Task 2 :- Stock Market Prediction And Forecasting Using Stacked LSTM

Author : - Mitali D Shinde

Level:-Beginner

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
```

In [2]:

```
data = pd.read_csv("stock.csv")
#import datset
```

In [3]:

```
data.head()
#show first 5 rows
```

Out[3]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	28-09-2018	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	27-09-2018	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	26-09-2018	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	25-09-2018	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	24-09-2018	233.55	239.20	230.75	234.00	233.30	3423509	7999.55

In [4]:

```
data.tail()
#show last 5 rows
```

Out[4]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2030	27-07-2010	117.6	119.50	112.00	118.80	118.65	586100	694.98
2031	26-07-2010	120.1	121.00	117.10	117.10	117.60	658440	780.01
2032	23-07-2010	121.8	121.95	120.25	120.35	120.65	281312	340.31
2033	22-07-2010	120.3	122.00	120.25	120.75	120.90	293312	355.17
2034	21-07-2010	122.1	123.00	121.05	121.10	121.55	658666	803.56

In [5]:

```
data.describe()
#gives statistical info
```

Out[5]:

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035.000000	2035.000000	2035.000000	2035.000000	2035.00000	2.035000e+03	2035.000000
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000

In [6]:

```
data.isnull()
#caluclating null values in dataset
```

Out[6]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
2030	False	False	False	False	False	False	False	False
2031	False	False	False	False	False	False	False	False
2032	False	False	False	False	False	False	False	False
2033	False	False	False	False	False	False	False	False
2034	False	False	False	False	False	False	False	False

2035 rows × 8 columns

Sorting Data by date

In [7]:

```
data['Date'] = pd.to_datetime(data['Date'])
print(type(data.Date[0]))
```

```
<class 'pandas._libs.tslibs.timestamps.Timestamp'>
```

In [8]:

```
df = data.sort_values(by='Date')
df.head()
```

Out[8]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2004	2010-01-09	121.65	123.65	120.2	122.7	123.00	2274887	2781.63
1983	2010-01-10	121.30	124.00	121.3	123.6	123.50	711861	875.04
1962	2010-01-11	126.00	128.40	123.0	124.5	124.55	1621553	2044.44
1941	2010-01-12	113.50	116.00	113.0	114.5	114.75	628391	722.87
2026	2010-02-08	117.30	118.50	116.6	118.1	117.30	663593	779.85

In [9]:

```
df.reset_index(inplace=True)
#reset index of dataframe
```

In [10]:

```
df.head()
```

Out[10]:

	index	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2004	2010-01-09	121.65	123.65	120.2	122.7	123.00	2274887	2781.63
1	1983	2010-01-10	121.30	124.00	121.3	123.6	123.50	711861	875.04
2	1962	2010-01-11	126.00	128.40	123.0	124.5	124.55	1621553	2044.44
3	1941	2010-01-12	113.50	116.00	113.0	114.5	114.75	628391	722.87
4	2026	2010-02-08	117.30	118.50	116.6	118.1	117.30	663593	779.85

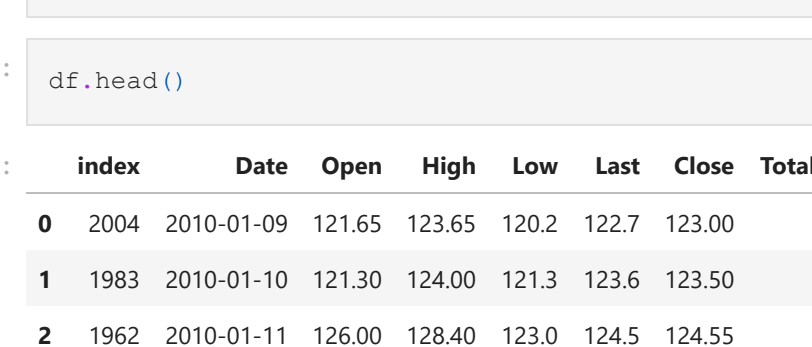
Data Visualization

In [12]:

```
plt.plot(data['Close'])
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x2906c14f370>]
```



In [13]:

```
df1 = df['Close']
df1
```

Out[13]:

```
0       123.00
1       123.50
2       124.55
3       114.75
4       117.30
         ...
2030    276.05
2031    273.85
2032    267.60
2033    265.20
2034    222.65
Name: Close, Length: 2035, dtype: float64
```

Min Max Scaler

In [15]:

```
scaler = MinMaxScaler(feature_range=(0,1))
df1 = scaler.fit_transform(np.array(df1).reshape(-1,1))
df1
```

Out[15]:

```
array([[0.17177288],
       [0.17381536],
       [0.17810458],
       ...,
       [0.76245915],
       [0.75265523],
       [0.57883987]])
```

Spliting the dataset

In [16]:

```
training_size = int(len(df1)*0.70)
test_size = len(df1)-training_size
train_data, test_data = df1[0:training_size,:],df1[training_size:len(df1),:1]
```

Converting an Array of values into a dataset matrix

In [17]:

```
def create_dataset(dataset, time_step=1):
    dataX,dataY = [],[]
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step),0]
        dataX.append(a)
        dataY.append(dataset[i+time_step,0])

    return np.array(dataX),np.array(dataY)
```

Spliting data into dataset matrix

In [18]:

```
time_step =100
X_train, y_train = create_dataset(train_data, time_step)
X_test,ytest = create_dataset(test_data, time_step)
```

In [19]:

```
print(X_train.shape),print(y_train.shape)
```

```
(1323, 100)
(1323,)
```

Out[19]:

```
(None, None)
```

In [20]:

```
print(X_test.shape), print(ytest.shape)
```

```
(510, 100)
(510,)
```

Out[20]:

```
(None, None)
```

In [23]:

```
X_train = X_train.reshape(X_train.shape[0],X_train.shape[1],1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1],1)
```

creating stacked LSTM model

In [30]:

```
from tensorflow.keras.models import Sequential
#from tensorflow.keras.models import Dense
#from tensorflow.keras.models import LSTM
from keras.layers.core import Dense
from keras.layers import Dense,LSTM,Embedding
```

In [34]:

```
model=  Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_7 (LSTM)                (None, 100, 50)           10400
_________________________________________________________________
lstm_8 (LSTM)                (None, 100, 50)           20200
_________________________________________________________________
lstm_9 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

