COSC 4337

RNN_First

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
ta = pd.read_csv('FB_training_data.csv')
```

[3]:
```
training_data = ta.iloc[:, 1].values
```

[4]:
```
from sklearn.preprocessing import MinMaxScaler
```

[5]:
```
scaler = MinMaxScaler()
```

[6]:
```
training_data = scaler.fit_transform(training_data.reshape(-1, 1))
```

[7]:
```
x_training_data = []

y_training_data =[]
```

[8]:
```
for i in range(40, len(training_data)):

    x_training_data.append(training_data[i-40:i, 0])

    y_training_data.append(training_data[i, 0])
```

[9]:
```
x_training_data = np.array(x_training_data)

y_training_data = np.array(y_training_data)
```

[10]:
```
print(x_training_data.shape)

print(y_training_data.shape)
```

```
(1218, 40)
(1218,)
```

[11]:
```
x_training_data = np.reshape(x_training_data, (x_training_data.shape[0],

                                               x_training_data.shape[1],
```

```
                                                 1))
```

[12]:
```
print(x_training_data.shape)
```

```
(1218, 40, 1)
```

[13]:
```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import LSTM

from tensorflow.keras.layers import Dropout
```

[14]:
```
rnn = Sequential()
```

[15]:
```
rnn.add(LSTM(units = 45, return_sequences = True, input_shape =␣
↪(x_training_data.shape[1], 1)))
```

```
WARNING:tensorflow:From C:\Users\RizkN\.conda\envs\tf1\lib\site-
packages\tensorflow_core\python\ops\resource_variable_ops.py:1630: calling
BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops)
with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```

[16]:
```
rnn.add(Dropout(0.2))
```

[17]:
```
rnn.add(LSTM(units = 45, return_sequences = True))

rnn.add(Dropout(0.2))

rnn.add(LSTM(units = 45, return_sequences = True))

rnn.add(Dropout(0.2))

rnn.add(LSTM(units = 45))

rnn.add(Dropout(0.2))
```

[18]:
```
rnn.add(Dense(units = 1))
```

[19]:
```
rnn.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

[20]:
```
rnn.fit(x_training_data, y_training_data, epochs = 100, batch_size = 32)
```

```
WARNING:tensorflow:From C:\Users\RizkN\.conda\envs\tf1\lib\site-
packages\tensorflow_core\python\ops\math_grad.py:1424: where (from
tensorflow.python.ops.array_ops) is deprecated and will be removed in a future
```

```
version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Train on 1218 samples
Epoch 1/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0413
Epoch 2/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0074
Epoch 3/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0076
Epoch 4/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0063
Epoch 5/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0053
Epoch 6/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0055
Epoch 7/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0056
Epoch 8/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0048
Epoch 9/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0057
Epoch 10/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0051
Epoch 11/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0059
Epoch 12/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0046
Epoch 13/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0046
Epoch 14/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0040
Epoch 15/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0045
Epoch 16/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0042
Epoch 17/100
1218/1218 [==============================] - 2s 2ms/sample - loss: 0.0042
Epoch 18/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0043
Epoch 19/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0044
Epoch 20/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0041
Epoch 21/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0040
Epoch 22/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0041
```

```
Epoch 23/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0044
Epoch 24/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0042
Epoch 25/100
1218/1218 [==============================] - 3s 2ms/sample - loss: 0.0039
Epoch 26/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0035
Epoch 27/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0034
Epoch 28/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0040
Epoch 29/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0036
Epoch 30/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0035
Epoch 31/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0035
Epoch 32/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0060
Epoch 33/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0035
Epoch 34/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0034
Epoch 35/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0034
Epoch 36/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0040
Epoch 37/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0034
Epoch 38/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0030
Epoch 39/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0031
Epoch 40/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0032
Epoch 41/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0033
Epoch 42/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0033
Epoch 43/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0031
Epoch 44/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0036
Epoch 45/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0030
Epoch 46/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0026
```

```
Epoch 47/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0026
Epoch 48/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0030
Epoch 49/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0030
Epoch 50/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0026
Epoch 51/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0026
Epoch 52/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0027
Epoch 53/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0027
Epoch 54/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0024
Epoch 55/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0029
Epoch 56/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0023
Epoch 57/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0025
Epoch 58/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0024
Epoch 59/100
1218/1218 [==============================] - 3s 3ms/sample - loss: 0.0024
Epoch 60/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0024
Epoch 61/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0026
Epoch 62/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0024
Epoch 63/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0023
Epoch 64/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 65/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 66/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 67/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 68/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0025
Epoch 69/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0021
Epoch 70/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0023
```

```
Epoch 71/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 72/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0025
Epoch 73/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0021
Epoch 74/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 75/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 76/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0021
Epoch 77/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0021
Epoch 78/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0020
Epoch 79/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0021
Epoch 80/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 81/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019 2s -
ETA: 0s - l
Epoch 82/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 83/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 84/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 85/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0019
Epoch 86/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0020
Epoch 87/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
Epoch 88/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0022
Epoch 89/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0018
Epoch 90/100
1218/1218 [==============================] - 4s 4ms/sample - loss: 0.0016
Epoch 91/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0016
Epoch 92/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0018
Epoch 93/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
Epoch 94/100
```

```
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
Epoch 95/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0016
Epoch 96/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0015 0s -
loss: 0
Epoch 97/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
Epoch 98/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
Epoch 99/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0016
Epoch 100/100
1218/1218 [==============================] - 4s 3ms/sample - loss: 0.0017
```

[20]:
```
<tensorflow.python.keras.callbacks.History at 0x12a79a1c308>
```

[21]:
```
test_data = pd.read_csv('FB_test_data.csv')

test_data = test_data.iloc[:, 1].values
```

[22]:
```
unscaled_training_data = pd.read_csv('FB_training_data.csv')

unscaled_test_data = pd.read_csv('FB_test_data.csv')
```

[23]:
```
all_data = pd.concat((unscaled_training_data['Open'],␣
↪unscaled_test_data['Open']), axis = 0)
```

[24]:
```
x_test_data = all_data[len(all_data) - len(test_data) - 40:].values
```

[25]:
```
x_test_data = np.reshape(x_test_data, (-1, 1))
```

[26]:
```
x_test_data = scaler.transform(x_test_data)
```

[27]:
```
final_x_test_data = []

for i in range(40, len(x_test_data)):

    final_x_test_data.append(x_test_data[i-40:i, 0])

final_x_test_data = np.array(final_x_test_data)
```

[28]:
```
final_x_test_data = np.reshape(final_x_test_data, (final_x_test_data.shape[0],

                                                   final_x_test_data.shape[1],

                                                   1))
```

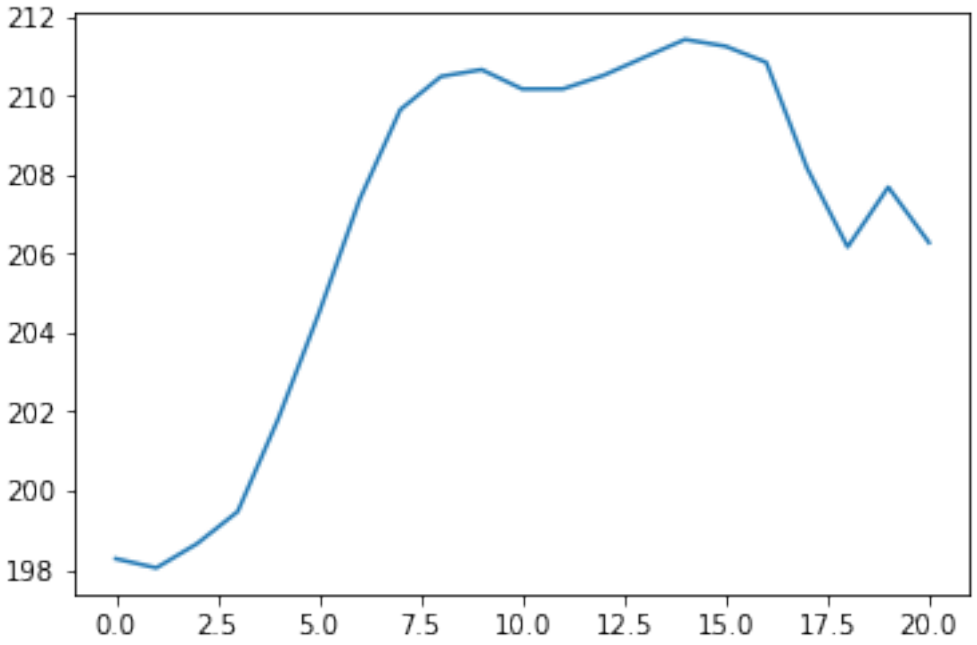
[29]:
```
predictions = rnn.predict(final_x_test_data)
```

[30]:
```
unscaled_predictions = scaler.inverse_transform(predictions)

plt.clf() #This clears the first prediction plot from our canvas

plt.plot(unscaled_predictions)
```

[30]:
```
[<matplotlib.lines.Line2D at 0x12a7b2e4188>]
```

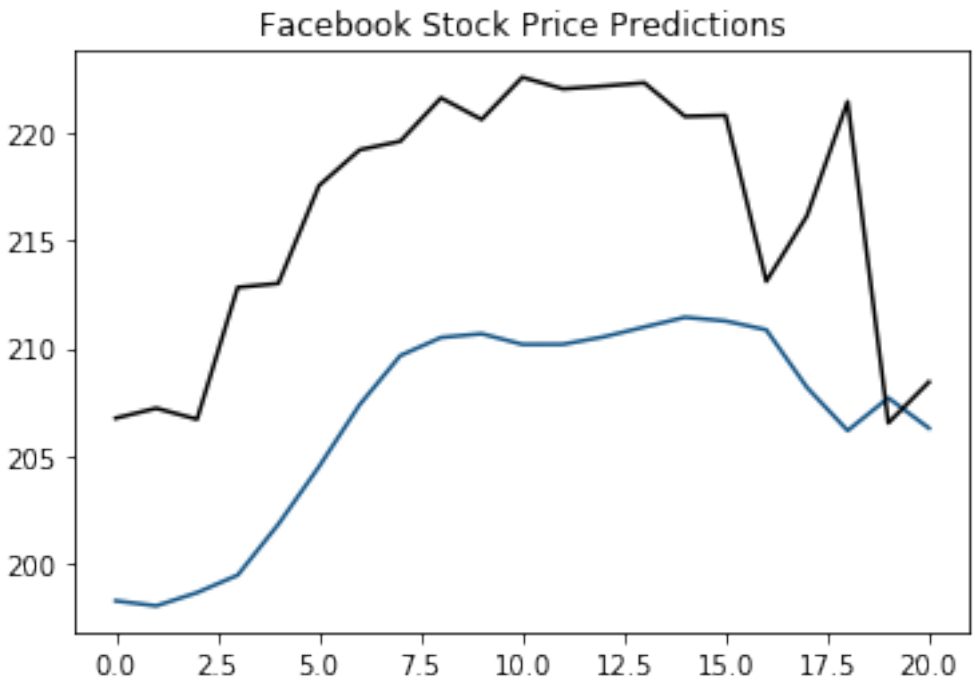


[31]:
```
plt.plot(unscaled_predictions, color = '#135485', label = "Predictions")

plt.plot(test_data, color = 'black', label = "Real Data")

plt.title('Facebook Stock Price Predictions')
```

[31]:
```
Text(0.5, 1.0, 'Facebook Stock Price Predictions')
```


Facebook Stock Price Predictions
220
215
210
205
200
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

[ ]: