google-stock-price-prediction

July 3, 2023

[28]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.losses import Huber
from keras.optimizers import Adam
```

[6]:
```
df = pd.read_csv("Google_Stock_Price_Test.csv")
```

[7]:
```
print(df.head())
```

```
       Date    Open    High     Low   Close     Volume
0  1/3/2017  778.81  789.63  775.80  786.14  1,657,300
1  1/4/2017  788.36  791.34  783.16  786.90  1,073,000
2  1/5/2017  786.08  794.48  785.02  794.02  1,335,200
3  1/6/2017  795.26  807.90  792.20  806.15  1,640,200
4  1/9/2017  806.40  809.97  802.83  806.65  1,272,400
```

[8]:
```
df = df.set_index('Date')
```

[17]:
```
closing_price = df['Close']
```

[18]:
```
# Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_closing_price = scaler.fit_transform(closing_price.values.reshape(-1, 1))
```

[19]:
```
# Split the data into training and testing sets
train_size = int(len(scaled_closing_price) * 0.8)
test_size = len(scaled_closing_price) - train_size

train_data = scaled_closing_price[:train_size]
test_data = scaled_closing_price[train_size:]
```

[20]:
```
# Create the LSTM model
model = Sequential()
```

```
model.add(LSTM(128, activation='tanh', input_shape=(1, 1)))
model.add(Dense(1))
```

[29]:
```
model.compile(loss=Huber(), optimizer=Adam(learning_rate=0.001))
```

[30]:
```
# Train the model
model.fit(train_data, train_data, epochs=100, batch_size=32)
```

```
Epoch 1/100
1/1 [==============================] - 2s 2s/step - loss: 0.0083
Epoch 2/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0080
Epoch 3/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0077
Epoch 4/100
1/1 [==============================] - 0s 12ms/step - loss: 0.0074
Epoch 5/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0071
Epoch 6/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0068
Epoch 7/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0065
Epoch 8/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0063
Epoch 9/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0060
Epoch 10/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0057
Epoch 11/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0055
Epoch 12/100
1/1 [==============================] - 0s 12ms/step - loss: 0.0052
Epoch 13/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0050
Epoch 14/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0047
Epoch 15/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0045
Epoch 16/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0042
Epoch 17/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0040
Epoch 18/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0038
Epoch 19/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0036
Epoch 20/100
```

```
1/1 [==============================] - 0s 10ms/step - loss: 0.0034
Epoch 21/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0032
Epoch 22/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0030
Epoch 23/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0028
Epoch 24/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0026
Epoch 25/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0024
Epoch 26/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0022
Epoch 27/100
1/1 [==============================] - 0s 12ms/step - loss: 0.0021
Epoch 28/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0019
Epoch 29/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0018
Epoch 30/100
1/1 [==============================] - 0s 10ms/step - loss: 0.0016
Epoch 31/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0015
Epoch 32/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0014
Epoch 33/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0012
Epoch 34/100
1/1 [==============================] - 0s 12ms/step - loss: 0.0011
Epoch 35/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0010
Epoch 36/100
1/1 [==============================] - 0s 13ms/step - loss: 9.0728e-04
Epoch 37/100
1/1 [==============================] - 0s 12ms/step - loss: 8.1226e-04
Epoch 38/100
1/1 [==============================] - 0s 8ms/step - loss: 7.2394e-04
Epoch 39/100
1/1 [==============================] - 0s 10ms/step - loss: 6.4217e-04
Epoch 40/100
1/1 [==============================] - 0s 12ms/step - loss: 5.6683e-04
Epoch 41/100
1/1 [==============================] - 0s 9ms/step - loss: 4.9775e-04
Epoch 42/100
1/1 [==============================] - 0s 9ms/step - loss: 4.3476e-04
Epoch 43/100
1/1 [==============================] - 0s 11ms/step - loss: 3.7764e-04
Epoch 44/100
```

```
1/1 [==============================] - 0s 11ms/step - loss: 3.2617e-04
Epoch 45/100
1/1 [==============================] - 0s 9ms/step - loss: 2.8007e-04
Epoch 46/100
1/1 [==============================] - 0s 12ms/step - loss: 2.3908e-04
Epoch 47/100
1/1 [==============================] - 0s 11ms/step - loss: 2.0289e-04
Epoch 48/100
1/1 [==============================] - 0s 9ms/step - loss: 1.7121e-04
Epoch 49/100
1/1 [==============================] - 0s 12ms/step - loss: 1.4374e-04
Epoch 50/100
1/1 [==============================] - 0s 14ms/step - loss: 1.2016e-04
Epoch 51/100
1/1 [==============================] - 0s 11ms/step - loss: 1.0018e-04
Epoch 52/100
1/1 [==============================] - 0s 14ms/step - loss: 8.3484e-05
Epoch 53/100
1/1 [==============================] - 0s 12ms/step - loss: 6.9746e-05
Epoch 54/100
1/1 [==============================] - 0s 10ms/step - loss: 5.8651e-05
Epoch 55/100
1/1 [==============================] - 0s 11ms/step - loss: 4.9892e-05
Epoch 56/100
1/1 [==============================] - 0s 12ms/step - loss: 4.3171e-05
Epoch 57/100
1/1 [==============================] - 0s 11ms/step - loss: 3.8209e-05
Epoch 58/100
1/1 [==============================] - 0s 12ms/step - loss: 3.4746e-05
Epoch 59/100
1/1 [==============================] - 0s 10ms/step - loss: 3.2540e-05
Epoch 60/100
1/1 [==============================] - 0s 14ms/step - loss: 3.1362e-05
Epoch 61/100
1/1 [==============================] - 0s 12ms/step - loss: 3.1004e-05
Epoch 62/100
1/1 [==============================] - 0s 13ms/step - loss: 3.1274e-05
Epoch 63/100
1/1 [==============================] - 0s 13ms/step - loss: 3.2000e-05
Epoch 64/100
1/1 [==============================] - 0s 13ms/step - loss: 3.3034e-05
Epoch 65/100
1/1 [==============================] - 0s 12ms/step - loss: 3.4250e-05
Epoch 66/100
1/1 [==============================] - 0s 12ms/step - loss: 3.5545e-05
Epoch 67/100
1/1 [==============================] - 0s 13ms/step - loss: 3.6835e-05
Epoch 68/100
```

```
1/1 [==============================] - 0s 13ms/step - loss: 3.8052e-05
Epoch 69/100
1/1 [==============================] - 0s 15ms/step - loss: 3.9143e-05
Epoch 70/100
1/1 [==============================] - 0s 12ms/step - loss: 4.0069e-05
Epoch 71/100
1/1 [==============================] - 0s 11ms/step - loss: 4.0806e-05
Epoch 72/100
1/1 [==============================] - 0s 12ms/step - loss: 4.1340e-05
Epoch 73/100
1/1 [==============================] - 0s 12ms/step - loss: 4.1670e-05
Epoch 74/100
1/1 [==============================] - 0s 11ms/step - loss: 4.1801e-05
Epoch 75/100
1/1 [==============================] - 0s 11ms/step - loss: 4.1747e-05
Epoch 76/100
1/1 [==============================] - 0s 12ms/step - loss: 4.1525e-05
Epoch 77/100
1/1 [==============================] - 0s 11ms/step - loss: 4.1154e-05
Epoch 78/100
1/1 [==============================] - 0s 13ms/step - loss: 4.0655e-05
Epoch 79/100
1/1 [==============================] - 0s 12ms/step - loss: 4.0051e-05
Epoch 80/100
1/1 [==============================] - 0s 16ms/step - loss: 3.9366e-05
Epoch 81/100
1/1 [==============================] - 0s 11ms/step - loss: 3.8624e-05
Epoch 82/100
1/1 [==============================] - 0s 15ms/step - loss: 3.7845e-05
Epoch 83/100
1/1 [==============================] - 0s 15ms/step - loss: 3.7050e-05
Epoch 84/100
1/1 [==============================] - 0s 11ms/step - loss: 3.6257e-05
Epoch 85/100
1/1 [==============================] - 0s 11ms/step - loss: 3.5480e-05
Epoch 86/100
1/1 [==============================] - 0s 11ms/step - loss: 3.4731e-05
Epoch 87/100
1/1 [==============================] - 0s 11ms/step - loss: 3.4021e-05
Epoch 88/100
1/1 [==============================] - 0s 11ms/step - loss: 3.3358e-05
Epoch 89/100
1/1 [==============================] - 0s 11ms/step - loss: 3.2747e-05
Epoch 90/100
1/1 [==============================] - 0s 11ms/step - loss: 3.2193e-05
Epoch 91/100
1/1 [==============================] - 0s 12ms/step - loss: 3.1696e-05
Epoch 92/100
```

