ASSIGNMENT:7

Based on this input, predict the CLOSING STOCK VALUE of INFOSYS after 1 month, 6 months and 1 year, using the following DEEP LEARNING models:

(1) Normal conventional RNN;

(2) LSTM;

(3) Bidirectional LSTM

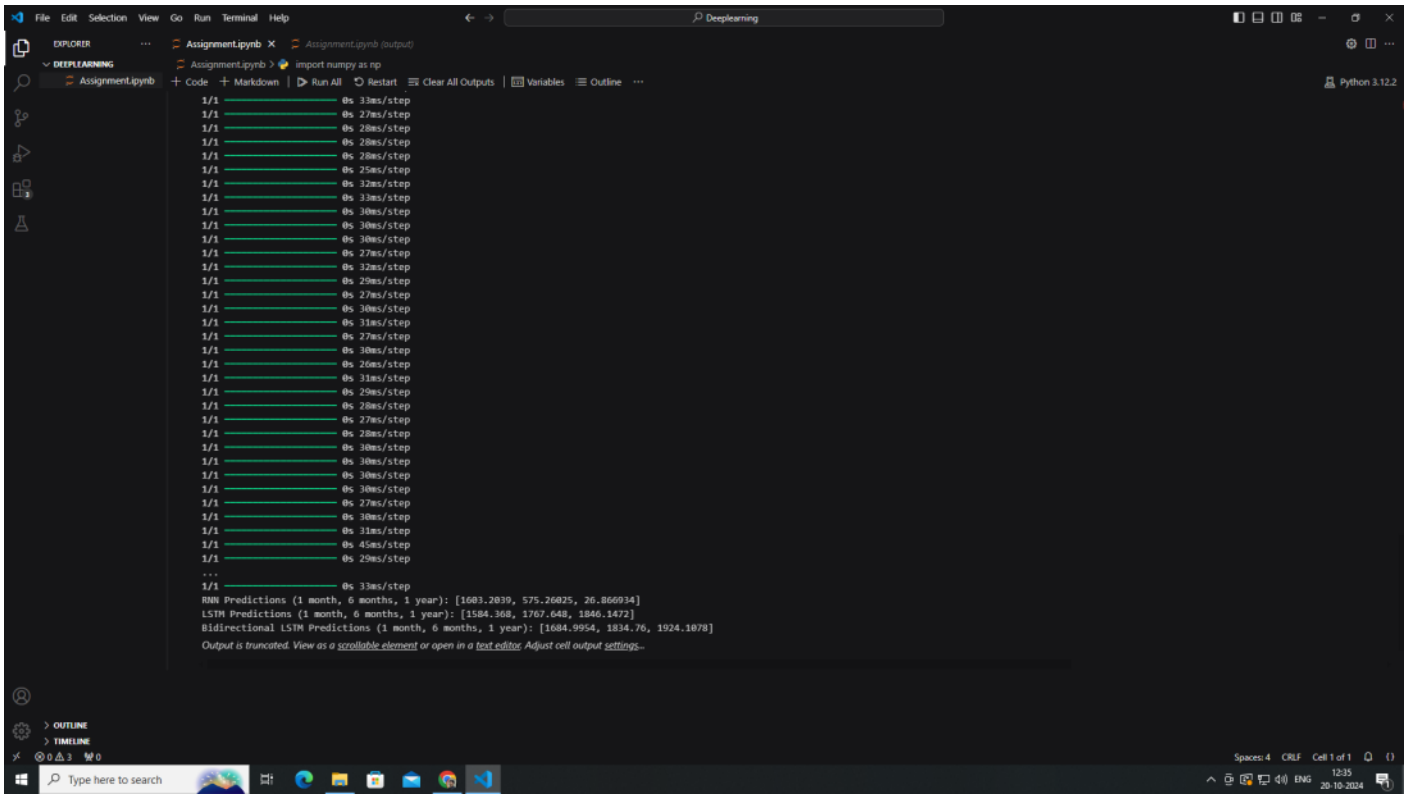
CODE:

```
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 Dense, SimpleRNN, LSTM, Bidirectional
from keras.layers import Dropout
import yfinance as yf

# Step 1: Load the data
# Download Infosys data for the past 5 years using yahoofinance
ticker = 'INFY.NS'
data = yf.download(ticker, start='2018-01-01', end='2023-01-01')
data = data['Close']

# Step 2: Preprocessing the data

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(np.array(data).reshape(-1, 1))


def create_dataset(dataset, time_step=100):
    X_data, Y_data = [], []
    for i in range(len(dataset)-time_step-1):
        X_data.append(dataset[i:(i+time_step), 0])
        Y_data.append(dataset[i+time_step, 0])
    return np.array(X_data), np.array(Y_data)


time_step = 100
X, Y = create_dataset(scaled_data, time_step)
X = X.reshape(X.shape[0], X.shape[1], 1)

# Split into training (80%) and testing (20%)
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
```

```
Y_train, Y_test = Y[:train_size], Y[train_size:]



# Model 1: Simple RNN
def build_rnn():
    model = Sequential()
    model.add(SimpleRNN(50, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
    model.add(Dropout(0.2))
    model.add(SimpleRNN(50, return_sequences=False))
    model.add(Dropout(0.2))
    model.add(Dense(1))  # Output layer
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

# Model 2: LSTM
def build_lstm():
    model = Sequential()
    model.add(LSTM(50, return_sequences=True, input_shape=(X_train.shape[1],
1)))
    model.add(Dropout(0.2))
    model.add(LSTM(50, return_sequences=False))
    model.add(Dropout(0.2))
    model.add(Dense(1))  # Output layer
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

# Model 3: Bidirectional LSTM
def build_bidirectional_lstm():
    model = Sequential()
    model.add(Bidirectional(LSTM(50, return_sequences=True),
input_shape=(X_train.shape[1], 1)))
    model.add(Dropout(0.2))
    model.add(Bidirectional(LSTM(50, return_sequences=False)))
    model.add(Dropout(0.2))
    model.add(Dense(1))  # Output layer
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

# Train and predict for each model
def train_and_predict(model, X_train, Y_train, X_test):
    # Train the model
    model.fit(X_train, Y_train, epochs=20, batch_size=64, verbose=1)

    # Predict future prices (1 month, 6 months, 1 year)
    future_steps = [22, 132, 252]  # Approx. trading days in 1 month, 6
months, and 1 year
```

```
    predictions = []
    for step in future_steps:

        last_data = X_test[-1]
        future_pred = []
        for _ in range(step):
            pred = model.predict(last_data.reshape(1, time_step, 1))
            future_pred.append(pred[0, 0])
            last_data = np.append(last_data[1:], pred)

        # Scale back the predictions to original values
        future_pred = scaler.inverse_transform(np.array(future_pred).reshape(-
1, 1))
        predictions.append(future_pred[-1, 0])

    return predictions


rnn_model = build_rnn()
lstm_model = build_lstm()
bidirectional_lstm_model = build_bidirectional_lstm()

# Train and predict with each model
rnn_predictions = train_and_predict(rnn_model, X_train, Y_train, X_test)
lstm_predictions = train_and_predict(lstm_model, X_train, Y_train, X_test)
bidirectional_lstm_predictions = train_and_predict(bidirectional_lstm_model,
X_train, Y_train, X_test)

# Step 4: Print the results
print("RNN Predictions (1 month, 6 months, 1 year):", rnn_predictions)
print("LSTM Predictions (1 month, 6 months, 1 year):", lstm_predictions)
print("Bidirectional LSTM Predictions (1 month, 6 months, 1 year):",
bidirectional_lstm_predictions)
```

OUTPUT:


File Edit Selection View Go Run Terminal Help
Deeplearning
EXPLORER
DEEPLEARNING
Assignment.ipynb
Assignment.ipynb (output)
Assignment.ipynb > import numpy as np
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline
Python 3.12.2
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
RNN Predictions (1 month, 6 months, 1 year): [1603.2039, 575.26025, 26.866934]
LSTM Predictions (1 month, 6 months, 1 year): [1584.368, 1767.648, 1846.1472]
Bidirectional LSTM Predictions (1 month, 6 months, 1 year): [1684.9954, 1834.76, 1924.1078]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
OUTLINE
TIMELINE
Spaces: 4 CRLF Cell 1 of 1
Type here to search
ENG
12:35
20-10-2024