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
```

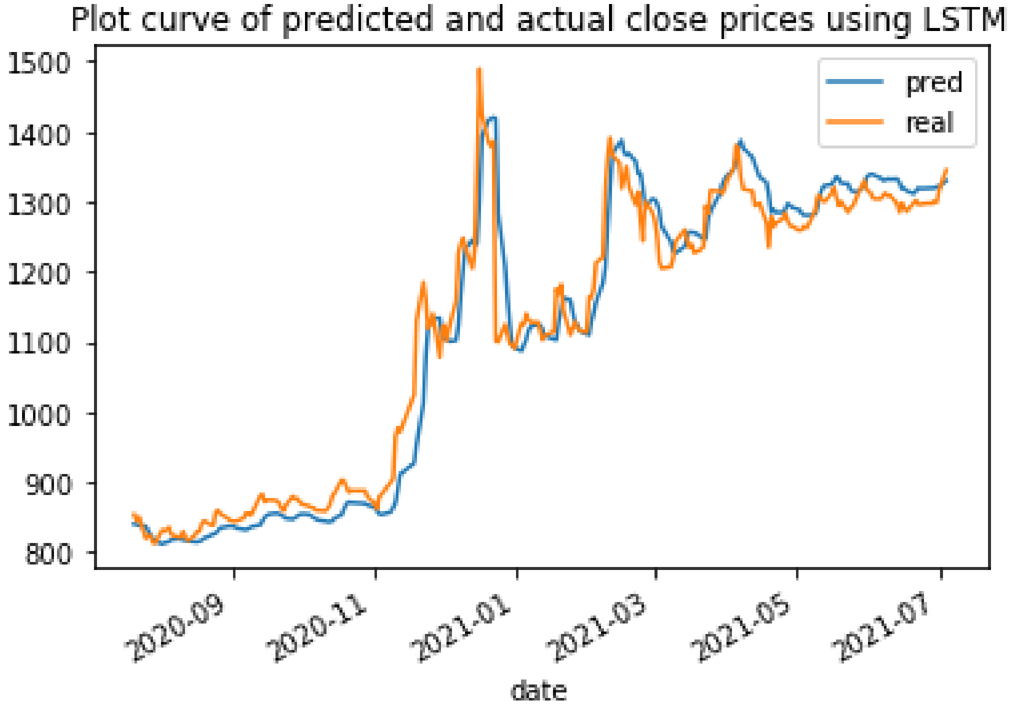
[131]:
```
print("### LSTM Model ###")
Y_test_res_LSTM = std_scalar.inverse_transform(y_test)
pre_LSTM = LSTM_act_prd[:,:1]

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)}")
```

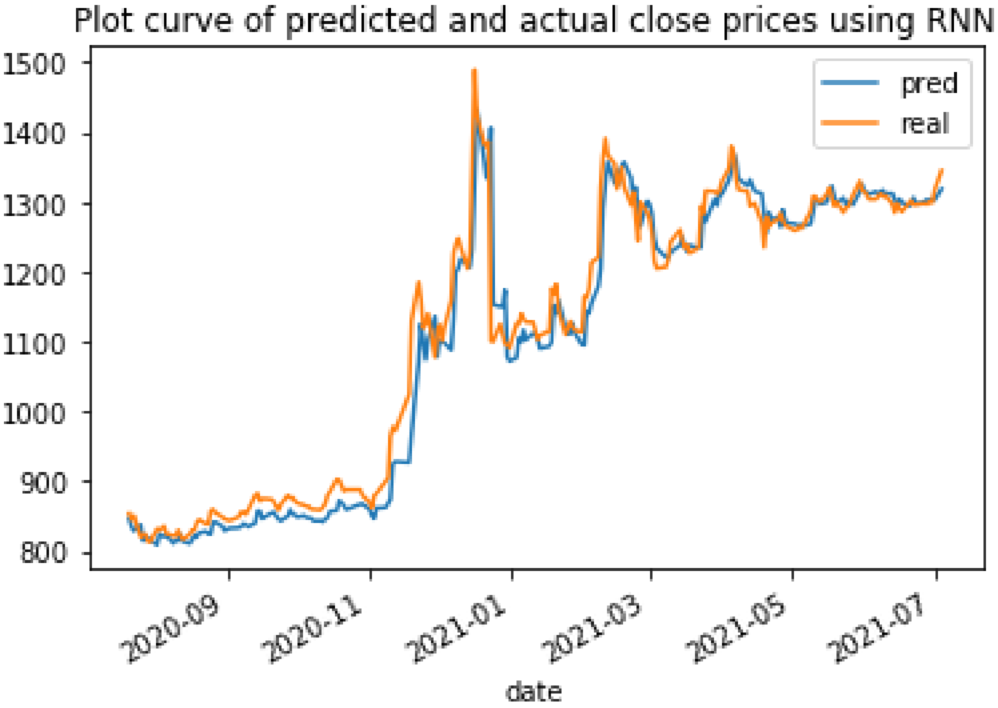
```
### LSTM Model ###
RMSE 46.422508767438416
MAE 30.296848655765892
```

[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()
```

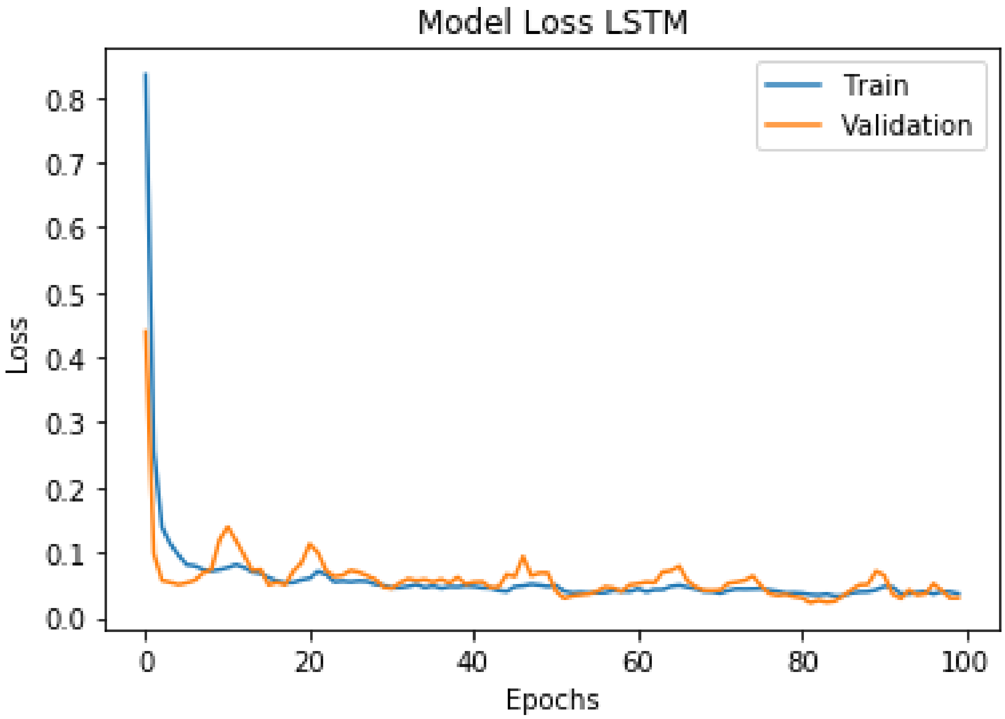


[137]:
```
plot =pd.DataFrame()
plot["pred"]=list(map(float, pre_RNN))
plot["real"]=list(map(float, Y_test_res_RNN))
```

```
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()
```

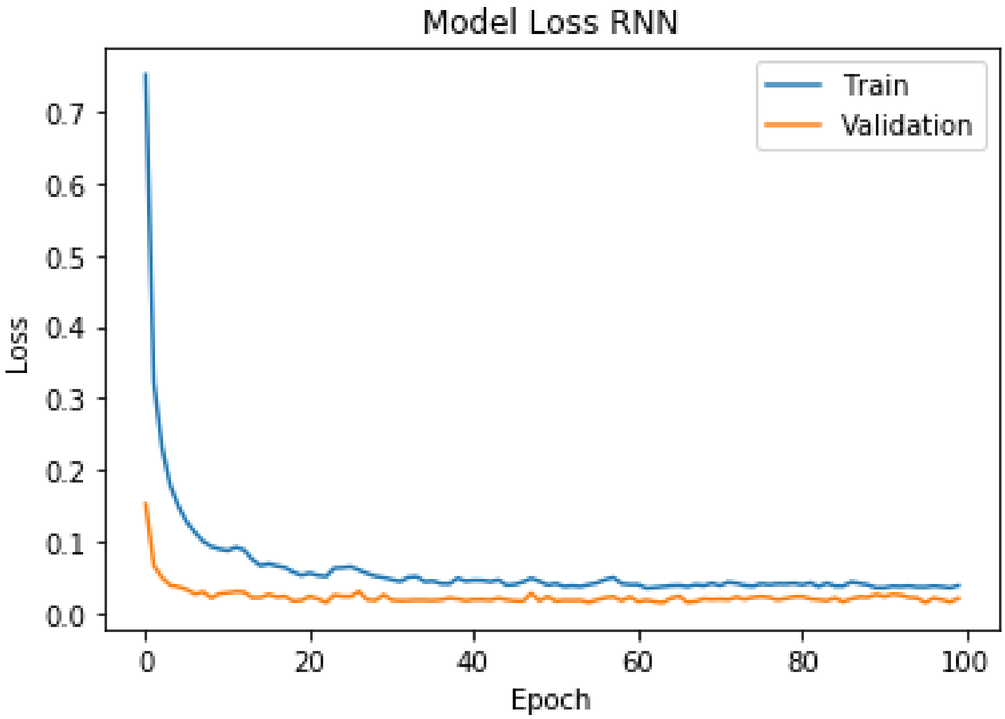


[147]:
```
plt.plot(LSTM_History.history['loss'])
plt.plot(LSTM_History.history['val_loss'])
plt.title('Model Loss LSTM')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```

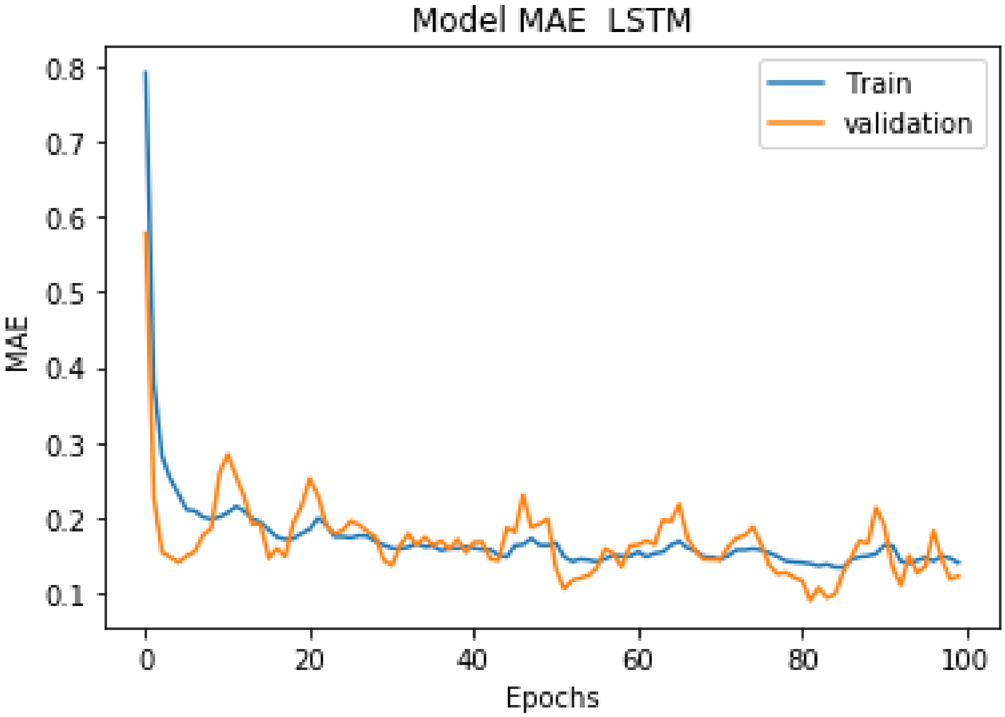


[148]:

```
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()
```

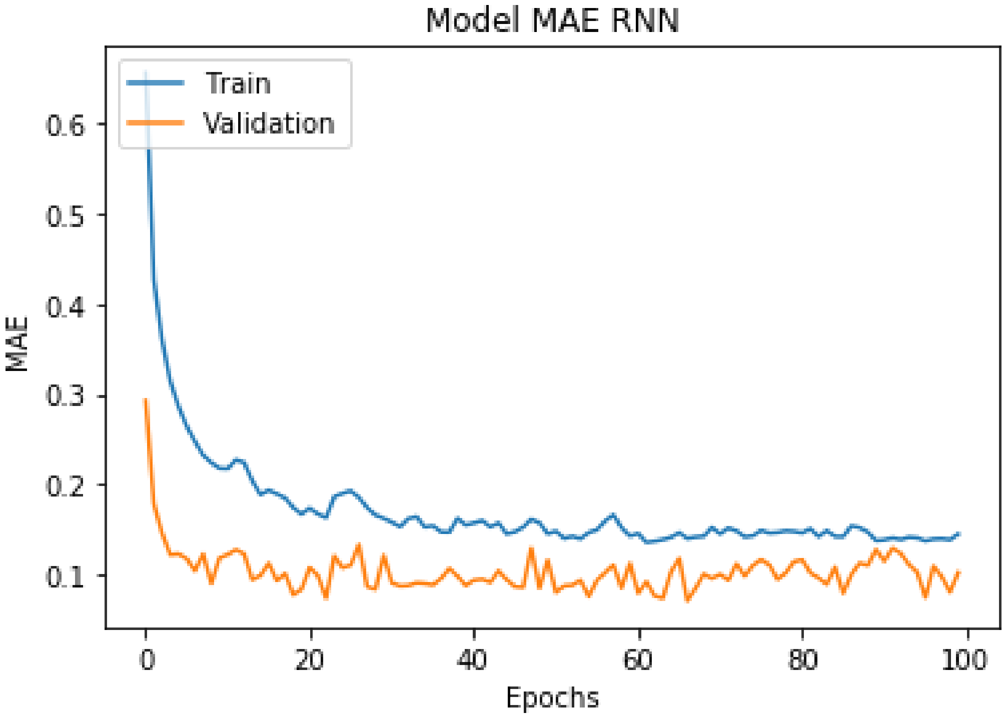


[149]:
```
plt.plot(LSTM_History.history['mae'])
plt.plot(LSTM_History.history['val_mae'])
plt.title('Model MAE  LSTM')
plt.ylabel('MAE')
plt.xlabel('Epochs')
plt.legend(['Train', 'validation'], loc='upper right')
plt.show()
```



[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