In [37]:

```
model.fit(X_train,y_train,validation_split=0.1,epochs=60,batch_size=64,verbose=1)
```

```
Epoch 1/60
19/19 [==============================] - 18s 435ms/step - loss: 0.0093 - val_loss: 0.0025
Epoch 2/60
19/19 [==============================] - 5s 253ms/step - loss: 0.0035 - val_loss: 0.0024
Epoch 3/60
19/19 [==============================] - 5s 265ms/step - loss: 0.0027 - val_loss: 0.0022
Epoch 4/60
19/19 [==============================] - 4s 236ms/step - loss: 0.0025 - val_loss: 0.0021
Epoch 5/60
19/19 [==============================] - 4s 231ms/step - loss: 0.0025 - val_loss: 0.0021
Epoch 6/60
19/19 [==============================] - 5s 250ms/step - loss: 0.0025 - val_loss: 0.0022
Epoch 7/60
19/19 [==============================] - 5s 274ms/step - loss: 0.0025 - val_loss: 0.0021
Epoch 8/60
19/19 [==============================] - 5s 264ms/step - loss: 0.0025 - val_loss: 0.0022
Epoch 9/60
19/19 [==============================] - 4s 227ms/step - loss: 0.0025 - val_loss: 0.0021
Epoch 10/60
19/19 [==============================] - 5s 262ms/step - loss: 0.0025 - val_loss: 0.0022
Epoch 11/60
19/19 [==============================] - 5s 267ms/step - loss: 0.0025 - val_loss: 0.0022
Epoch 12/60
19/19 [==============================] - 4s 232ms/step - loss: 0.0025 - val_loss: 0.0023
Epoch 13/60
19/19 [==============================] - 5s 264ms/step - loss: 0.0026 - val_loss: 0.0021
Epoch 14/60
19/19 [==============================] - 5s 269ms/step - loss: 0.0025 - val_loss: 0.0021
Epoch 15/60
19/19 [==============================] - 5s 248ms/step - loss: 0.0024 - val_loss: 0.0022
Epoch 16/60
19/19 [==============================] - 5s 246ms/step - loss: 0.0024 - val_loss: 0.0022
Epoch 17/60
19/19 [==============================] - 5s 267ms/step - loss: 0.0024 - val_loss: 0.0022
Epoch 18/60
19/19 [==============================] - 5s 270ms/step - loss: 0.0023 - val_loss: 0.0022
Epoch 19/60
19/19 [==============================] - 4s 227ms/step - loss: 0.0023 - val_loss: 0.0021
Epoch 20/60
19/19 [==============================] - 5s 267ms/step - loss: 0.0021 - val_loss: 0.0021
Epoch 21/60
19/19 [==============================] - 5s 269ms/step - loss: 0.0020 - val_loss: 0.0020
Epoch 22/60
19/19 [==============================] - 4s 209ms/step - loss: 0.0020 - val_loss: 0.0020
Epoch 23/60
19/19 [==============================] - 5s 269ms/step - loss: 0.0021 - val_loss: 0.0021
Epoch 24/60
19/19 [==============================] - 5s 276ms/step - loss: 0.0020 - val_loss: 0.0021
Epoch 25/60
19/19 [==============================] - 4s 229ms/step - loss: 0.0019 - val_loss: 0.0022
Epoch 26/60
19/19 [==============================] - 5s 266ms/step - loss: 0.0020 - val_loss: 0.0020
Epoch 27/60
19/19 [==============================] - 5s 271ms/step - loss: 0.0019 - val_loss: 0.0021
Epoch 28/60
19/19 [==============================] - 5s 263ms/step - loss: 0.0019 - val_loss: 0.0021
Epoch 29/60
19/19 [==============================] - 4s 239ms/step - loss: 0.0020 - val_loss: 0.0021
Epoch 30/60
19/19 [==============================] - 5s 272ms/step - loss: 0.0019 - val_loss: 0.0021
Epoch 31/60
19/19 [==============================] - 5s 266ms/step - loss: 0.0021 - val_loss: 0.0020
Epoch 32/60
19/19 [==============================] - 4s 223ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 33/60
19/19 [==============================] - 5s 261ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 34/60
19/19 [==============================] - 5s 269ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 35/60
19/19 [==============================] - 4s 233ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 36/60
19/19 [==============================] - 5s 267ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 37/60
19/19 [==============================] - 5s 274ms/step - loss: 0.0019 - val_loss: 0.0022
Epoch 38/60
19/19 [==============================] - 5s 274ms/step - loss: 0.0019 - val_loss: 0.0022
Epoch 39/60
19/19 [==============================] - 5s 274ms/step - loss: 0.0019 - val_loss: 0.0021
Epoch 40/60
19/19 [==============================] - 5s 271ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 41/60
19/19 [==============================] - 5s 271ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 42/60
19/19 [==============================] - 5s 270ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 43/60
19/19 [==============================] - 5s 270ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 44/60
19/19 [==============================] - 8s 440ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 45/60
19/19 [==============================] - 3s 150ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 46/60
19/19 [==============================] - 4s 200ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 47/60
19/19 [==============================] - 3s 165ms/step - loss: 0.0019 - val_loss: 0.0026
Epoch 48/60
19/19 [==============================] - 3s 172ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 49/60
19/19 [==============================] - 3s 164ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 50/60
19/19 [==============================] - 3s 161ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 51/60
19/19 [==============================] - 3s 164ms/step - loss: 0.0018 - val_loss: 0.0020
Epoch 52/60
19/19 [==============================] - 3s 164ms/step - loss: 0.0019 - val_loss: 0.0023
Epoch 53/60
19/19 [==============================] - 3s 151ms/step - loss: 0.0019 - val_loss: 0.0019
Epoch 54/60
19/19 [==============================] - 3s 150ms/step - loss: 0.0018 - val_loss: 0.0019
Epoch 55/60
19/19 [==============================] - 3s 165ms/step - loss: 0.0018 - val_loss: 0.0021
Epoch 56/60
19/19 [==============================] - 4s 185ms/step - loss: 0.0018 - val_loss: 0.0018
Epoch 57/60
19/19 [==============================] - 3s 179ms/step - loss: 0.0018 - val_loss: 0.0018
Epoch 58/60
19/19 [==============================] - 3s 173ms/step - loss: 0.0018 - val_loss: 0.0019
Epoch 59/60
19/19 [==============================] - 3s 170ms/step - loss: 0.0018 - val_loss: 0.0018
Epoch 60/60
19/19 [==============================] - 3s 175ms/step - loss: 0.0018 - val_loss: 0.0018
```

