Stock Price Prediction Using Neural Networks

Overview

The script builds and evaluates three different neural network models LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), and a DNN (Dense Neural Network), to predict stock prices based on historical data.

Objectives

1. **Stock Price Prediction**: The primary goal is to predict future stock prices based on historical data.

2. **Model Comparison**: The program allows users to compare different predictive models (LSTM, GRU, Dense Neural Network) to evaluate which performs best on the given dataset.

How it Works

1. **Model Selection**:

   - **User Interaction**: The program prompts the user to select which models to use for predictions:
     1. LSTM Model
     2. GRU Model
     3. Dense Neural Network
     4. All Models

2. **Data loading and preprocessing**:

   - **Data Download**: Use below Python script to download the stock Historical Data:-
     stock_data_download.py
   - **Load Data**: The program begins by loading stock price data from an Excel file.
   - **Normalize Prices**: Using MinMaxScaler, it normalizes the stock prices to a range of [0, 1]. This step is essential for effective training of neural networks.

3. **Generate Samples**:

   - **Create Training Samples**: The program generates sequences of historical prices (features) and their corresponding next prices (labels) based on a specified window size (e.x, 100 days). This helps the model learn from past trends to predict future prices.

4. **Data Splitting**:

    - **Train-Test Split**: The dataset is divided into training and testing sets (typically 80% for training and 20% for testing). This allows the model to learn from one subset of data and evaluate its performance on another.

5. **Model Training**:

    - Based on the user’s selection, the chosen models are trained on the training dataset. Each model has a defined architecture with specific layers and hyperparameters.

6. **Evaluation**:

    - The script assesses the performance of each model using key metrics, including Mean Squared Error (MSE) and Mean Absolute Error (MAE), to gauge accuracy.
    - It identifies the model with the lowest MSE and the closest predicted stock price as the best performer, presenting the results clearly for comparison.

Models

The script effectively demonstrates how each model predicts stock prices:

- **LSTM**: Captures long-term dependencies in time series data and is effective for stock price prediction due to its memory capabilities.

- **GRU**: Offers similar capabilities as LSTM but with a more streamlined structure, making it faster. However, it performs slightly less effectively than LSTM in some case.

- **Dense Neural Network**: This model processes the input as a flat structure rather than sequences, making it less suited for time series data compared to LSTM and GRU.

Customization Options

You can customize the program in several ways:

1. **Adjusting parameters**:
    - **Window Size**: Modify the lookback_window variable to change how many past days the model should consider when making a prediction.
    - **Batch Size and Epochs**: Change the batch size and the number of epochs in the model training functions to fine tune performance.

2. **Data Sources**:

- You can change the input data by modifying the fetch_stock_dataset function to change Excel files – Use '***stock_data_download*.py**' python file to download data from Yahoo Finance.

3. **Additional Metrics**:
    - You can implement additional evaluation metrics (ex., R-squared, Root Mean Squared Error) to gain more insights into model performance.
4. **Visualization**:
    - Customize the visualization section to display additional information, such as confidence intervals or comparisons with other stocks.
5. **Cross-Validation**:
    - Implement cross-validation techniques to assess model performance more robustly and avoid overfitting.

Results\Output

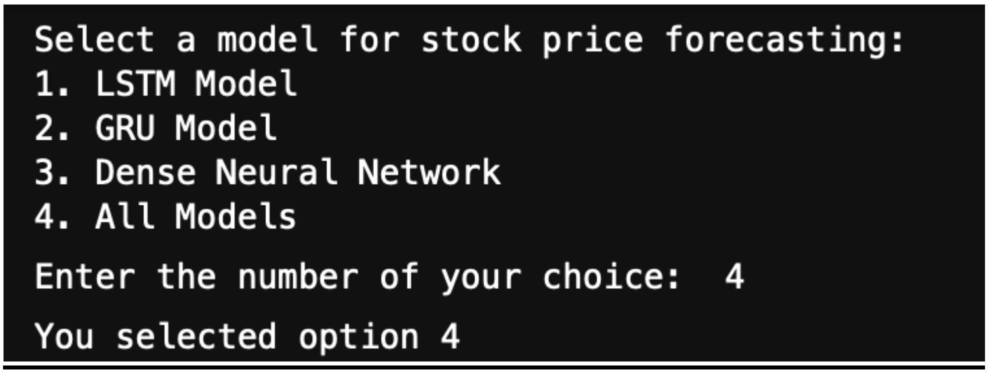
According to the output of the script:

- The **GRU Model** consistently performs the best, demonstrating the lowest MSE and MAE values and closet value predicted.