```
1/1 [==============================] - 0s 12ms/step - loss: 3.1258e-05
Epoch 93/100
1/1 [==============================] - 0s 12ms/step - loss: 3.0875e-05
Epoch 94/100
1/1 [==============================] - 0s 11ms/step - loss: 3.0545e-05
Epoch 95/100
1/1 [==============================] - 0s 11ms/step - loss: 3.0266e-05
Epoch 96/100
1/1 [==============================] - 0s 12ms/step - loss: 3.0033e-05
Epoch 97/100
1/1 [==============================] - 0s 12ms/step - loss: 2.9841e-05
Epoch 98/100
1/1 [==============================] - 0s 11ms/step - loss: 2.9686e-05
Epoch 99/100
1/1 [==============================] - 0s 11ms/step - loss: 2.9562e-05
Epoch 100/100
1/1 [==============================] - 0s 10ms/step - loss: 2.9465e-05
```

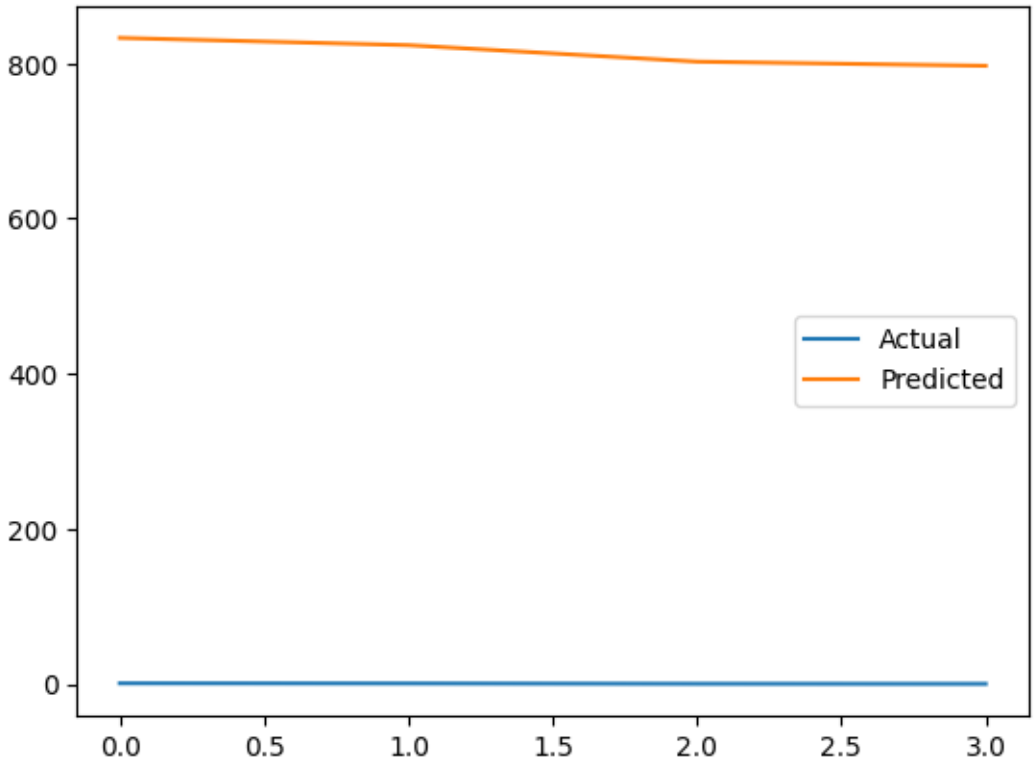
[30]:
```
<keras.callbacks.History at 0x7f04453cde40>
```

[31]:
```
# Make predictions
predictions = model.predict(test_data)
```

```
1/1 [==============================] - 1s 627ms/step
```

[32]:
```
# Inverse scale the predictions
predictions = scaler.inverse_transform(predictions)
```

[35]:
```
# Plot the predictions
plt.plot(test_data, label='Actual')
plt.plot(predictions, label='Predicted')
plt.legend()
plt.show()
```


800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Actual
Predicted

[ ]: