 - val_mae: 0.1768 -
426ms/epoch - 14ms/step
Epoch 38/100
30/30 - 0s - loss: 0.0725 - mae: 0.1786 - val_loss: 0.0785 - val_mae: 0.2232 -
345ms/epoch - 11ms/step
Epoch 39/100
30/30 - 0s - loss: 0.0849 - mae: 0.1987 - val_loss: 0.0559 - val_mae: 0.1659 -
311ms/epoch - 10ms/step
Epoch 40/100
30/30 - 0s - loss: 0.0696 - mae: 0.1775 - val_loss: 0.0720 - val_mae: 0.2147 -
430ms/epoch - 14ms/step
Epoch 41/100
30/30 - 0s - loss: 0.0711 - mae: 0.1778 - val_loss: 0.0563 - val_mae: 0.1717 -
296ms/epoch - 10ms/step
Epoch 42/100
30/30 - 0s - loss: 0.0679 - mae: 0.1715 - val_loss: 0.0653 - val_mae: 0.2027 -
398ms/epoch - 13ms/step
Epoch 43/100
30/30 - 0s - loss: 0.0722 - mae: 0.1788 - val_loss: 0.0655 - val_mae: 0.1968 -
392ms/epoch - 13ms/step
Epoch 44/100
30/30 - 0s - loss: 0.0705 - mae: 0.1807 - val_loss: 0.0989 - val_mae: 0.2490 -
296ms/epoch - 10ms/step
Epoch 45/100
30/30 - 0s - loss: 0.0916 - mae: 0.2116 - val_loss: 0.0615 - val_mae: 0.1841 -
305ms/epoch - 10ms/step
Epoch 46/100
30/30 - 0s - loss: 0.0672 - mae: 0.1711 - val_loss: 0.0571 - val_mae: 0.1886 -
346ms/epoch - 12ms/step
Epoch 47/100
30/30 - 0s - loss: 0.0734 - mae: 0.1786 - val_loss: 0.0532 - val_mae: 0.1707 -
444ms/epoch - 15ms/step
Epoch 48/100
30/30 - 0s - loss: 0.0701 - mae: 0.1736 - val_loss: 0.0629 - val_mae: 0.1982 -
368ms/epoch - 12ms/step
Epoch 49/100
30/30 - 0s - loss: 0.0729 - mae: 0.1743 - val_loss: 0.0500 - val_mae: 0.1583 -
443ms/epoch - 15ms/step
Epoch 50/100
30/30 - 0s - loss: 0.0611 - mae: 0.1640 - val_loss: 0.0513 - val_mae: 0.1761 -
327ms/epoch - 11ms/step
Epoch 51/100
```

```
30/30 - 0s - loss: 0.0683 - mae: 0.1705 - val_loss: 0.0494 - val_mae: 0.1590 -
336ms/epoch - 11ms/step
Epoch 52/100
30/30 - 1s - loss: 0.0620 - mae: 0.1649 - val_loss: 0.0808 - val_mae: 0.2255 -
506ms/epoch - 17ms/step
Epoch 53/100
30/30 - 0s - loss: 0.0778 - mae: 0.1882 - val_loss: 0.0590 - val_mae: 0.1833 -
362ms/epoch - 12ms/step
Epoch 54/100
30/30 - 0s - loss: 0.0648 - mae: 0.1650 - val_loss: 0.0687 - val_mae: 0.2087 -
430ms/epoch - 14ms/step
Epoch 55/100
30/30 - 0s - loss: 0.0763 - mae: 0.1869 - val_loss: 0.0575 - val_mae: 0.1816 -
440ms/epoch - 15ms/step
Epoch 56/100
30/30 - 0s - loss: 0.0654 - mae: 0.1689 - val_loss: 0.0679 - val_mae: 0.2076 -
458ms/epoch - 15ms/step
Epoch 57/100
30/30 - 0s - loss: 0.0717 - mae: 0.1820 - val_loss: 0.0499 - val_mae: 0.1620 -
411ms/epoch - 14ms/step
Epoch 58/100
30/30 - 0s - loss: 0.0607 - mae: 0.1587 - val_loss: 0.0621 - val_mae: 0.1984 -
313ms/epoch - 10ms/step
Epoch 59/100
30/30 - 0s - loss: 0.0704 - mae: 0.1759 - val_loss: 0.0488 - val_mae: 0.1616 -
305ms/epoch - 10ms/step
Epoch 60/100
30/30 - 0s - loss: 0.0579 - mae: 0.1545 - val_loss: 0.0512 - val_mae: 0.1809 -
313ms/epoch - 10ms/step
Epoch 61/100
30/30 - 0s - loss: 0.0609 - mae: 0.1624 - val_loss: 0.0506 - val_mae: 0.1711 -
320ms/epoch - 11ms/step
Epoch 62/100
30/30 - 0s - loss: 0.0579 - mae: 0.1557 - val_loss: 0.0505 - val_mae: 0.1788 -
310ms/epoch - 10ms/step
Epoch 63/100
30/30 - 0s - loss: 0.0641 - mae: 0.1657 - val_loss: 0.0428 - val_mae: 0.1443 -
307ms/epoch - 10ms/step
Epoch 64/100
30/30 - 0s - loss: 0.0612 - mae: 0.1581 - val_loss: 0.0463 - val_mae: 0.1687 -
343ms/epoch - 11ms/step
Epoch 65/100
30/30 - 0s - loss: 0.0624 - mae: 0.1633 - val_loss: 0.0410 - val_mae: 0.1456 -
321ms/epoch - 11ms/step
Epoch 66/100
30/30 - 0s - loss: 0.0552 - mae: 0.1535 - val_loss: 0.0451 - val_mae: 0.1644 -
310ms/epoch - 10ms/step
Epoch 67/100
```

```
30/30 - 0s - loss: 0.0562 - mae: 0.1548 - val_loss: 0.0382 - val_mae: 0.1381 -
305ms/epoch - 10ms/step
Epoch 68/100
30/30 - 0s - loss: 0.0562 - mae: 0.1529 - val_loss: 0.0410 - val_mae: 0.1509 -
298ms/epoch - 10ms/step
Epoch 69/100
30/30 - 0s - loss: 0.0614 - mae: 0.1660 - val_loss: 0.0636 - val_mae: 0.2006 -
303ms/epoch - 10ms/step
Epoch 70/100
30/30 - 0s - loss: 0.0740 - mae: 0.1836 - val_loss: 0.0422 - val_mae: 0.1444 -
310ms/epoch - 10ms/step
Epoch 71/100
30/30 - 0s - loss: 0.0522 - mae: 0.1495 - val_loss: 0.0448 - val_mae: 0.1700 -
308ms/epoch - 10ms/step
Epoch 72/100
30/30 - 0s - loss: 0.0582 - mae: 0.1548 - val_loss: 0.0437 - val_mae: 0.1581 -
307ms/epoch - 10ms/step
Epoch 73/100
30/30 - 0s - loss: 0.0552 - mae: 0.1534 - val_loss: 0.0450 - val_mae: 0.1699 -
309ms/epoch - 10ms/step
Epoch 74/100
30/30 - 0s - loss: 0.0581 - mae: 0.1592 - val_loss: 0.0389 - val_mae: 0.1420 -
300ms/epoch - 10ms/step
Epoch 75/100
30/30 - 0s - loss: 0.0639 - mae: 0.1682 - val_loss: 0.0675 - val_mae: 0.2102 -
301ms/epoch - 10ms/step
Epoch 76/100
30/30 - 0s - loss: 0.0706 - mae: 0.1816 - val_loss: 0.0454 - val_mae: 0.1586 -
308ms/epoch - 10ms/step
Epoch 77/100
30/30 - 0s - loss: 0.0573 - mae: 0.1554 - val_loss: 0.0424 - val_mae: 0.1639 -
298ms/epoch - 10ms/step
Epoch 78/100
30/30 - 0s - loss: 0.0660 - mae: 0.1671 - val_loss: 0.0702 - val_mae: 0.2094 -
301ms/epoch - 10ms/step
Epoch 79/100
30/30 - 0s - loss: 0.0727 - mae: 0.1822 - val_loss: 0.0895 - val_mae: 0.2323 -
294ms/epoch - 10ms/step
Epoch 80/100
30/30 - 0s - loss: 0.0924 - mae: 0.2171 - val_loss: 0.0537 - val_mae: 0.1768 -
300ms/epoch - 10ms/step
Epoch 81/100
30/30 - 0s - loss: 0.0634 - mae: 0.1627 - val_loss: 0.0569 - val_mae: 0.1897 -
294ms/epoch - 10ms/step
Epoch 82/100
30/30 - 0s - loss: 0.0699 - mae: 0.1745 - val_loss: 0.0364 - val_mae: 0.1332 -
313ms/epoch - 10ms/step
Epoch 83/100
```

```
30/30 - 0s - loss: 0.0574 - mae: 0.1555 - val_loss: 0.0338 - val_mae: 0.1285 -
313ms/epoch - 10ms/step
Epoch 84/100
30/30 - 0s - loss: 0.0559 - mae: 0.1582 - val_loss: 0.0447 - val_mae: 0.1693 -
299ms/epoch - 10ms/step
Epoch 85/100
30/30 - 0s - loss: 0.0603 - mae: 0.1632 - val_loss: 0.0352 - val_mae: 0.1344 -
306ms/epoch - 10ms/step
Epoch 86/100
30/30 - 0s - loss: 0.0511 - mae: 0.1456 - val_loss: 0.0328 - val_mae: 0.1312 -
302ms/epoch - 10ms/step
Epoch 87/100
30/30 - 0s - loss: 0.0479 - mae: 0.1438 - val_loss: 0.0329 - val_mae: 0.1354 -
299ms/epoch - 10ms/step
Epoch 88/100
30/30 - 0s - loss: 0.0484 - mae: 0.1425 - val_loss: 0.0324 - val_mae: 0.1296 -
326ms/epoch - 11ms/step
Epoch 89/100
30/30 - 0s - loss: 0.0523 - mae: 0.1465 - val_loss: 0.0351 - val_mae: 0.1457 -
303ms/epoch - 10ms/step
Epoch 90/100
30/30 - 0s - loss: 0.0548 - mae: 0.1491 - val_loss: 0.0341 - val_mae: 0.1331 -
293ms/epoch - 10ms/step
Epoch 91/100
30/30 - 0s - loss: 0.0482 - mae: 0.1436 - val_loss: 0.0389 - val_mae: 0.1567 -
294ms/epoch - 10ms/step
Epoch 92/100
30/30 - 0s - loss: 0.0584 - mae: 0.1596 - val_loss: 0.0334 - val_mae: 0.1321 -
310ms/epoch - 10ms/step
Epoch 93/100
30/30 - 0s - loss: 0.0483 - mae: 0.1443 - val_loss: 0.0330 - val_mae: 0.1377 -
311ms/epoch - 10ms/step
Epoch 94/100
30/30 - 0s - loss: 0.0516 - mae: 0.1494 - val_loss: 0.0317 - val_mae: 0.1271 -
302ms/epoch - 10ms/step
Epoch 95/100
30/30 - 0s - loss: 0.0475 - mae: 0.1417 - val_loss: 0.0305 - val_mae: 0.1270 -
293ms/epoch - 10ms/step
Epoch 96/100
30/30 - 0s - loss: 0.0531 - mae: 0.1462 - val_loss: 0.0313 - val_mae: 0.1314 -
314ms/epoch - 10ms/step
Epoch 97/100
30/30 - 0s - loss: 0.0496 - mae: 0.1430 - val_loss: 0.0307 - val_mae: 0.1274 -
292ms/epoch - 10ms/step
Epoch 98/100
30/30 - 0s - loss: 0.0472 - mae: 0.1392 - val_loss: 0.0305 - val_mae: 0.1292 -
296ms/epoch - 10ms/step
Epoch 99/100
```

```
30/30 - 0s - loss: 0.0488 - mae: 0.1438 - val_loss: 0.0324 - val_mae: 0.1365 -
326ms/epoch - 11ms/step
Epoch 100/100
30/30 - 0s - loss: 0.0486 - mae: 0.1429 - val_loss: 0.0305 - val_mae: 0.1260 -
285ms/epoch - 9ms/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)}")
```

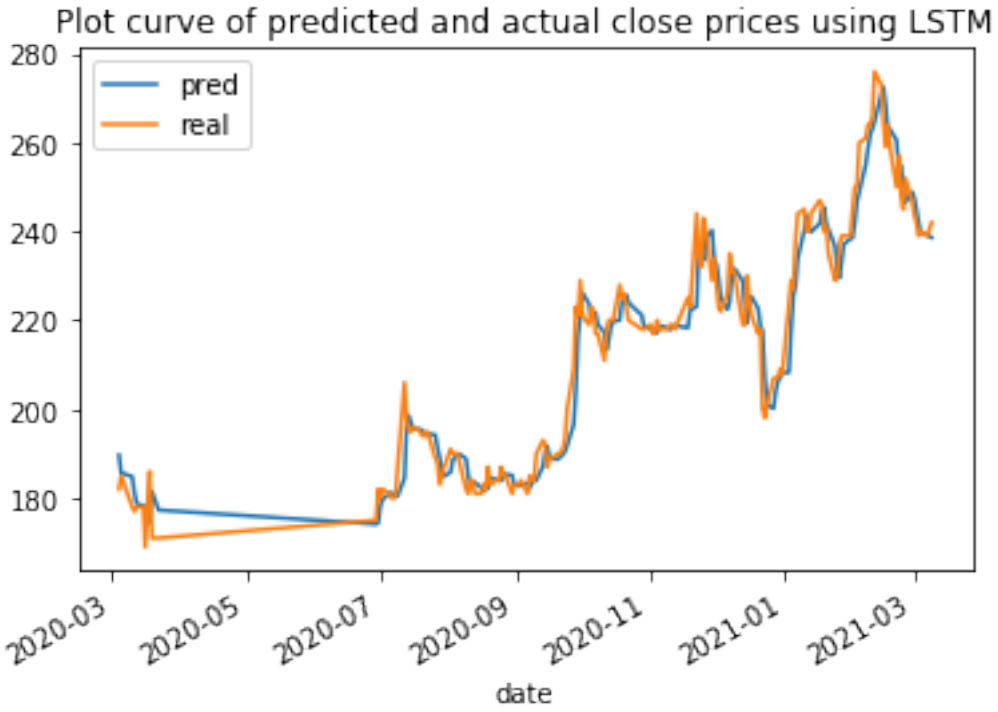
```
### RNN Model ###
RMSE 6.4697278074282405
MAE 5.187774324677681
```

[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)}")
```

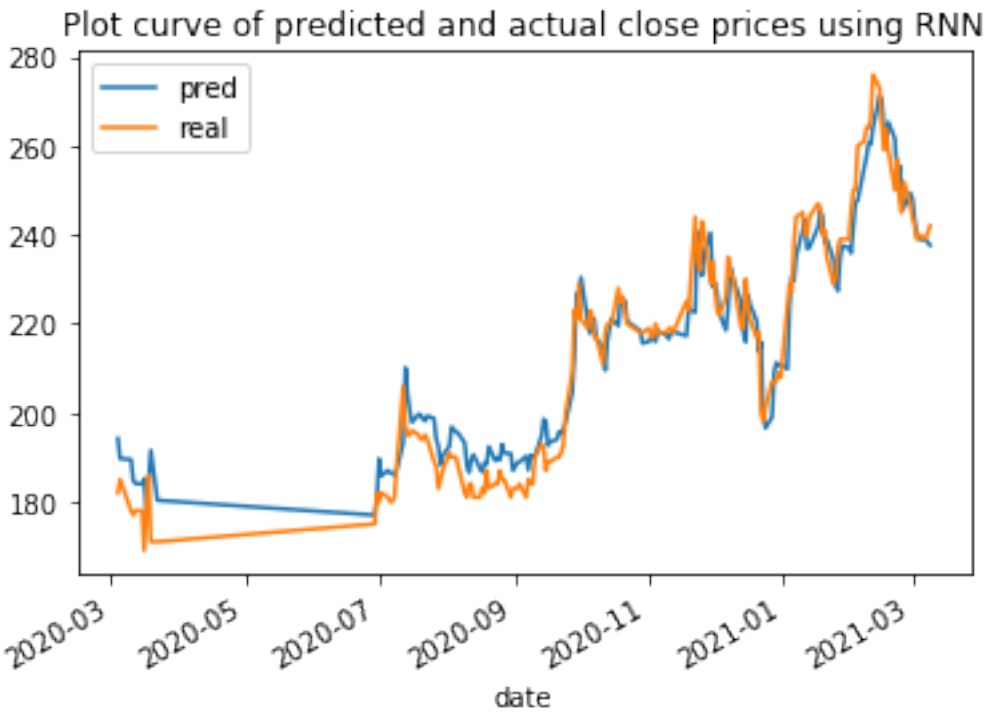
```
### LSTM Model ###
RMSE 5.652457303598954
MAE 3.958984041474556
```

[27]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
```

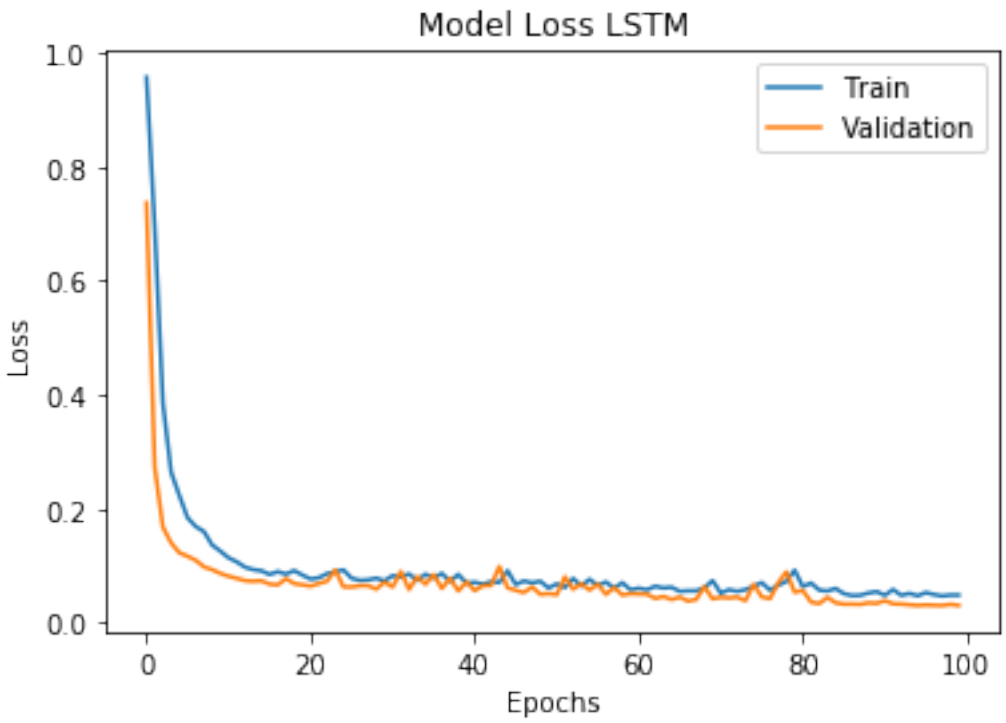
```
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



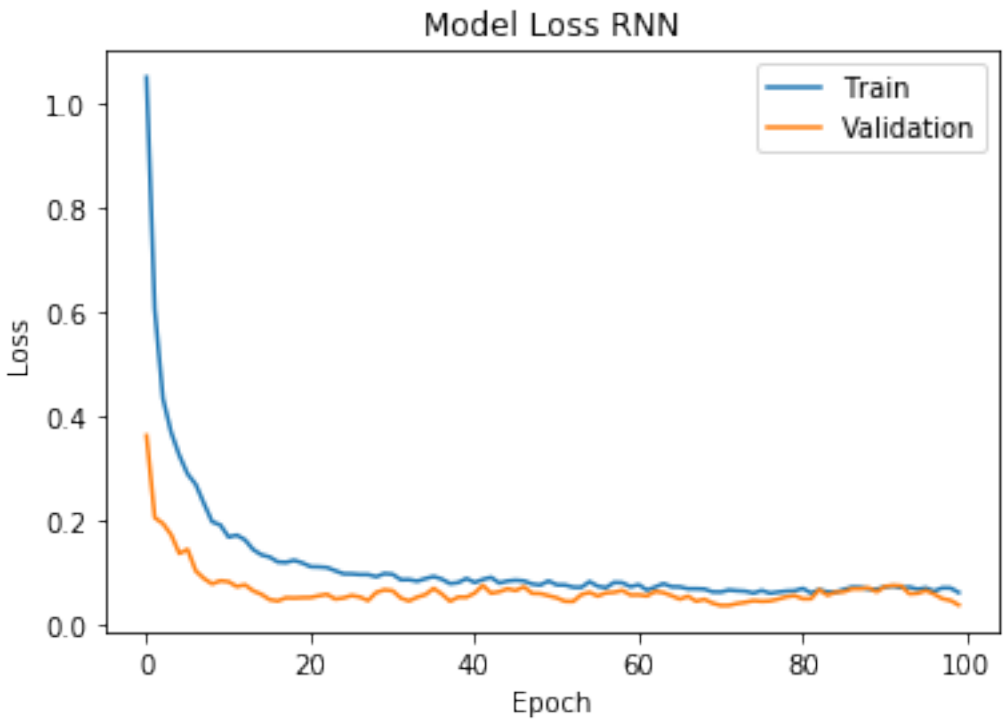
[28]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



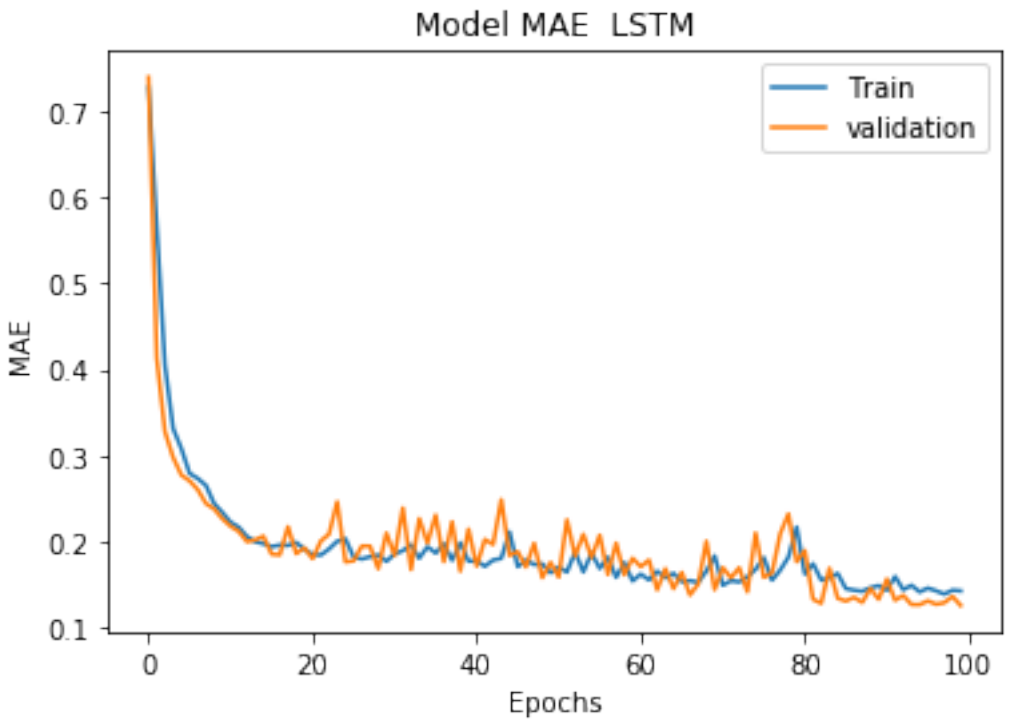
[29]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



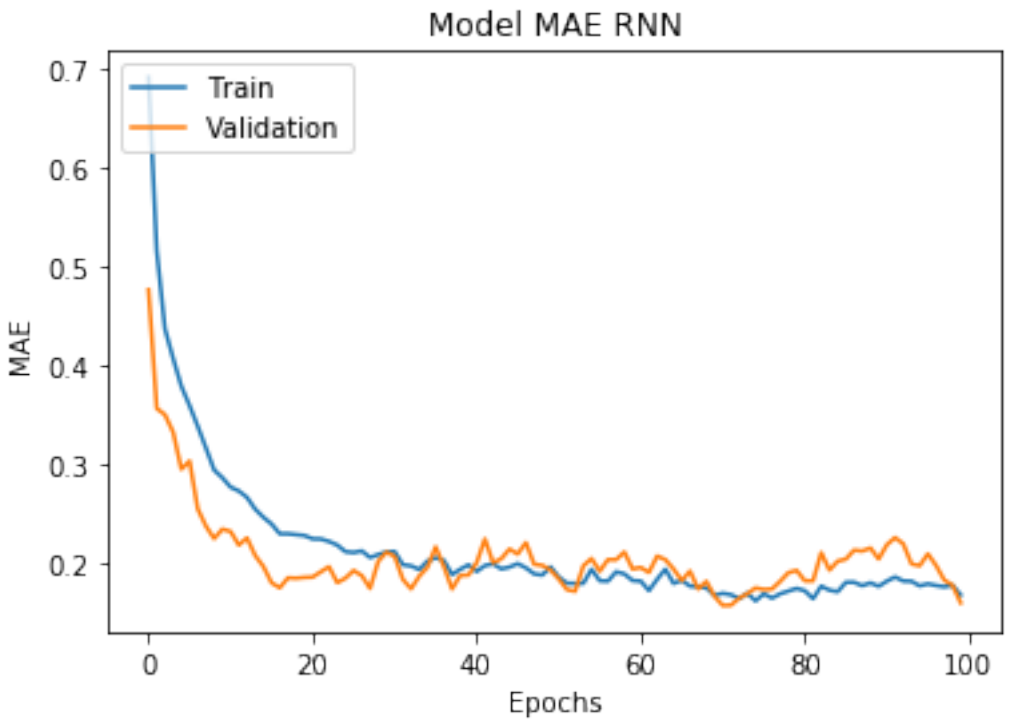
[30]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



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

[ ]:

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)}")
```

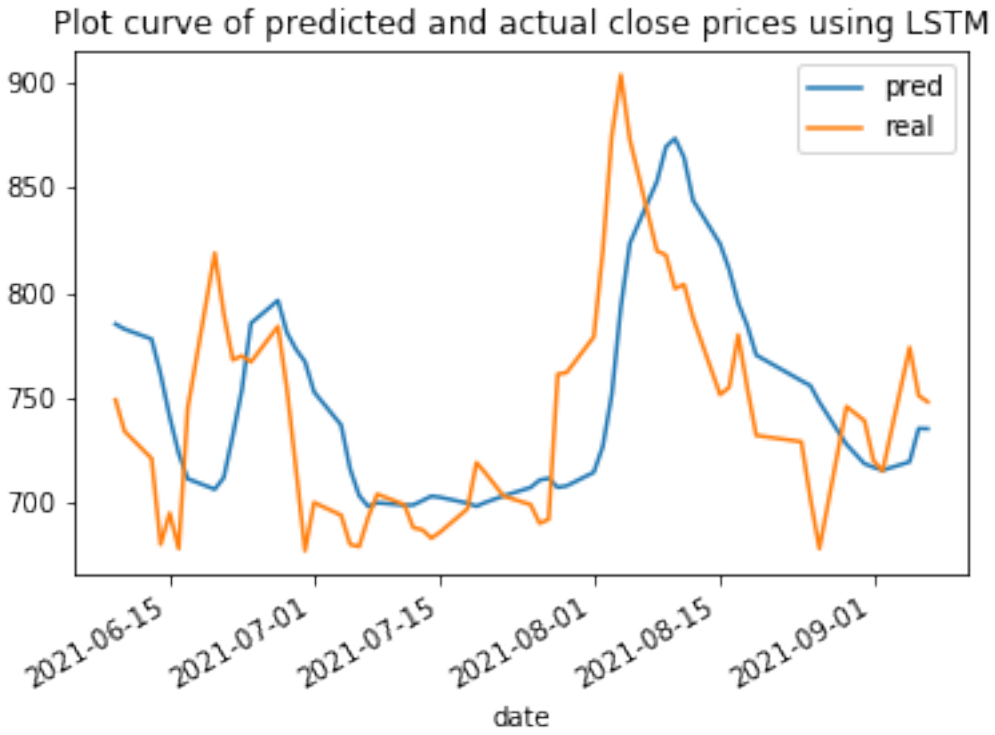
```
### RNN Model ###
RMSE 28.728600151721597
MAE 21.959531690253588
```

[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}"  )
```