The LSTM model follows, while the Dense Neural Network generally shows lower accuracy due to the nature of time series data.

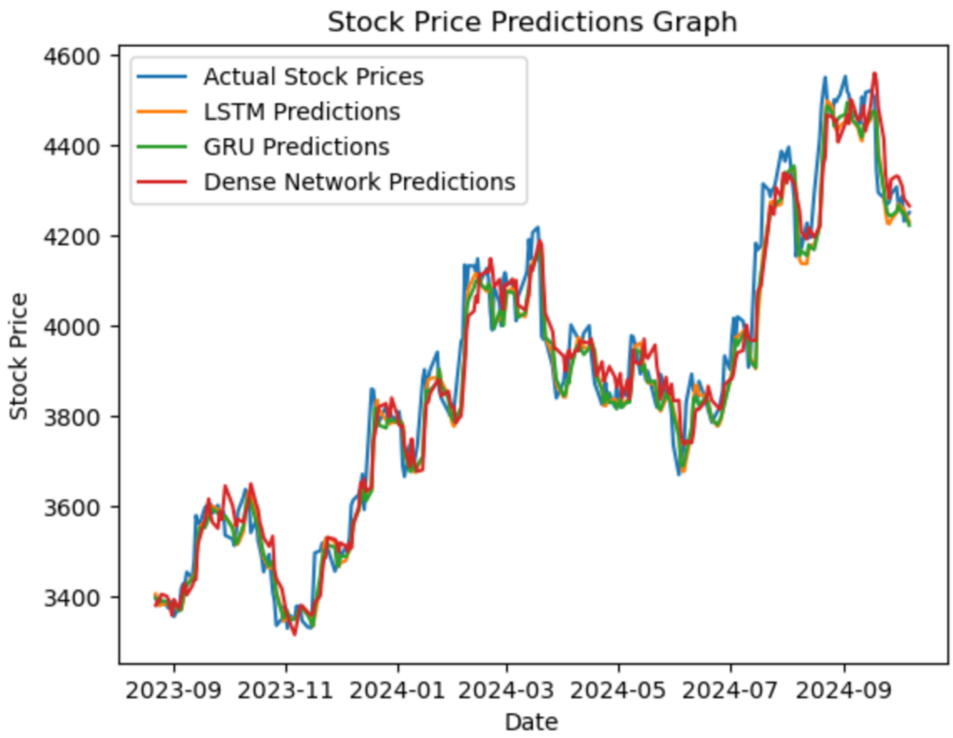
Screenshots

1. User Input screenshot:-

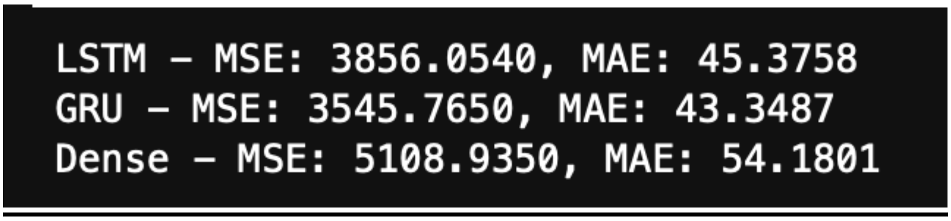


```
Select a model for stock price forecasting:
1. LSTM Model
2. GRU Model
3. Dense Neural Network
4. All Models
Enter the number of your choice:  4
You selected option 4
```

