Preprocessing historical stock data, normalizing it, creating sequences for training, and implementing an LSTM model to predict future stock prices:

Step 1: Data Preprocessing

- Loaded historical stock price data from an **Excel file** while skipping unnecessary rows.
- Renamed the columns for clarity (Date, Close, High, Low, Open, Volume).
- Handled missing values by **removing any incomplete data rows**.
- Converted the "Date" column into **datetime format** and set it as the index.
- Extracted the **"Close" price** as the target variable for prediction.

Step 2: Data Normalization

- Applied **Min-Max Scaling** to transform the stock prices into a range of [0,1].
- Normalization helps in **stabilizing the training process** and preventing large value differences from affecting model performance.

Step 3: Sequence Creation for LSTM

- Created a function to generate **sequences of 50 days** of stock prices as input to predict the next day's price.
- This step is crucial since **LSTM models work best with sequential patterns**.
- Split the data into **80% training and 20% testing**.

Step 4: Model Building

The LSTM model consists of:

- **Two LSTM layers (50 units each)**: These layers help the model learn long-term dependencies in stock prices.
- **Dropout layers (20%)**: Prevents overfitting by randomly dropping connections during training.
- **Dense layers (25 units, then 1 output unit)**: Fully connected layers to refine predictions.
- **Adam optimizer** and **Mean Squared Error (MSE) loss** used for training.

Step 5: Model Training & Evaluation

- The model was trained for **50 epochs** with a **batch size of 32**.
- Training loss steadily decreased, showing that the model was learning effectively.

- Validation loss remained low, indicating that the model **generalized well** to unseen data.

Model Performance Analysis

- **Training Loss:** Started at **0.0023** and decreased to **1.1072e-04**, indicating strong learning.
- **Validation Loss:** Reduced from **9.7737e-04** to **9.8055e-04**, showing that the model is performing well on test data.
- The **architecture was effective in learning stock price patterns**, but some **overfitting** was observed due to the validation loss being slightly higher than the training loss.

Conclusion:

As part of this project, an LSTM-based stock price prediction model was developed to explore the potential of deep learning in time-series forecasting. The process involved preprocessing historical stock data, normalizing it, creating sequences for training, and implementing an LSTM model to predict future stock prices.

The model effectively captured patterns in stock prices, achieving low training and validation loss, indicating strong learning capabilities. However, slight overfitting was observed, suggesting room for further optimization. While this represents just one aspect of the broader project, the results demonstrate the effectiveness of deep learning for financial forecasting and provide a foundation for further advancements.