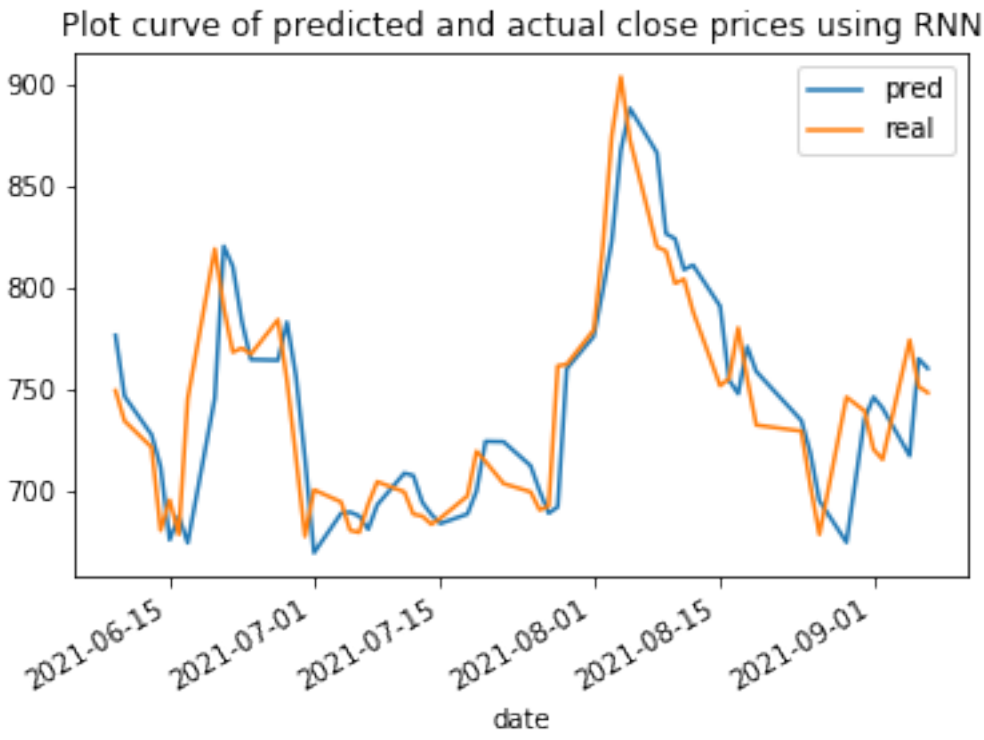
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

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

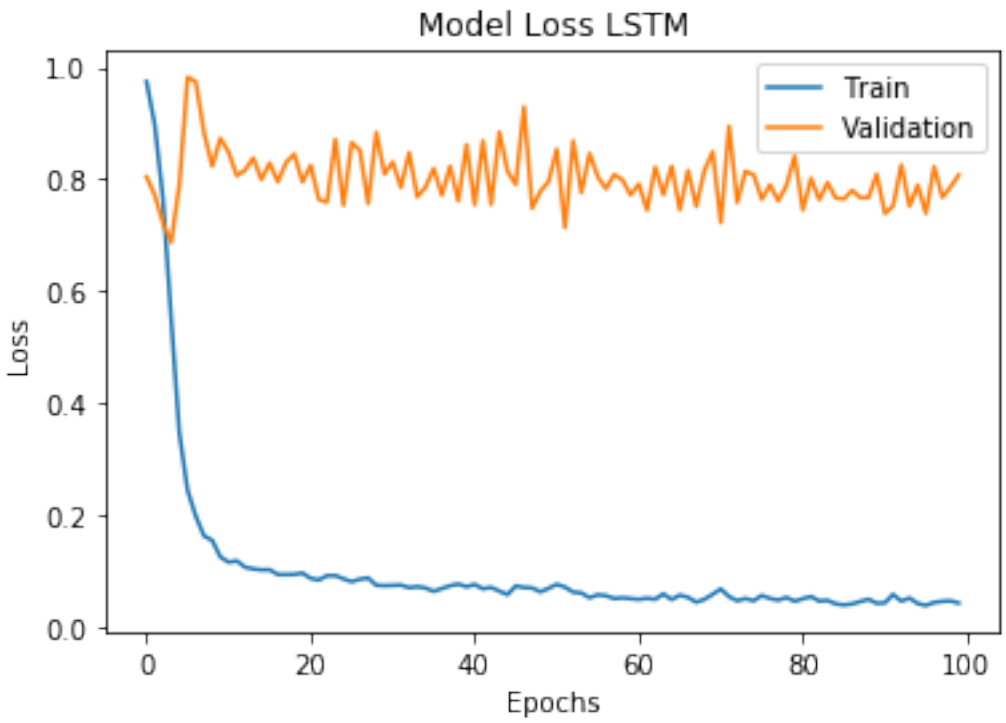
[27]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



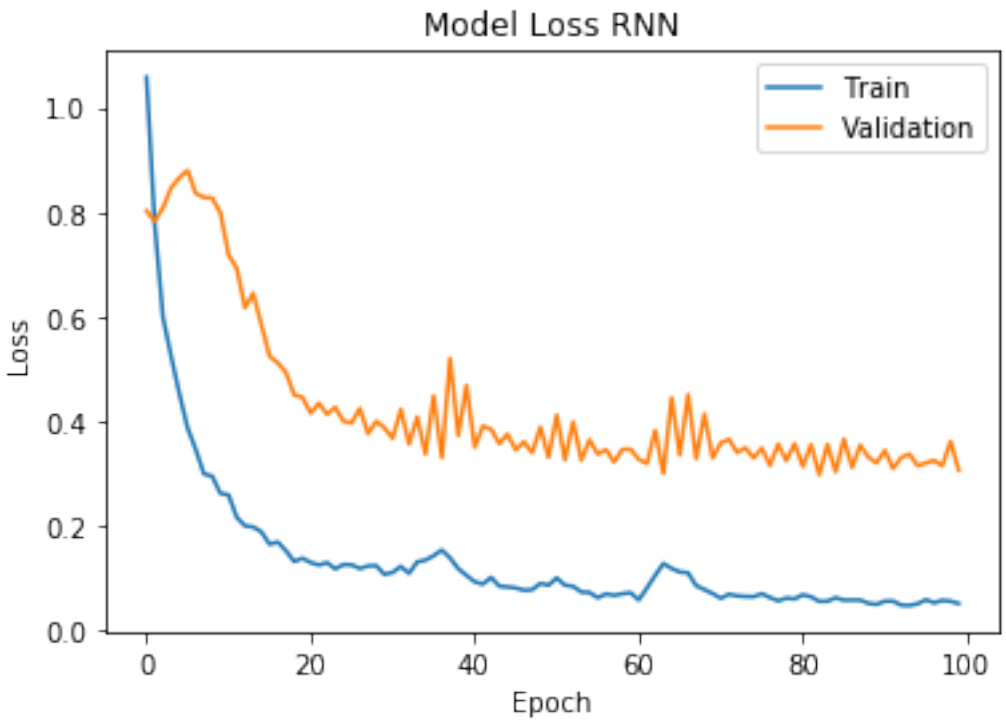
[28]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



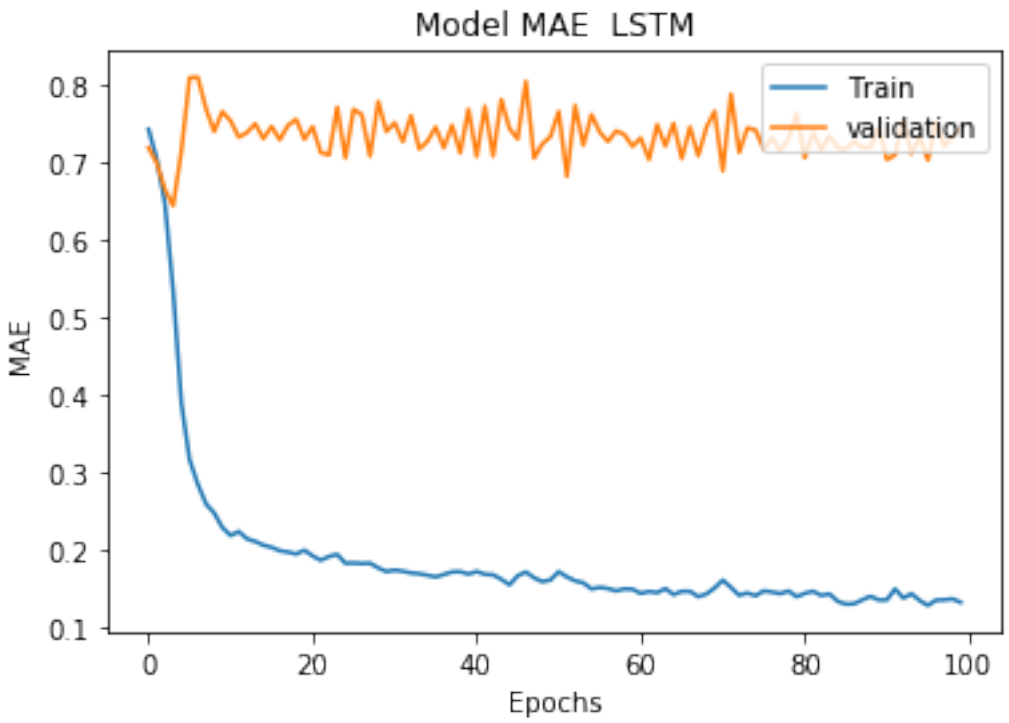
[29]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



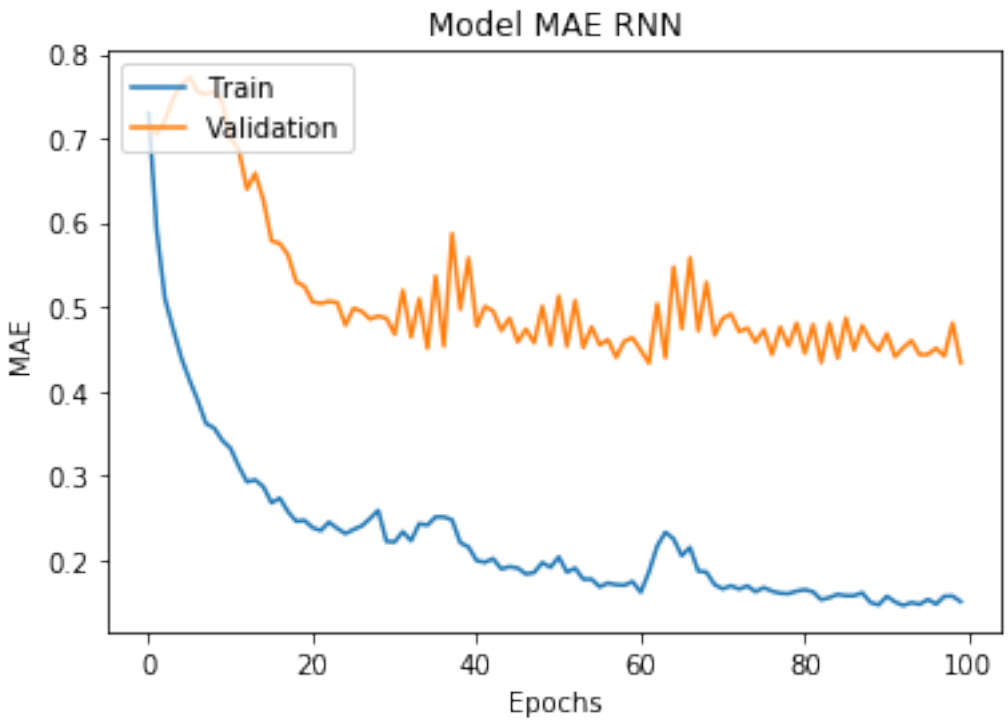
[30]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[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

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)}")
```

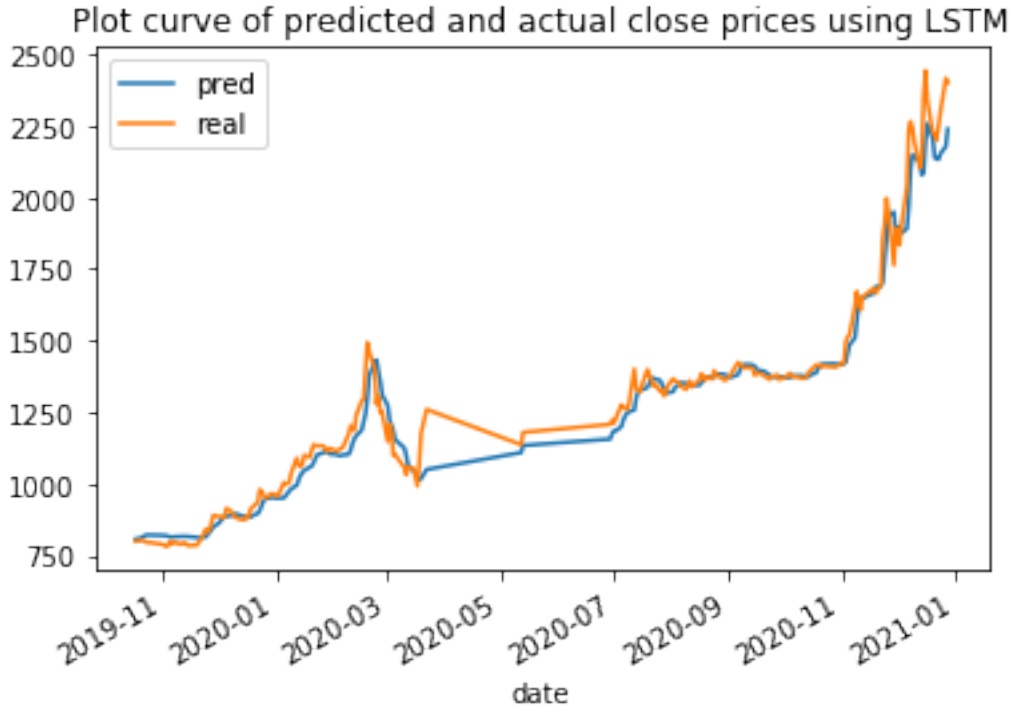
```
### RNN Model ###
RMSE 76.88429113784437
MAE 46.06268971100514
```

[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}"  )
```

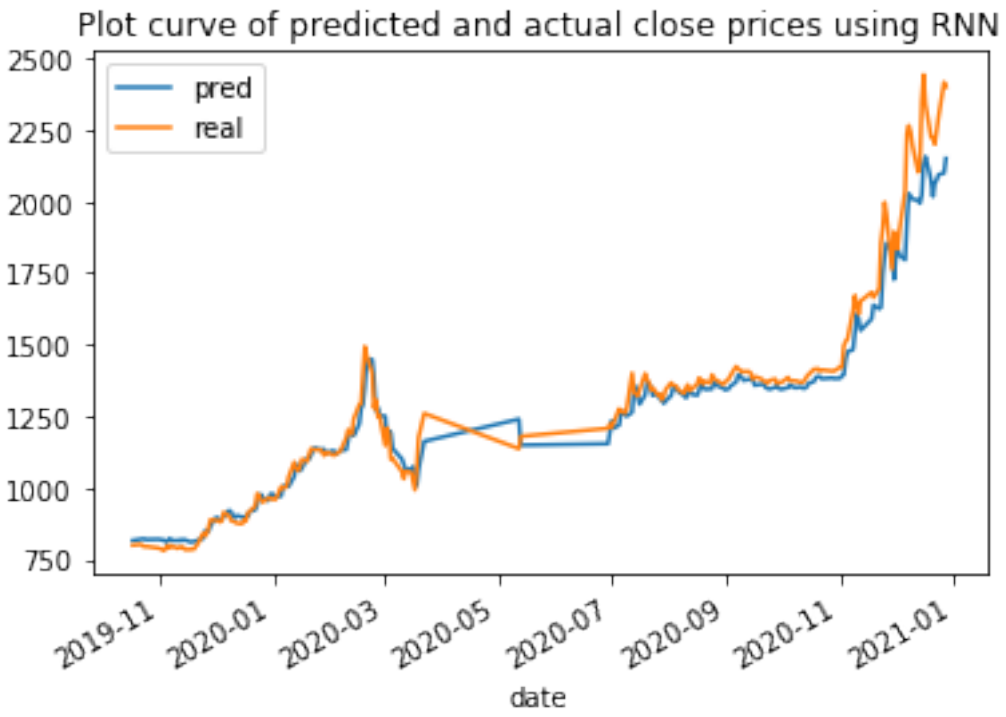
```
print(f"MAE {mean_absolute_error(Y_test_res_LSTM, pre_LSTM)}")
```

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

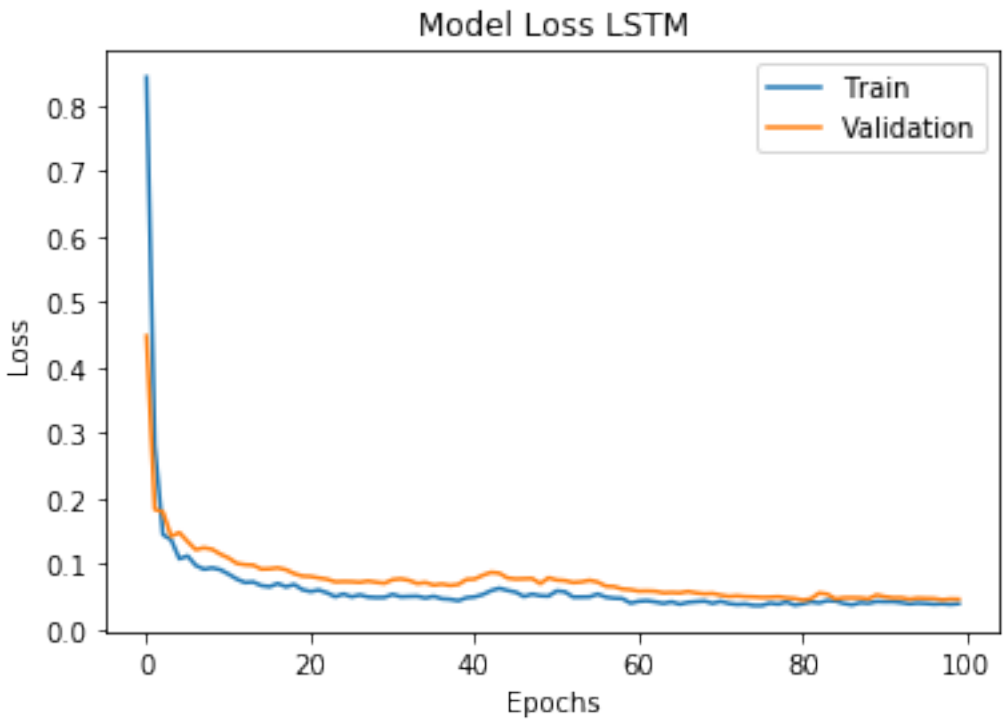
[31]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_LSTM))
plot["real"]=list(map(float, Y_test_res_LSTM))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using LSTM")
plt.show()
```



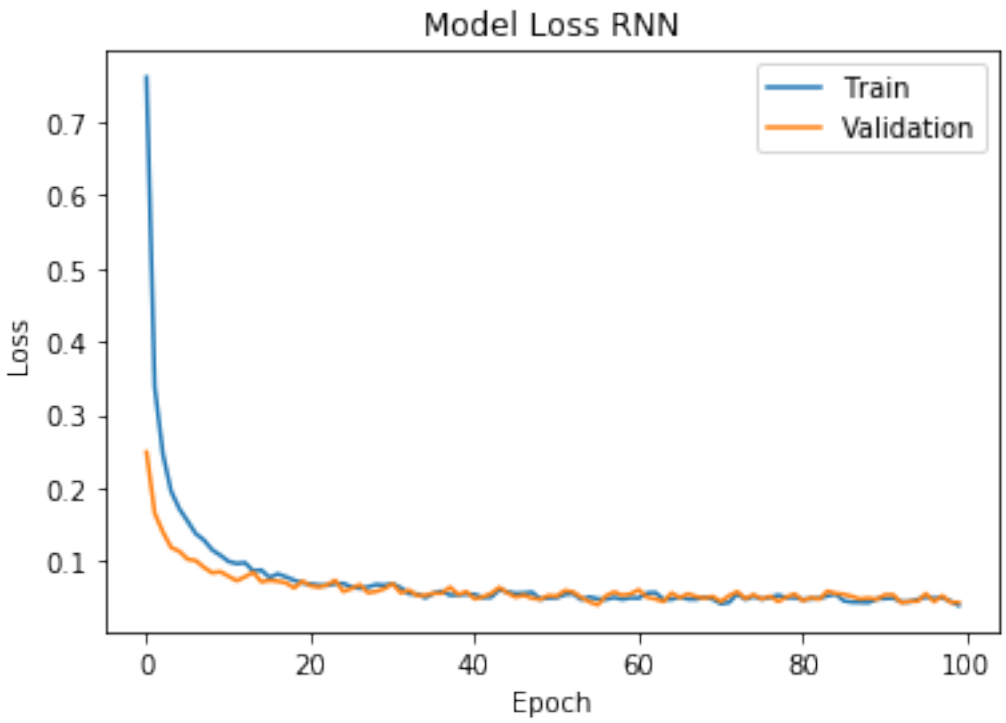
[32]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



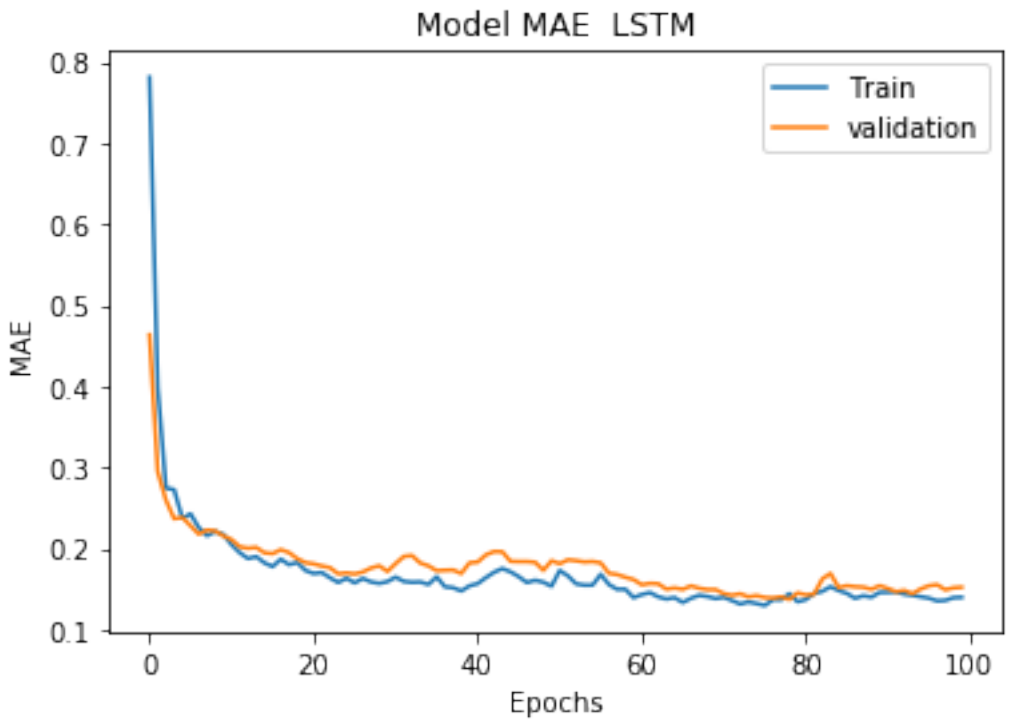
[33]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



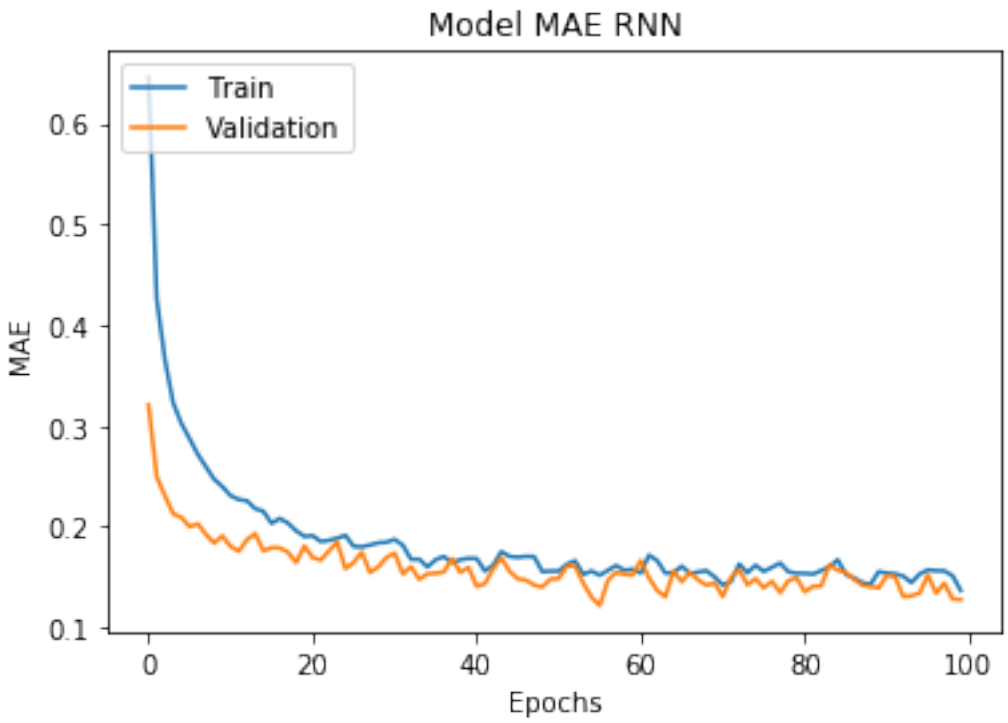
[34]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[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

Assignment4

July 29, 2022

1 Stock Price Prediction of NABIL BANK

1.1 Import the Required Libraries

[106]:
```
import warnings
warnings.filterwarnings('ignore')
```

[107]:
```
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
from matplotlib import pyplot as plt
# import Helper as hp
# import cufflinks as cf
# cf.set_config_file(theme='pearl',sharing='public',offline=True)
```

1.2 Load Data

[108]:
```
nabil_df = pd.read_csv("data/nabil.csv")
nabil_df.shape
```

[108]: (2392, 8)

[109]:
```
nabil_df.head()
```

[109]:

	Symbol	Date	Open	High	Low	Close	Percent Change	Volume
0	NABIL	2022-07-12	796.0	804.9	787.0	796.9	0.62	42720
1	NABIL	2022-07-11	831.0	831.0	792.0	792.0	-4.12	69864
2	NABIL	2022-07-08	845.0	847.0	825.5	826.0	-2.25	30318
3	NABIL	2022-07-07	864.0	886.0	844.0	845.0	-2.20	51271
4	NABIL	2022-07-06	885.0	889.0	837.0	864.0	-0.92	58061

1.3 Removing the Unwanted Columns

[110]:
```
nabil_df.drop(columns=['Symbol','Percent Change','Volume'],inplace=True)
```

[111]:
```
nabil_df.head()
```

[111]:

	Date	Open	High	Low	Close
0	2022-07-12	796.0	804.9	787.0	796.9
1	2022-07-11	831.0	831.0	792.0	792.0
2	2022-07-08	845.0	847.0	825.5	826.0
3	2022-07-07	864.0	886.0	844.0	845.0
4	2022-07-06	885.0	889.0	837.0	864.0

[112]:
```
nabil_df.shape
```

[112]:
```
(2392, 5)
```

Converting the Date into Panda's Date Time

[113]:
```
nabil_df['Date'] = pd.to_datetime(nabil_df['Date'])
```

1.4 Sorting the Date by Date in Ascending Order

[114]:
```
nabil_df=nabil_df.sort_values(by='Date')
```

1.5 Setting Features and Target Column

[115]:
```
features = ['Date','Close']
```

[116]:
```
X = nabil_df[features]
```

[117]:
```
X.set_index("Date",inplace=True)
```

1.6 Splitting the Data Into Training, Validation and Test Set

[118]:
```
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

[119]:
```
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

[120]:
```
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

[121]:
```
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

[122]:
```
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'])
```

[123]:
```
RNN_Model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_6 (SimpleRNN)    (None, 5, 50)             2600

 dropout_10 (Dropout)        (None, 5, 50)             0

 simple_rnn_7 (SimpleRNN)    (None, 50)                5050

 dropout_11 (Dropout)        (None, 50)                0

 dense_5 (Dense)             (None, 50)                2550

=================================================================
Total params: 10,200
Trainable params: 10,200
```

```
Non-trainable params: 0
_________________________________________________________________
```

1.11 Building LSTM Model

[124]:
```
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'])
```

[125]:
```
LSTM_Model.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 5, 50)             10400

 dropout_12 (Dropout)        (None, 5, 50)             0

 lstm_5 (LSTM)               (None, 50)                20200

 dropout_13 (Dropout)        (None, 50)                0

 dense_6 (Dense)             (None, 50)                2550

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

1.12 Fitting the RNN Model

[126]:
```
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
39/39 - 2s - loss: 0.7526 - mae: 0.6569 - val_loss: 0.1520 - val_mae: 0.2931 -
2s/epoch - 55ms/step
Epoch 2/100
39/39 - 0s - loss: 0.3223 - mae: 0.4272 - val_loss: 0.0657 - val_mae: 0.1782 -
225ms/epoch - 6ms/step
Epoch 3/100
```

```
39/39 - 0s - loss: 0.2306 - mae: 0.3600 - val_loss: 0.0489 - val_mae: 0.1455 -
219ms/epoch - 6ms/step
Epoch 4/100
39/39 - 0s - loss: 0.1781 - mae: 0.3145 - val_loss: 0.0384 - val_mae: 0.1217 -
216ms/epoch - 6ms/step
Epoch 5/100
39/39 - 0s - loss: 0.1486 - mae: 0.2865 - val_loss: 0.0362 - val_mae: 0.1226 -
217ms/epoch - 6ms/step
Epoch 6/100
39/39 - 0s - loss: 0.1268 - mae: 0.2644 - val_loss: 0.0324 - val_mae: 0.1174 -
219ms/epoch - 6ms/step
Epoch 7/100
39/39 - 0s - loss: 0.1124 - mae: 0.2476 - val_loss: 0.0261 - val_mae: 0.1038 -
220ms/epoch - 6ms/step
Epoch 8/100
39/39 - 0s - loss: 0.0997 - mae: 0.2322 - val_loss: 0.0291 - val_mae: 0.1225 -
218ms/epoch - 6ms/step
Epoch 9/100
39/39 - 0s - loss: 0.0924 - mae: 0.2240 - val_loss: 0.0205 - val_mae: 0.0900 -
219ms/epoch - 6ms/step
Epoch 10/100
39/39 - 0s - loss: 0.0889 - mae: 0.2178 - val_loss: 0.0266 - val_mae: 0.1182 -
224ms/epoch - 6ms/step
Epoch 11/100
39/39 - 0s - loss: 0.0866 - mae: 0.2173 - val_loss: 0.0279 - val_mae: 0.1219 -
221ms/epoch - 6ms/step
Epoch 12/100
39/39 - 0s - loss: 0.0914 - mae: 0.2269 - val_loss: 0.0294 - val_mae: 0.1273 -
225ms/epoch - 6ms/step
Epoch 13/100
39/39 - 0s - loss: 0.0876 - mae: 0.2241 - val_loss: 0.0288 - val_mae: 0.1226 -
214ms/epoch - 5ms/step
Epoch 14/100
39/39 - 0s - loss: 0.0742 - mae: 0.2039 - val_loss: 0.0209 - val_mae: 0.0941 -
220ms/epoch - 6ms/step
Epoch 15/100
39/39 - 0s - loss: 0.0655 - mae: 0.1886 - val_loss: 0.0211 - val_mae: 0.0988 -
225ms/epoch - 6ms/step
Epoch 16/100
39/39 - 0s - loss: 0.0682 - mae: 0.1934 - val_loss: 0.0257 - val_mae: 0.1122 -
216ms/epoch - 6ms/step
Epoch 17/100
39/39 - 0s - loss: 0.0653 - mae: 0.1894 - val_loss: 0.0218 - val_mae: 0.0931 -
217ms/epoch - 6ms/step
Epoch 18/100
39/39 - 0s - loss: 0.0629 - mae: 0.1848 - val_loss: 0.0227 - val_mae: 0.1009 -
218ms/epoch - 6ms/step
Epoch 19/100
```

```
39/39 - 0s - loss: 0.0566 - mae: 0.1740 - val_loss: 0.0163 - val_mae: 0.0773 -
217ms/epoch - 6ms/step
Epoch 20/100
39/39 - 0s - loss: 0.0522 - mae: 0.1669 - val_loss: 0.0173 - val_mae: 0.0835 -
228ms/epoch - 6ms/step
Epoch 21/100
39/39 - 0s - loss: 0.0552 - mae: 0.1735 - val_loss: 0.0228 - val_mae: 0.1076 -
216ms/epoch - 6ms/step
Epoch 22/100
39/39 - 0s - loss: 0.0520 - mae: 0.1670 - val_loss: 0.0194 - val_mae: 0.0986 -
218ms/epoch - 6ms/step
Epoch 23/100
39/39 - 0s - loss: 0.0506 - mae: 0.1629 - val_loss: 0.0144 - val_mae: 0.0736 -
236ms/epoch - 6ms/step
Epoch 24/100
39/39 - 0s - loss: 0.0622 - mae: 0.1864 - val_loss: 0.0247 - val_mae: 0.1204 -
224ms/epoch - 6ms/step
Epoch 25/100
39/39 - 0s - loss: 0.0628 - mae: 0.1899 - val_loss: 0.0227 - val_mae: 0.1072 -
218ms/epoch - 6ms/step
Epoch 26/100
39/39 - 0s - loss: 0.0642 - mae: 0.1927 - val_loss: 0.0225 - val_mae: 0.1101 -
215ms/epoch - 6ms/step
Epoch 27/100
39/39 - 0s - loss: 0.0596 - mae: 0.1853 - val_loss: 0.0297 - val_mae: 0.1329 -
219ms/epoch - 6ms/step
Epoch 28/100
39/39 - 0s - loss: 0.0547 - mae: 0.1741 - val_loss: 0.0181 - val_mae: 0.0863 -
222ms/epoch - 6ms/step
Epoch 29/100
39/39 - 0s - loss: 0.0510 - mae: 0.1658 - val_loss: 0.0171 - val_mae: 0.0835 -
218ms/epoch - 6ms/step
Epoch 30/100
39/39 - 0s - loss: 0.0491 - mae: 0.1622 - val_loss: 0.0249 - val_mae: 0.1211 -
240ms/epoch - 6ms/step
Epoch 31/100
39/39 - 0s - loss: 0.0462 - mae: 0.1579 - val_loss: 0.0176 - val_mae: 0.0905 -
228ms/epoch - 6ms/step
Epoch 32/100
39/39 - 0s - loss: 0.0436 - mae: 0.1527 - val_loss: 0.0169 - val_mae: 0.0872 -
217ms/epoch - 6ms/step
Epoch 33/100
39/39 - 0s - loss: 0.0495 - mae: 0.1618 - val_loss: 0.0171 - val_mae: 0.0877 -
225ms/epoch - 6ms/step
Epoch 34/100
39/39 - 0s - loss: 0.0504 - mae: 0.1641 - val_loss: 0.0176 - val_mae: 0.0906 -
221ms/epoch - 6ms/step
Epoch 35/100
```

```
39/39 - 0s - loss: 0.0434 - mae: 0.1525 - val_loss: 0.0175 - val_mae: 0.0899 -
215ms/epoch - 6ms/step
Epoch 36/100
39/39 - 0s - loss: 0.0443 - mae: 0.1539 - val_loss: 0.0171 - val_mae: 0.0883 -
224ms/epoch - 6ms/step
Epoch 37/100
39/39 - 0s - loss: 0.0406 - mae: 0.1470 - val_loss: 0.0180 - val_mae: 0.0955 -
222ms/epoch - 6ms/step
Epoch 38/100
39/39 - 0s - loss: 0.0403 - mae: 0.1467 - val_loss: 0.0205 - val_mae: 0.1063 -
216ms/epoch - 6ms/step
Epoch 39/100
39/39 - 0s - loss: 0.0483 - mae: 0.1622 - val_loss: 0.0192 - val_mae: 0.0980 -
226ms/epoch - 6ms/step
Epoch 40/100
39/39 - 0s - loss: 0.0441 - mae: 0.1547 - val_loss: 0.0169 - val_mae: 0.0877 -
222ms/epoch - 6ms/step
Epoch 41/100
39/39 - 0s - loss: 0.0453 - mae: 0.1570 - val_loss: 0.0183 - val_mae: 0.0936 -
223ms/epoch - 6ms/step
Epoch 42/100
39/39 - 0s - loss: 0.0450 - mae: 0.1597 - val_loss: 0.0183 - val_mae: 0.0947 -
369ms/epoch - 9ms/step
Epoch 43/100
39/39 - 0s - loss: 0.0434 - mae: 0.1527 - val_loss: 0.0175 - val_mae: 0.0913 -
396ms/epoch - 10ms/step
Epoch 44/100
39/39 - 0s - loss: 0.0458 - mae: 0.1570 - val_loss: 0.0204 - val_mae: 0.1038 -
215ms/epoch - 6ms/step
Epoch 45/100
39/39 - 0s - loss: 0.0383 - mae: 0.1448 - val_loss: 0.0179 - val_mae: 0.0942 -
215ms/epoch - 6ms/step
Epoch 46/100
39/39 - 0s - loss: 0.0395 - mae: 0.1472 - val_loss: 0.0164 - val_mae: 0.0864 -
218ms/epoch - 6ms/step
Epoch 47/100
39/39 - 0s - loss: 0.0430 - mae: 0.1523 - val_loss: 0.0161 - val_mae: 0.0856 -
224ms/epoch - 6ms/step
Epoch 48/100
39/39 - 0s - loss: 0.0482 - mae: 0.1612 - val_loss: 0.0274 - val_mae: 0.1291 -
221ms/epoch - 6ms/step
Epoch 49/100
39/39 - 0s - loss: 0.0438 - mae: 0.1571 - val_loss: 0.0164 - val_mae: 0.0846 -
219ms/epoch - 6ms/step
Epoch 50/100
39/39 - 0s - loss: 0.0386 - mae: 0.1450 - val_loss: 0.0225 - val_mae: 0.1156 -
218ms/epoch - 6ms/step
Epoch 51/100
```

```
39/39 - 0s - loss: 0.0407 - mae: 0.1479 - val_loss: 0.0161 - val_mae: 0.0801 -
220ms/epoch - 6ms/step
Epoch 52/100
39/39 - 0s - loss: 0.0366 - mae: 0.1399 - val_loss: 0.0174 - val_mae: 0.0874 -
217ms/epoch - 6ms/step
Epoch 53/100
39/39 - 0s - loss: 0.0377 - mae: 0.1422 - val_loss: 0.0172 - val_mae: 0.0880 -
220ms/epoch - 6ms/step
Epoch 54/100
39/39 - 0s - loss: 0.0365 - mae: 0.1395 - val_loss: 0.0174 - val_mae: 0.0935 -
225ms/epoch - 6ms/step
Epoch 55/100
39/39 - 0s - loss: 0.0393 - mae: 0.1464 - val_loss: 0.0144 - val_mae: 0.0754 -
208ms/epoch - 5ms/step
Epoch 56/100
39/39 - 0s - loss: 0.0416 - mae: 0.1494 - val_loss: 0.0181 - val_mae: 0.0935 -
214ms/epoch - 5ms/step
Epoch 57/100
39/39 - 0s - loss: 0.0463 - mae: 0.1592 - val_loss: 0.0208 - val_mae: 0.1019 -
210ms/epoch - 5ms/step
Epoch 58/100
39/39 - 0s - loss: 0.0492 - mae: 0.1661 - val_loss: 0.0222 - val_mae: 0.1104 -
226ms/epoch - 6ms/step
Epoch 59/100
39/39 - 0s - loss: 0.0407 - mae: 0.1518 - val_loss: 0.0166 - val_mae: 0.0848 -
267ms/epoch - 7ms/step
Epoch 60/100
39/39 - 0s - loss: 0.0393 - mae: 0.1425 - val_loss: 0.0220 - val_mae: 0.1121 -
232ms/epoch - 6ms/step
Epoch 61/100
39/39 - 0s - loss: 0.0393 - mae: 0.1454 - val_loss: 0.0153 - val_mae: 0.0792 -
215ms/epoch - 6ms/step
Epoch 62/100
39/39 - 0s - loss: 0.0343 - mae: 0.1354 - val_loss: 0.0176 - val_mae: 0.0919 -
214ms/epoch - 5ms/step
Epoch 63/100
39/39 - 0s - loss: 0.0352 - mae: 0.1367 - val_loss: 0.0154 - val_mae: 0.0762 -
209ms/epoch - 5ms/step
Epoch 64/100
39/39 - 0s - loss: 0.0364 - mae: 0.1385 - val_loss: 0.0140 - val_mae: 0.0734 -
227ms/epoch - 6ms/step
Epoch 65/100
39/39 - 0s - loss: 0.0377 - mae: 0.1410 - val_loss: 0.0202 - val_mae: 0.1026 -
213ms/epoch - 5ms/step
Epoch 66/100
39/39 - 0s - loss: 0.0381 - mae: 0.1462 - val_loss: 0.0230 - val_mae: 0.1176 -
207ms/epoch - 5ms/step
Epoch 67/100
```

```
39/39 - 0s - loss: 0.0364 - mae: 0.1398 - val_loss: 0.0146 - val_mae: 0.0709 -
209ms/epoch - 5ms/step
Epoch 68/100
39/39 - 0s - loss: 0.0391 - mae: 0.1415 - val_loss: 0.0159 - val_mae: 0.0846 -
212ms/epoch - 5ms/step
Epoch 69/100
39/39 - 0s - loss: 0.0379 - mae: 0.1423 - val_loss: 0.0193 - val_mae: 0.1003 -
210ms/epoch - 5ms/step
Epoch 70/100
39/39 - 0s - loss: 0.0411 - mae: 0.1520 - val_loss: 0.0184 - val_mae: 0.0956 -
216ms/epoch - 6ms/step
Epoch 71/100
39/39 - 0s - loss: 0.0380 - mae: 0.1449 - val_loss: 0.0189 - val_mae: 0.1001 -
209ms/epoch - 5ms/step
Epoch 72/100
39/39 - 0s - loss: 0.0427 - mae: 0.1514 - val_loss: 0.0183 - val_mae: 0.0936 -
216ms/epoch - 6ms/step
Epoch 73/100
39/39 - 0s - loss: 0.0412 - mae: 0.1482 - val_loss: 0.0217 - val_mae: 0.1115 -
213ms/epoch - 5ms/step
Epoch 74/100
39/39 - 0s - loss: 0.0381 - mae: 0.1416 - val_loss: 0.0187 - val_mae: 0.0982 -
210ms/epoch - 5ms/step
Epoch 75/100
39/39 - 0s - loss: 0.0370 - mae: 0.1428 - val_loss: 0.0213 - val_mae: 0.1099 -
214ms/epoch - 5ms/step
Epoch 76/100
39/39 - 0s - loss: 0.0405 - mae: 0.1488 - val_loss: 0.0226 - val_mae: 0.1163 -
208ms/epoch - 5ms/step
Epoch 77/100
39/39 - 0s - loss: 0.0392 - mae: 0.1458 - val_loss: 0.0212 - val_mae: 0.1103 -
212ms/epoch - 5ms/step
Epoch 78/100
39/39 - 0s - loss: 0.0404 - mae: 0.1465 - val_loss: 0.0177 - val_mae: 0.0944 -
214ms/epoch - 5ms/step
Epoch 79/100
39/39 - 0s - loss: 0.0402 - mae: 0.1482 - val_loss: 0.0198 - val_mae: 0.1018 -
208ms/epoch - 5ms/step
Epoch 80/100
39/39 - 0s - loss: 0.0413 - mae: 0.1476 - val_loss: 0.0223 - val_mae: 0.1140 -
208ms/epoch - 5ms/step
Epoch 81/100
39/39 - 0s - loss: 0.0391 - mae: 0.1459 - val_loss: 0.0225 - val_mae: 0.1160 -
219ms/epoch - 6ms/step
Epoch 82/100
39/39 - 0s - loss: 0.0418 - mae: 0.1506 - val_loss: 0.0192 - val_mae: 0.1016 -
214ms/epoch - 5ms/step
Epoch 83/100
```

```
39/39 - 0s - loss: 0.0365 - mae: 0.1418 - val_loss: 0.0181 - val_mae: 0.0961 -
209ms/epoch - 5ms/step
Epoch 84/100
39/39 - 0s - loss: 0.0414 - mae: 0.1485 - val_loss: 0.0168 - val_mae: 0.0891 -
205ms/epoch - 5ms/step
Epoch 85/100
39/39 - 0s - loss: 0.0372 - mae: 0.1420 - val_loss: 0.0207 - val_mae: 0.1081 -
212ms/epoch - 5ms/step
Epoch 86/100
39/39 - 0s - loss: 0.0373 - mae: 0.1418 - val_loss: 0.0151 - val_mae: 0.0791 -
211ms/epoch - 5ms/step
Epoch 87/100
39/39 - 0s - loss: 0.0431 - mae: 0.1538 - val_loss: 0.0195 - val_mae: 0.0999 -
229ms/epoch - 6ms/step
Epoch 88/100
39/39 - 0s - loss: 0.0411 - mae: 0.1518 - val_loss: 0.0219 - val_mae: 0.1124 -
207ms/epoch - 5ms/step
Epoch 89/100
39/39 - 0s - loss: 0.0389 - mae: 0.1469 - val_loss: 0.0214 - val_mae: 0.1098 -
207ms/epoch - 5ms/step
Epoch 90/100
39/39 - 0s - loss: 0.0349 - mae: 0.1374 - val_loss: 0.0255 - val_mae: 0.1277 -
213ms/epoch - 5ms/step
Epoch 91/100
39/39 - 0s - loss: 0.0350 - mae: 0.1382 - val_loss: 0.0225 - val_mae: 0.1144 -
209ms/epoch - 5ms/step
Epoch 92/100
39/39 - 0s - loss: 0.0370 - mae: 0.1404 - val_loss: 0.0258 - val_mae: 0.1291 -
204ms/epoch - 5ms/step
Epoch 93/100
39/39 - 0s - loss: 0.0365 - mae: 0.1386 - val_loss: 0.0246 - val_mae: 0.1231 -
211ms/epoch - 5ms/step
Epoch 94/100
39/39 - 0s - loss: 0.0374 - mae: 0.1412 - val_loss: 0.0213 - val_mae: 0.1107 -
210ms/epoch - 5ms/step
Epoch 95/100
39/39 - 0s - loss: 0.0359 - mae: 0.1405 - val_loss: 0.0205 - val_mae: 0.1023 -
206ms/epoch - 5ms/step
Epoch 96/100
39/39 - 0s - loss: 0.0359 - mae: 0.1371 - val_loss: 0.0141 - val_mae: 0.0745 -
209ms/epoch - 5ms/step
Epoch 97/100
39/39 - 0s - loss: 0.0373 - mae: 0.1391 - val_loss: 0.0205 - val_mae: 0.1089 -
208ms/epoch - 5ms/step
Epoch 98/100
39/39 - 0s - loss: 0.0363 - mae: 0.1395 - val_loss: 0.0182 - val_mae: 0.0974 -
207ms/epoch - 5ms/step
Epoch 99/100
```

```
39/39 - 0s - loss: 0.0348 - mae: 0.1384 - val_loss: 0.0150 - val_mae: 0.0805 -
212ms/epoch - 5ms/step
Epoch 100/100
39/39 - 0s - loss: 0.0378 - mae: 0.1447 - val_loss: 0.0199 - val_mae: 0.1019 -
212ms/epoch - 5ms/step
```

[127]:
```
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
39/39 - 5s - loss: 0.8346 - mae: 0.7923 - val_loss: 0.4397 - val_mae: 0.5781 -
5s/epoch - 125ms/step
Epoch 2/100
39/39 - 0s - loss: 0.2562 - mae: 0.3796 - val_loss: 0.0979 - val_mae: 0.2256 -
379ms/epoch - 10ms/step
Epoch 3/100
39/39 - 0s - loss: 0.1387 - mae: 0.2817 - val_loss: 0.0571 - val_mae: 0.1537 -
379ms/epoch - 10ms/step
Epoch 4/100
39/39 - 0s - loss: 0.1133 - mae: 0.2517 - val_loss: 0.0543 - val_mae: 0.1473 -
382ms/epoch - 10ms/step
Epoch 5/100
39/39 - 0s - loss: 0.0959 - mae: 0.2313 - val_loss: 0.0509 - val_mae: 0.1403 -
384ms/epoch - 10ms/step
Epoch 6/100
39/39 - 0s - loss: 0.0812 - mae: 0.2105 - val_loss: 0.0535 - val_mae: 0.1488 -
382ms/epoch - 10ms/step
Epoch 7/100
39/39 - 0s - loss: 0.0794 - mae: 0.2088 - val_loss: 0.0575 - val_mae: 0.1547 -
379ms/epoch - 10ms/step
Epoch 8/100
39/39 - 0s - loss: 0.0739 - mae: 0.2007 - val_loss: 0.0687 - val_mae: 0.1770 -
377ms/epoch - 10ms/step
Epoch 9/100
39/39 - 0s - loss: 0.0716 - mae: 0.1979 - val_loss: 0.0736 - val_mae: 0.1852 -
377ms/epoch - 10ms/step
Epoch 10/100
39/39 - 0s - loss: 0.0735 - mae: 0.2007 - val_loss: 0.1193 - val_mae: 0.2592 -
371ms/epoch - 10ms/step
Epoch 11/100
39/39 - 0s - loss: 0.0767 - mae: 0.2064 - val_loss: 0.1395 - val_mae: 0.2841 -
379ms/epoch - 10ms/step
Epoch 12/100
39/39 - 0s - loss: 0.0813 - mae: 0.2148 - val_loss: 0.1186 - val_mae: 0.2553 -
383ms/epoch - 10ms/step
Epoch 13/100
39/39 - 0s - loss: 0.0770 - mae: 0.2091 - val_loss: 0.0957 - val_mae: 0.2278 -
```

```
379ms/epoch - 10ms/step
Epoch 14/100
39/39 - 0s - loss: 0.0701 - mae: 0.1981 - val_loss: 0.0724 - val_mae: 0.1911 -
375ms/epoch - 10ms/step
Epoch 15/100
39/39 - 0s - loss: 0.0680 - mae: 0.1939 - val_loss: 0.0738 - val_mae: 0.1918 -
380ms/epoch - 10ms/step
Epoch 16/100
39/39 - 0s - loss: 0.0615 - mae: 0.1837 - val_loss: 0.0504 - val_mae: 0.1457 -
382ms/epoch - 10ms/step
Epoch 17/100
39/39 - 0s - loss: 0.0565 - mae: 0.1742 - val_loss: 0.0542 - val_mae: 0.1585 -
381ms/epoch - 10ms/step
Epoch 18/100
39/39 - 0s - loss: 0.0536 - mae: 0.1711 - val_loss: 0.0506 - val_mae: 0.1480 -
380ms/epoch - 10ms/step
Epoch 19/100
39/39 - 0s - loss: 0.0541 - mae: 0.1720 - val_loss: 0.0713 - val_mae: 0.1929 -
379ms/epoch - 10ms/step
Epoch 20/100
39/39 - 0s - loss: 0.0580 - mae: 0.1788 - val_loss: 0.0841 - val_mae: 0.2156 -
382ms/epoch - 10ms/step
Epoch 21/100
39/39 - 0s - loss: 0.0605 - mae: 0.1844 - val_loss: 0.1131 - val_mae: 0.2517 -
382ms/epoch - 10ms/step
Epoch 22/100
39/39 - 0s - loss: 0.0709 - mae: 0.1990 - val_loss: 0.0997 - val_mae: 0.2293 -
378ms/epoch - 10ms/step
Epoch 23/100
39/39 - 0s - loss: 0.0664 - mae: 0.1911 - val_loss: 0.0720 - val_mae: 0.1893 -
378ms/epoch - 10ms/step
Epoch 24/100
39/39 - 0s - loss: 0.0550 - mae: 0.1747 - val_loss: 0.0635 - val_mae: 0.1782 -
377ms/epoch - 10ms/step
Epoch 25/100
39/39 - 0s - loss: 0.0561 - mae: 0.1746 - val_loss: 0.0651 - val_mae: 0.1828 -
380ms/epoch - 10ms/step
Epoch 26/100
39/39 - 1s - loss: 0.0551 - mae: 0.1731 - val_loss: 0.0726 - val_mae: 0.1952 -
688ms/epoch - 18ms/step
Epoch 27/100
39/39 - 0s - loss: 0.0563 - mae: 0.1760 - val_loss: 0.0696 - val_mae: 0.1905 -
392ms/epoch - 10ms/step
Epoch 28/100
39/39 - 0s - loss: 0.0557 - mae: 0.1755 - val_loss: 0.0645 - val_mae: 0.1817 -
373ms/epoch - 10ms/step
Epoch 29/100
39/39 - 0s - loss: 0.0512 - mae: 0.1683 - val_loss: 0.0588 - val_mae: 0.1744 -
```

```
375ms/epoch - 10ms/step
Epoch 30/100
39/39 - 0s - loss: 0.0496 - mae: 0.1634 - val_loss: 0.0467 - val_mae: 0.1433 -
378ms/epoch - 10ms/step
Epoch 31/100
39/39 - 0s - loss: 0.0459 - mae: 0.1592 - val_loss: 0.0450 - val_mae: 0.1362 -
376ms/epoch - 10ms/step
Epoch 32/100
39/39 - 0s - loss: 0.0462 - mae: 0.1584 - val_loss: 0.0537 - val_mae: 0.1627 -
377ms/epoch - 10ms/step
Epoch 33/100
39/39 - 0s - loss: 0.0480 - mae: 0.1605 - val_loss: 0.0597 - val_mae: 0.1784 -
377ms/epoch - 10ms/step
Epoch 34/100
39/39 - 0s - loss: 0.0511 - mae: 0.1660 - val_loss: 0.0568 - val_mae: 0.1634 -
373ms/epoch - 10ms/step
Epoch 35/100
39/39 - 0s - loss: 0.0462 - mae: 0.1611 - val_loss: 0.0587 - val_mae: 0.1739 -
377ms/epoch - 10ms/step
Epoch 36/100
39/39 - 0s - loss: 0.0488 - mae: 0.1620 - val_loss: 0.0548 - val_mae: 0.1631 -
377ms/epoch - 10ms/step
Epoch 37/100
39/39 - 0s - loss: 0.0453 - mae: 0.1560 - val_loss: 0.0589 - val_mae: 0.1690 -
377ms/epoch - 10ms/step
Epoch 38/100
39/39 - 0s - loss: 0.0476 - mae: 0.1591 - val_loss: 0.0533 - val_mae: 0.1577 -
378ms/epoch - 10ms/step
Epoch 39/100
39/39 - 0s - loss: 0.0472 - mae: 0.1587 - val_loss: 0.0621 - val_mae: 0.1712 -
375ms/epoch - 10ms/step
Epoch 40/100
39/39 - 0s - loss: 0.0486 - mae: 0.1609 - val_loss: 0.0516 - val_mae: 0.1536 -
374ms/epoch - 10ms/step
Epoch 41/100
39/39 - 0s - loss: 0.0478 - mae: 0.1594 - val_loss: 0.0547 - val_mae: 0.1673 -
379ms/epoch - 10ms/step
Epoch 42/100
39/39 - 0s - loss: 0.0459 - mae: 0.1576 - val_loss: 0.0553 - val_mae: 0.1675 -
380ms/epoch - 10ms/step
Epoch 43/100
39/39 - 0s - loss: 0.0454 - mae: 0.1573 - val_loss: 0.0464 - val_mae: 0.1448 -
374ms/epoch - 10ms/step
Epoch 44/100
39/39 - 0s - loss: 0.0419 - mae: 0.1500 - val_loss: 0.0472 - val_mae: 0.1430 -
379ms/epoch - 10ms/step
Epoch 45/100
39/39 - 0s - loss: 0.0403 - mae: 0.1476 - val_loss: 0.0664 - val_mae: 0.1862 -
```

```
369ms/epoch - 9ms/step
Epoch 46/100
39/39 - 0s - loss: 0.0477 - mae: 0.1627 - val_loss: 0.0633 - val_mae: 0.1807 -
365ms/epoch - 9ms/step
Epoch 47/100
39/39 - 0s - loss: 0.0490 - mae: 0.1640 - val_loss: 0.0937 - val_mae: 0.2302 -
363ms/epoch - 9ms/step
Epoch 48/100
39/39 - 0s - loss: 0.0516 - mae: 0.1721 - val_loss: 0.0645 - val_mae: 0.1867 -
373ms/epoch - 10ms/step
Epoch 49/100
39/39 - 0s - loss: 0.0500 - mae: 0.1634 - val_loss: 0.0685 - val_mae: 0.1917 -
369ms/epoch - 9ms/step
Epoch 50/100
39/39 - 0s - loss: 0.0478 - mae: 0.1630 - val_loss: 0.0684 - val_mae: 0.1983 -
369ms/epoch - 9ms/step
Epoch 51/100
39/39 - 0s - loss: 0.0495 - mae: 0.1663 - val_loss: 0.0407 - val_mae: 0.1318 -
368ms/epoch - 9ms/step
Epoch 52/100
39/39 - 0s - loss: 0.0407 - mae: 0.1476 - val_loss: 0.0296 - val_mae: 0.1049 -
365ms/epoch - 9ms/step
Epoch 53/100
39/39 - 0s - loss: 0.0382 - mae: 0.1417 - val_loss: 0.0334 - val_mae: 0.1168 -
366ms/epoch - 9ms/step
Epoch 54/100
39/39 - 0s - loss: 0.0390 - mae: 0.1446 - val_loss: 0.0349 - val_mae: 0.1197 -
367ms/epoch - 9ms/step
Epoch 55/100
39/39 - 0s - loss: 0.0380 - mae: 0.1433 - val_loss: 0.0364 - val_mae: 0.1231 -
359ms/epoch - 9ms/step
Epoch 56/100
39/39 - 0s - loss: 0.0380 - mae: 0.1412 - val_loss: 0.0405 - val_mae: 0.1331 -
381ms/epoch - 10ms/step
Epoch 57/100
39/39 - 0s - loss: 0.0385 - mae: 0.1449 - val_loss: 0.0475 - val_mae: 0.1588 -
428ms/epoch - 11ms/step
Epoch 58/100
39/39 - 0s - loss: 0.0417 - mae: 0.1498 - val_loss: 0.0457 - val_mae: 0.1518 -
360ms/epoch - 9ms/step
Epoch 59/100
39/39 - 0s - loss: 0.0407 - mae: 0.1487 - val_loss: 0.0403 - val_mae: 0.1339 -
359ms/epoch - 9ms/step
Epoch 60/100
39/39 - 0s - loss: 0.0402 - mae: 0.1483 - val_loss: 0.0506 - val_mae: 0.1622 -
353ms/epoch - 9ms/step
Epoch 61/100
39/39 - 0s - loss: 0.0444 - mae: 0.1542 - val_loss: 0.0523 - val_mae: 0.1636 -
```

```
361ms/epoch - 9ms/step
Epoch 62/100
39/39 - 0s - loss: 0.0398 - mae: 0.1478 - val_loss: 0.0548 - val_mae: 0.1694 -
364ms/epoch - 9ms/step
Epoch 63/100
39/39 - 0s - loss: 0.0432 - mae: 0.1522 - val_loss: 0.0538 - val_mae: 0.1640 -
367ms/epoch - 9ms/step
Epoch 64/100
39/39 - 0s - loss: 0.0432 - mae: 0.1541 - val_loss: 0.0701 - val_mae: 0.1967 -
365ms/epoch - 9ms/step
Epoch 65/100
39/39 - 0s - loss: 0.0480 - mae: 0.1633 - val_loss: 0.0721 - val_mae: 0.1950 -
372ms/epoch - 10ms/step
Epoch 66/100
39/39 - 0s - loss: 0.0499 - mae: 0.1683 - val_loss: 0.0790 - val_mae: 0.2176 -
368ms/epoch - 9ms/step
Epoch 67/100
39/39 - 0s - loss: 0.0468 - mae: 0.1596 - val_loss: 0.0570 - val_mae: 0.1726 -
379ms/epoch - 10ms/step
Epoch 68/100
39/39 - 0s - loss: 0.0442 - mae: 0.1554 - val_loss: 0.0474 - val_mae: 0.1555 -
363ms/epoch - 9ms/step
Epoch 69/100
39/39 - 0s - loss: 0.0405 - mae: 0.1479 - val_loss: 0.0421 - val_mae: 0.1444 -
373ms/epoch - 10ms/step
Epoch 70/100
39/39 - 0s - loss: 0.0401 - mae: 0.1472 - val_loss: 0.0419 - val_mae: 0.1443 -
367ms/epoch - 9ms/step
Epoch 71/100
39/39 - 0s - loss: 0.0375 - mae: 0.1442 - val_loss: 0.0425 - val_mae: 0.1436 -
370ms/epoch - 9ms/step
Epoch 72/100
39/39 - 0s - loss: 0.0417 - mae: 0.1493 - val_loss: 0.0529 - val_mae: 0.1616 -
366ms/epoch - 9ms/step
Epoch 73/100
39/39 - 0s - loss: 0.0443 - mae: 0.1570 - val_loss: 0.0547 - val_mae: 0.1728 -
366ms/epoch - 9ms/step
Epoch 74/100
39/39 - 0s - loss: 0.0439 - mae: 0.1568 - val_loss: 0.0574 - val_mae: 0.1763 -
355ms/epoch - 9ms/step
Epoch 75/100
39/39 - 0s - loss: 0.0443 - mae: 0.1586 - val_loss: 0.0640 - val_mae: 0.1874 -
355ms/epoch - 9ms/step
Epoch 76/100
39/39 - 0s - loss: 0.0446 - mae: 0.1567 - val_loss: 0.0487 - val_mae: 0.1642 -
356ms/epoch - 9ms/step
Epoch 77/100
39/39 - 0s - loss: 0.0425 - mae: 0.1527 - val_loss: 0.0368 - val_mae: 0.1359 -
```

```
357ms/epoch - 9ms/step
Epoch 78/100
39/39 - 0s - loss: 0.0406 - mae: 0.1482 - val_loss: 0.0341 - val_mae: 0.1249 -
362ms/epoch - 9ms/step
Epoch 79/100
39/39 - 0s - loss: 0.0379 - mae: 0.1421 - val_loss: 0.0355 - val_mae: 0.1270 -
458ms/epoch - 12ms/step
Epoch 80/100
39/39 - 0s - loss: 0.0383 - mae: 0.1407 - val_loss: 0.0322 - val_mae: 0.1205 -
424ms/epoch - 11ms/step
Epoch 81/100
39/39 - 0s - loss: 0.0375 - mae: 0.1400 - val_loss: 0.0303 - val_mae: 0.1161 -
374ms/epoch - 10ms/step
Epoch 82/100
39/39 - 0s - loss: 0.0354 - mae: 0.1384 - val_loss: 0.0231 - val_mae: 0.0895 -
390ms/epoch - 10ms/step
Epoch 83/100
39/39 - 0s - loss: 0.0348 - mae: 0.1356 - val_loss: 0.0267 - val_mae: 0.1071 -
493ms/epoch - 13ms/step
Epoch 84/100
39/39 - 0s - loss: 0.0362 - mae: 0.1378 - val_loss: 0.0239 - val_mae: 0.0934 -
381ms/epoch - 10ms/step
Epoch 85/100
39/39 - 1s - loss: 0.0333 - mae: 0.1338 - val_loss: 0.0256 - val_mae: 0.0981 -
504ms/epoch - 13ms/step
Epoch 86/100
39/39 - 1s - loss: 0.0336 - mae: 0.1336 - val_loss: 0.0336 - val_mae: 0.1277 -
519ms/epoch - 13ms/step
Epoch 87/100
39/39 - 0s - loss: 0.0380 - mae: 0.1440 - val_loss: 0.0428 - val_mae: 0.1475 -
402ms/epoch - 10ms/step
Epoch 88/100
39/39 - 0s - loss: 0.0396 - mae: 0.1476 - val_loss: 0.0506 - val_mae: 0.1692 -
476ms/epoch - 12ms/step
Epoch 89/100
39/39 - 0s - loss: 0.0400 - mae: 0.1489 - val_loss: 0.0512 - val_mae: 0.1654 -
427ms/epoch - 11ms/step
Epoch 90/100
39/39 - 0s - loss: 0.0428 - mae: 0.1516 - val_loss: 0.0712 - val_mae: 0.2132 -
441ms/epoch - 11ms/step
Epoch 91/100
39/39 - 0s - loss: 0.0479 - mae: 0.1631 - val_loss: 0.0649 - val_mae: 0.1897 -
474ms/epoch - 12ms/step
Epoch 92/100
39/39 - 0s - loss: 0.0469 - mae: 0.1623 - val_loss: 0.0362 - val_mae: 0.1333 -
466ms/epoch - 12ms/step
Epoch 93/100
39/39 - 0s - loss: 0.0370 - mae: 0.1418 - val_loss: 0.0294 - val_mae: 0.1103 -
```

```
382ms/epoch - 10ms/step
Epoch 94/100
39/39 - 0s - loss: 0.0367 - mae: 0.1388 - val_loss: 0.0424 - val_mae: 0.1489 -
469ms/epoch - 12ms/step
Epoch 95/100
39/39 - 0s - loss: 0.0390 - mae: 0.1430 - val_loss: 0.0347 - val_mae: 0.1264 -
358ms/epoch - 9ms/step
Epoch 96/100
39/39 - 0s - loss: 0.0402 - mae: 0.1469 - val_loss: 0.0367 - val_mae: 0.1350 -
361ms/epoch - 9ms/step
Epoch 97/100
39/39 - 0s - loss: 0.0365 - mae: 0.1420 - val_loss: 0.0522 - val_mae: 0.1821 -
493ms/epoch - 13ms/step
Epoch 98/100
39/39 - 0s - loss: 0.0403 - mae: 0.1488 - val_loss: 0.0417 - val_mae: 0.1445 -
473ms/epoch - 12ms/step
Epoch 99/100
39/39 - 0s - loss: 0.0397 - mae: 0.1458 - val_loss: 0.0301 - val_mae: 0.1184 -
496ms/epoch - 13ms/step
Epoch 100/100
39/39 - 1s - loss: 0.0370 - mae: 0.1400 - val_loss: 0.0304 - val_mae: 0.1220 -
525ms/epoch - 13ms/step
```

1.13 Make Predictions

[128]:
```
RNN_Predictions = RNN_Model.predict(X_test)
LSTM_predictions = LSTM_Model.predict(X_test)
```

1.14 Inverse Transform the Values

[129]:
```
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)

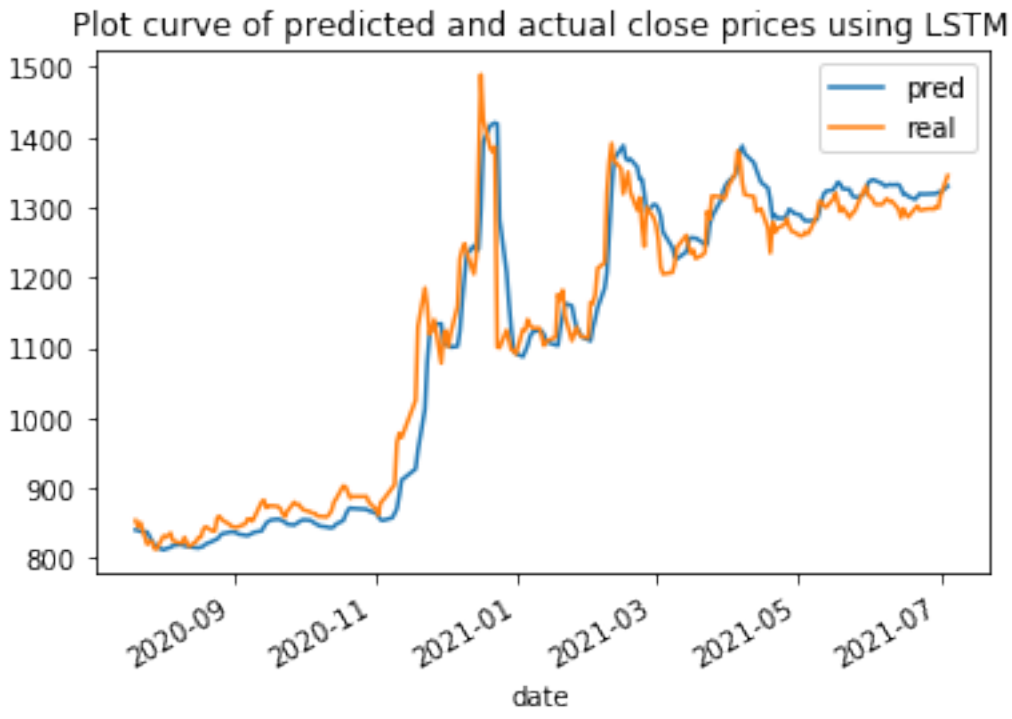
[130]:
```
print("### RNN Model ###")
Y_test_res_RNN = std_scalar.inverse_transform(y_test)
pre_RNN = RNN_act_prd[:,:1]

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 38.98487522765638
MAE 24.33409862029247
```

[131]:
```
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)}")
```

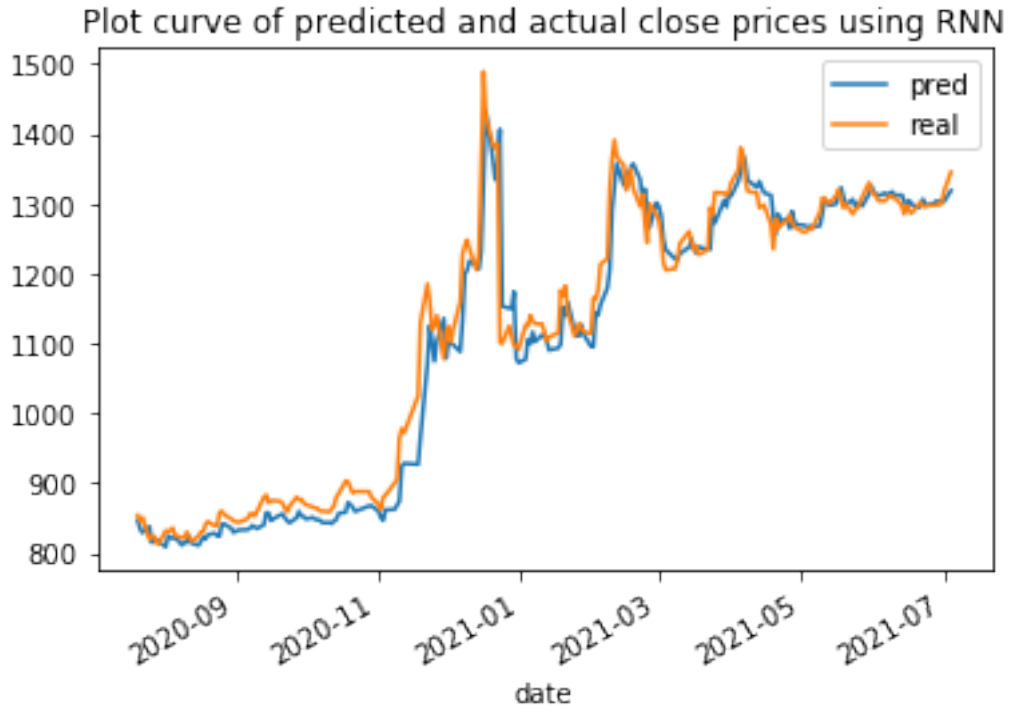
```
### LSTM Model ###
RMSE 46.422508767438416
MAE 30.296848655765892
```

[136]:
```
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()
```

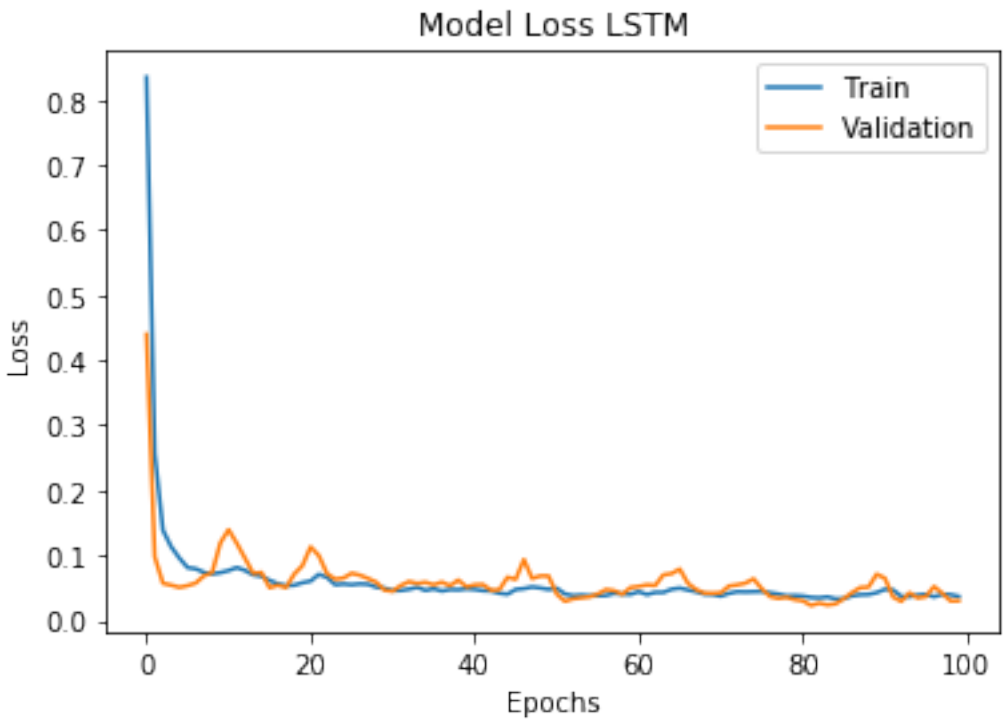


[137]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
```

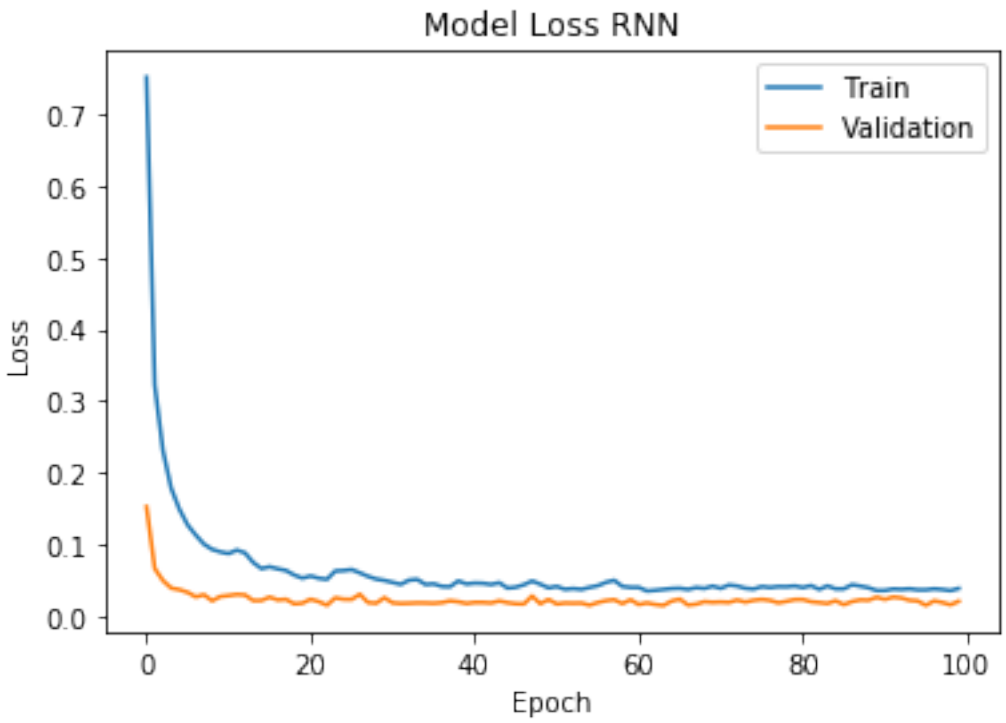
```
plot["date"]=X_test_split.index[:-steps]
plot.plot(kind="line", x="date", title="Plot curve of predicted and actual␣
 ↪close prices using RNN")
plt.show()
```



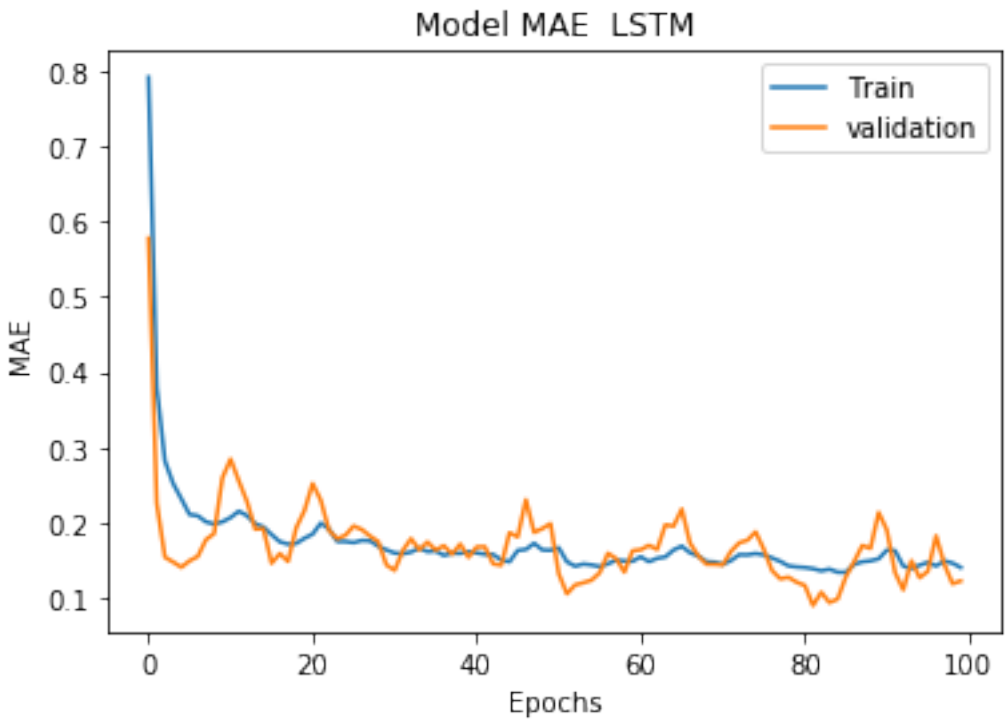
[147]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



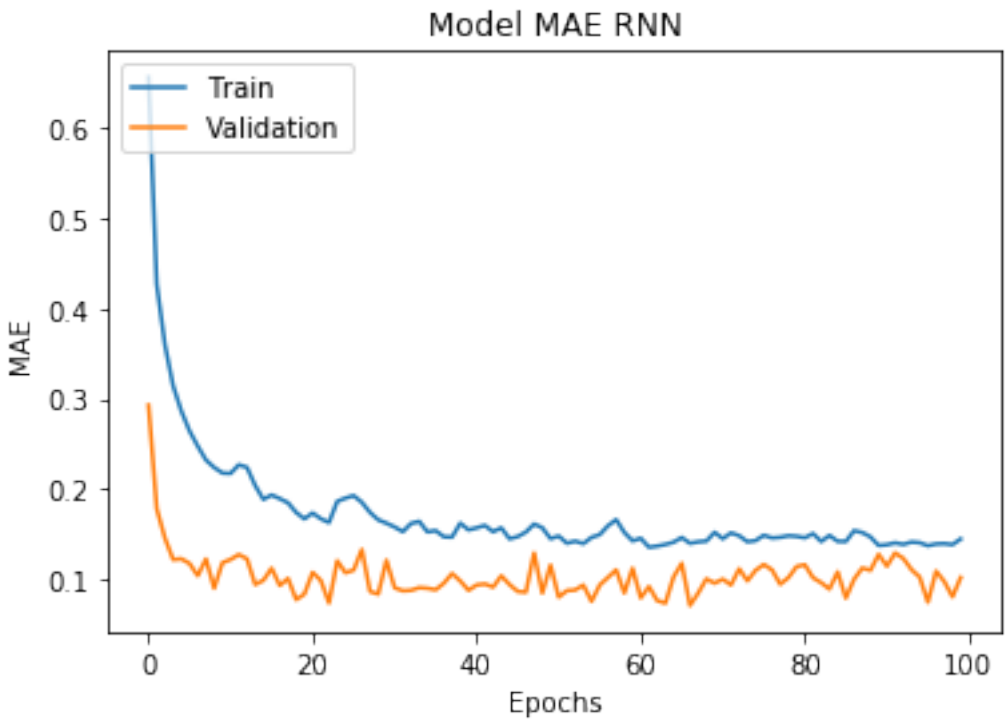
[148]:
```
plt.plot(RNN_History.history['loss'])
plt.plot(RNN_History.history['val_loss'])
plt.title('Model Loss RNN')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```



[149]:
```
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()
```



[150]:
```
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 NABIL Bank LSTM and RNN Models used for Stock Price Prediction
2. The Error is Low for RNN Model

OverAllConclusion

July 30, 2022

1 Overall Conclusion

We can see the values of RMSE and MAE for 5 diffrent companies using RNN and LSTM Model from table below. From the table we can see that RNN performed well for ADBL,NABIL and UPPER and LSTM performed well for LBLL and NLIC.

S.N	**Company Name**	**Model**	**RMSE**	**MAE**
1	ADBL	RNN	9.71	6.33
2	ADBL	LSTM	11.51	7.43
3	LBLL	RNN	6.46	3.95
4	LBLL	LSTM	5.65	3.95
5	NABIL	RNN	38.98	24.33
6	NABIL	LSTM	46.422	30.29
7	NLIC	RNN	76.88	46.06
8	NLIC	LSTM	63.47	40.05
9	UPPER	RNN	28.72	21.95
10	UPPER	LSTM	49.72	39.95