2. Stock price predication graph for all models:-

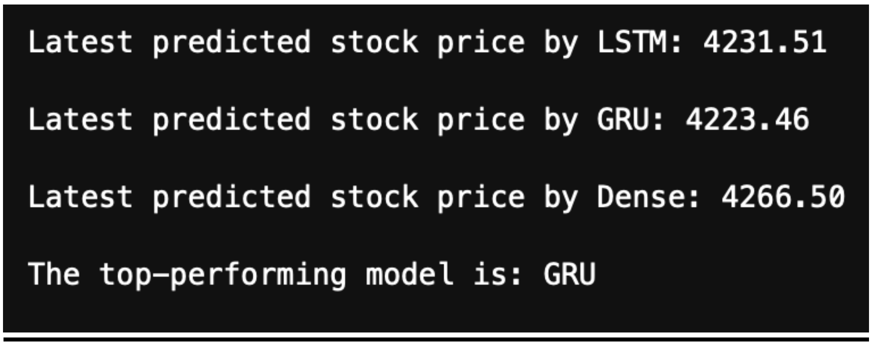


3. MSE and MAE screenshot:-



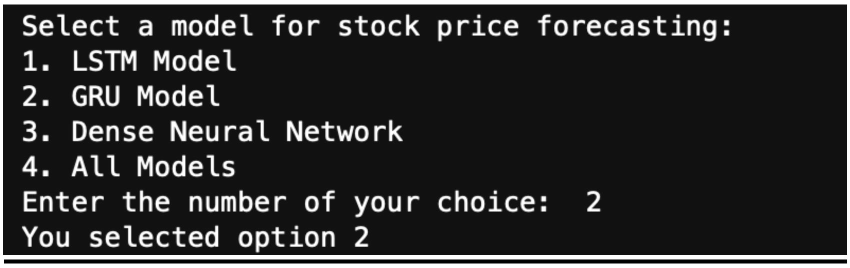
```
LSTM - MSE: 3856.0540, MAE: 45.3758
GRU - MSE: 3545.7650, MAE: 43.3487
Dense - MSE: 5108.9350, MAE: 54.1801
```

4. **Display prediction stock price and best performing model:**

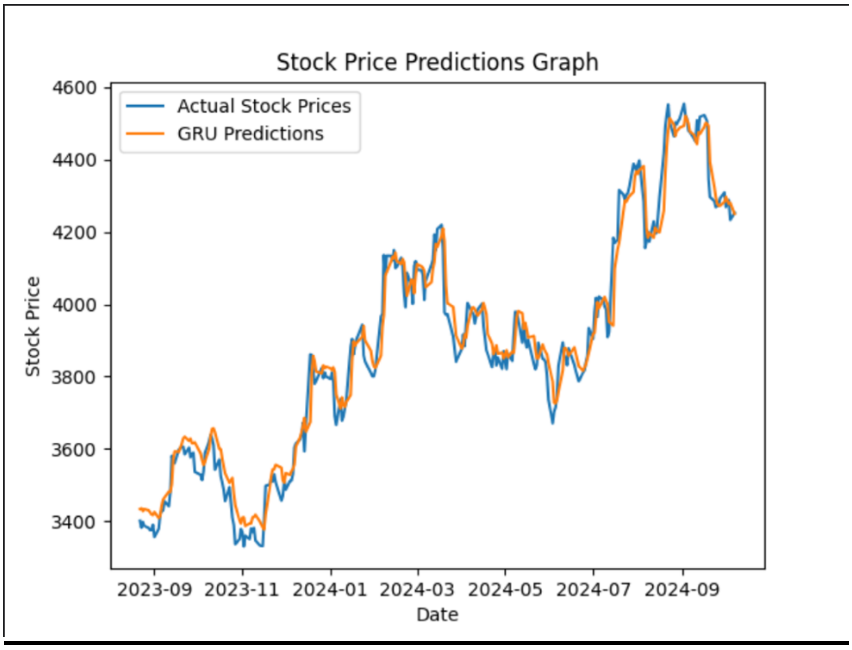


```
Latest predicted stock price by LSTM: 4231.51

Latest predicted stock price by GRU: 4223.46

Latest predicted stock price by Dense: 4266.50

The top-performing model is: GRU
```

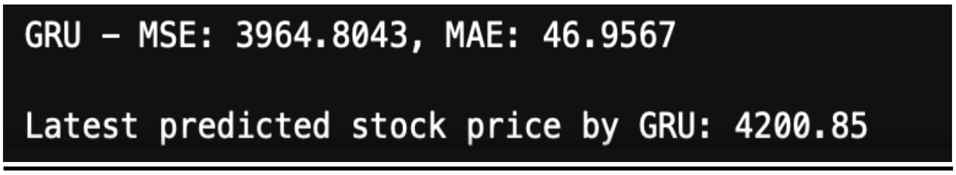
5. **Single Model Run Screenshot:**



```
Select a model for stock price forecasting:
1. LSTM Model
2. GRU Model
3. Dense Neural Network
4. All Models
Enter the number of your choice:  2
You selected option 2
```



6. **Predication GRU price output:**



```
GRU - MSE: 3964.8043, MAE: 46.9567

Latest predicted stock price by GRU: 4200.85
```