Out[37]:

```
<keras.callbacks.History at 0x2906d846370>
```

Prediction and checking performance

In [38]:

```
test_predict = model.predict(X_test)
```

In [40]:

```
test_predicted = scaler.inverse_transform(test_predict)
test_predicted
```

Out[40]:
```
array([[0.19877563],
       [0.20863256],
       [0.21439093],
       [0.21683013],
       [0.22098458],
       [0.2257538 ],
       [0.23061134],
       [0.23428102],
       [0.23549809],
       [0.23440348],
       [0.23179586],
       [0.2294604 ],
       [0.2264227 ],
       [0.2241689 ],
       [0.22316475],
       [0.22030309],
       [0.21719496],
       [0.208776  ],
       [0.19747512],
       [0.18736435],
       [0.18204911],
       [0.18559907],
       [0.19935603],
       [0.21022528],
       [0.22365783],
       [0.23899241],
       [0.25223222],
       [0.2621811 ],
       [0.2690742 ],
       [0.27201223],
       [0.27169237],
       [0.2694829 ],
       [0.26507705],
       [0.2579176 ],
       [0.24768731],
       [0.23754838],
       [0.22918957],
       [0.22429195],
       [0.21218273],
       [0.20055102],
       [0.19359952],
       [0.19147928],
       [0.19825895],
       [0.21189325],
       [0.22222334],
       [0.22776055],
       [0.23065765],
       [0.23046583],
       [0.22806075],
       [0.22244759],
       [0.2149372 ],
       [0.20629841],
       [0.19629173],
       [0.18659042],
       [0.1779144 ],
       [0.1704881 ],
       [0.16460063],
       [0.16404149],
       [0.15744159],
       [0.15008022],
       [0.14389244],
       [0.14341323],
       [0.1485012 ],
       [0.15637182],
       [0.16178924],
       [0.16502209],
       [0.16706105],
       [0.1693511 ],
       [0.17224406],
       [0.17570646],
       [0.17858931],
       [0.18109366],
       [0.1826345 ],
       [0.183276  ],
       [0.1827614 ],
       [0.18225756],
       [0.18144503],
       [0.18062292],
       [0.17979683],
       [0.18091425],
       [0.18663186],
       [0.19663906],
       [0.21165475],
       [0.23627423],
       [0.27275732],
       [0.32903576],
       [0.33876806],
       [0.31897613],
       [0.2852512 ],
       [0.24880567],
       [0.21689828],
       [0.19235164],
       [0.17819896],
       [0.17561047],
       [0.18382582],
       [0.20080426],
       [0.2232731 ],
       [0.24680077],
       [0.26635742],
       [0.2849499 ],
       [0.30080643],
       [0.31622148],
       [0.32583767],
       [0.3314623 ],
       [0.3515619 ],
       [0.34814095],
       [0.32949078],
       [0.3029544 ],
       [0.27488342],
       [0.24866048],
       [0.22647361],
       [0.2091589 ],
       [0.19675659],
       [0.18963093],
       [0.18820119],
       [0.19232741],
       [0.19770074],
       [0.21014816],
       [0.22833677],
       [0.2514416 ],
       [0.27613875],
       [0.29644868],
       [0.3153276 ],
       [0.34280673],
       [0.37650394],
       [0.3775213 ],
       [0.35929567],
       [0.33378533],
       [0.30741522],
       [0.28404695],
       [0.26491755],
       [0.24975893],
       [0.24153055],
       [0.23919651],
       [0.24073897],
       [0.24594866],
       [0.2538405 ],
       [0.26320618],
       [0.2729892 ],
       [0.27427313],
       [0.28023762],
       [0.28751937],
       [0.29954997],
       [0.3209055 ],
       [0.34735984],
       [0.39691615],
       [0.4055079 ],
       [0.38887835],
       [0.3603642 ],
       [0.3298679 ],
       [0.30292985],
       [0.28150147],
       [0.25874406],
       [0.25774014],
       [0.26208803],
       [0.27081996],
       [0.2735822 ],
       [0.28191605],
       [0.29409778],
       [0.30791843],
       [0.3207568 ],
       [0.3439389 ],
       [0.37302995],
       [0.42389756],
       [0.4317357 ],
       [0.41417024],
       [0.38466635],
       [0.35084188],
       [0.31930855],
       [0.29251063],
       [0.27117828],
       [0.25554812],
       [0.2469172 ],
       [0.24644081],
       [0.2522387 ],
       [0.26247522],
       [0.27632028],
       [0.28327134],
       [0.29109526],
       [0.29916963],
       [0.3089851 ],
       [0.3174784 ],
       [0.3226803 ],
       [0.34089777],
       [0.36402318],
       [0.3922261 ],
       [0.43109065],
       [0.4282217 ],
       [0.4034908 ],
       [0.369779  ],
       [0.33475655],
       [0.30262044],
       [0.27663493],
       [0.25880587],
       [0.24968053],
       [0.24865662],
       [0.25452068],
       [0.26731598],
       [0.28497148],
       [0.30556384],
       [0.32292455],
       [0.3364513 ],
       [0.34827632],
       [0.35831845],
       [0.36478364],
       [0.37052932],
       [0.3846943 ],
       [0.40562767],
       [0.4450674 ],
       [0.44834968],
       [0.4306308 ],
       [0.4041233 ],
       [0.37460554],
       [0.34678286],
       [0.32175907],
       [0.30187237],
       [0.28795195],
       [0.28172433],
       [0.28400874],
       [0.29285938],
       [0.30748004],
       [0.3261604 ],
       [0.3385142 ],
       [0.34537596],
       [0.35346967],
       [0.36264098],
       [0.3746283 ],
       [0.3995345 ],
       [0.43189797],
       [0.4835994 ],
       [0.49245384],
       [0.48208138],
       [0.46112955],
       [0.4380242 ],
       [0.41646177],
       [0.39557567],
       [0.3791833 ],
       [0.36653852],
       [0.35770774],
       [0.35221854],
       [0.35321632],
       [0.36018193],
       [0.3468857 ],
       [0.332862  ],
       [0.32363072],
       [0.32616994],
       [0.33756635],
       [0.35576862],
       [0.39129296],
       [0.44145697],
       [0.47500607],
       [0.49281454],
       [0.49701136],
       [0.4929422 ],
       [0.48471186],
       [0.47317362],
       [0.4596439 ],
       [0.44199425],
       [0.42208943],
       [0.4058092 ],
       [0.3918645 ],
       [0.38326454],
       [0.3810567 ],
       [0.35718477],
       [0.3333943 ],
       [0.31524014],
       [0.31159344],
       [0.32167348],
       [0.34408528],
       [0.3669338 ],
       [0.40465072],
       [0.45352876],
       [0.4865621 ],
       [0.50223213],
       [0.50448954],
       [0.49650562],
       [0.48211533],
       [0.464486  ],
       [0.44846302],
       [0.4328074 ],
       [0.41995695],
       [0.4133531 ],
       [0.41307876],
       [0.4165135 ],
       [0.38830322],
       [0.3566776 ],
       [0.33150417],
       [0.32277343],
       [0.3266984 ],
       [0.3555309 ],
       [0.39586934],
       [0.46994644],
       [0.52084726],
       [0.5530856 ],
       [0.5705283 ],
       [0.57685083],
       [0.5783484 ],
       [0.5813461 ],
       [0.5821593 ],
       [0.5805773 ],
       [0.5768878 ],
       [0.57297313],
       [0.5681247 ],
       [0.56701684],
       [0.56487566],
       [0.5657822 ],
       [0.50843984],
       [0.44591287],
       [0.3956595 ],
       [0.36625478],
       [0.36659724],
       [0.40235388],
       [0.4535268 ],
       [0.5341223 ],
       [0.6091457 ],
       [0.66136223],
       [0.6900344 ],
       [0.70131475],
       [0.70405346],
       [0.69977427],
       [0.69039285],
       [0.6789057 ],
       [0.6671546 ],
       [0.6547349 ],
       [0.6424684 ],
       [0.6346222 ],
       [0.6278652 ],
       [0.6129975 ],
       [0.5926108 ],
       [0.5705284 ],
       [0.5438053 ],
       [0.55909586],
       [0.5923401 ],
       [0.62855226],
       [0.6565895 ],
       [0.67780906],
       [0.6925581 ],
       [0.7009816 ],
       [0.7035565 ],
       [0.702772  ],
       [0.6990972 ],
       [0.6953444 ],
       [0.6859847 ],
       [0.68468535],
       [0.67089236],
       [0.6559852 ],
       [0.65020233],
       [0.6378669 ],
       [0.61427647],
       [0.6107022 ],
       [0.61628634],
       [0.6243042 ],
       [0.629047  ],
       [0.62969553],
       [0.63032   ],
       [0.62901664],
       [0.6307121 ],
       [0.6342121 ],
       [0.63796556],
       [0.64066494],
       [0.65675783],
       [0.6586332 ],
       [0.65853393],
       [0.6540451 ],
       [0.6289456 ],
       [0.60055226],
       [0.5924366 ],
       [0.595454  ],
       [0.60377717],
       [0.61001   ],
       [0.6094493 ],
       [0.606939  ],
       [0.60484886],
       [0.60191995],
       [0.5991448 ],
       [0.59903026],
       [0.60100186],
       [0.60158384],
       [0.6238219 ],
       [0.6328535 ],
       [0.6415329 ],
       [0.6339103 ],
       [0.6296079 ],
       [0.60522664],
       [0.5976968 ],
       [0.6006105 ],
       [0.6083956 ],
       [0.6180308 ],
       [0.627283  ],
       [0.6343643 ],
       [0.6377186 ],
       [0.6383463 ],
       [0.63987154],
       [0.6420572 ],
       [0.64422613],
       [0.64800733],
       [0.6579777 ],
       [0.6514807 ],
       [0.64062315],
       [0.6288326 ],
       [0.61559033],
       [0.6081062 ],
       [0.58571255],
       [0.5760689 ],
       [0.57003933],
       [0.5678906 ],
       [0.5719031 ],
       [0.5778707 ],
       [0.5838211 ],
       [0.5904355 ],
       [0.5972764 ],
       [0.6042481 ],
       [0.610542  ],
       [0.6154498 ],
       [0.6163239 ],
       [0.61691666],
       [0.617901  ],
       [0.6166172 ],
       [0.61644936],
       [0.61510164],
       [0.60977554],
       [0.6031688 ],
       [0.5860772 ],
       [0.56026566],
       [0.5543793 ],
       [0.5630329 ],
       [0.57530403],
       [0.5886157 ],
       [0.596734  ],
       [0.5987884 ],
       [0.5970076 ],
       [0.59577906],
       [0.59628725],
       [0.5988521 ],
       [0.60122794],
       [0.60244125],
       [0.605133  ],
       [0.6123482 ],
       [0.6153068 ],
       [0.6261472 ],
       [0.6265145 ],
       [0.61019033],
       [0.58395123],
       [0.5727208 ],
       [0.56477   ],
       [0.5624791 ],
       [0.5601452 ],
       [0.55676574],
       [0.5564801 ],
       [0.5602019 ],
       [0.56666774],
       [0.5694465 ],
       [0.570824  ],
       [0.57232624],
       [0.57320833],
       [0.57500273],
       [0.60514843],
       [0.62544477],
       [0.6333447 ],
       [0.6395249 ],
       [0.6331128 ],
       [0.6093834 ],
       [0.5808501 ],
       [0.55590326],
       [0.5380355 ],
       [0.52894557],
       [0.52768886],
       [0.5299064 ],
       [0.5326908 ],
       [0.53480065],
       [0.539865  ],
       [0.5449145 ],
       [0.5501902 ],
       [0.5555255 ],
       [0.5592697 ],
       [0.59456515],
       [0.62064856],
       [0.632039  ],
       [0.637148  ],
       [0.6428363 ],
       [0.63935924],
       [0.6162916 ],
       [0.5866144 ],
       [0.55950296],
       [0.5378702 ],
       [0.5233345 ],
       [0.51587313],
       [0.51417404],
       [0.51829034],
       [0.5252716 ],
       [0.5332518 ],
       [0.54144365],
       [0.581919  ],
       [0.6125869 ],
       [0.63582885],
       [0.6453069 ],
       [0.62906766],
       [0.59951943],
       [0.60473543],
       [0.6077304 ],
       [0.614234  ],
       [0.6103852 ],
       [0.6047071 ],
       [0.5777329 ],
       [0.5901386 ],
       [0.6018545 ],
       [0.6078624 ],
       [0.60932916],
       [0.6062948 ]], dtype=float32)
```

Performance

In [42]:
```
import math
from sklearn.metrics import mean_squared_error
```

In [43]:
```
performance = math.sqrt(mean_squared_error(ytest,test_predict))
performance
```

Out[43]:
```
0.14648360181570982
```

In [ ]: