Forecasting Stock Prices using LSTM

1. Introduction

Stock price forecasting is a classic time series problem that can significantly benefit from deep learning techniques. Traditional statistical models like ARIMA often fall short in capturing complex temporal dependencies. This project applies a Long Short-Term Memory (LSTM) neural network to predict Tesla's (TSLA) closing stock prices, using previous 60 days' prices as input.

2. Dataset Description

The dataset used was TSLA.csv, containing historical stock prices of Tesla, Inc. The relevant column used for this task was:

Close: the daily closing price of the stock.

Only the univariate time series (Close) was used to keep the model simple and focused on temporal learning.

3. Exploratory Data Analysis (EDA)

Initial EDA included:

- Plotting closing prices over time to observe trends and volatility.
- Checking for missing values (none found).
- Confirming the stationarity was not necessary due to the model choice (LSTM handles non-stationary series).

The stock showed clear upward and downward trends over periods, making it suitable for LSTM modeling.

4. Data Preprocessing

Preprocessing steps:

- Scaling: Applied MinMaxScaler to normalize values between 0 and 1.
- Sequence Preparation: Created a sliding window of 60 days of past prices to predict the next day's price.
- Train-Test Split: 80% of data was used for training, 20% for testing.

All sequences were reshaped to fit the LSTM model input shape of [samples, timesteps, features].

5. LSTM Model Architecture

The model was built using Keras with the following architecture:

- LSTM Layer 1: 50 units, return_sequences=True
- Dropout Layer: 0.2
- LSTM Layer 2: 50 units, return_sequences=False
- Dropout Layer: 0.2
- Dense Output Layer: 1 unit

Training Details:

- Optimizer: Adam
- Loss: Mean Squared Error (MSE)
- Epochs: 20
- Batch Size: 32
- Validation Split: 10%

6. Forecasting and Evaluation

Predictions were made using the trained model on the test data. The predicted values were inverse-transformed to the original scale.

The following plot was generated to compare actual vs predicted closing prices:

[Graph Placeholder: Actual vs Predicted]

(In your actual PDF, this would be the matplotlib plot of actual vs predicted values.)

7. Results

The LSTM model was able to learn short-term patterns in the TSLA stock price data. While not perfect, the forecasted values tracked the overall trends quite well.

Key Observations:

- Model was responsive to directional changes in price.
- Slight lag observed in highly volatile transitions.

8. Conclusion & Future Work

This project demonstrates the effectiveness of LSTM networks in univariate time series forecasting.

Future Improvements:

- Incorporate additional features (Open, High, Low, Volume).
- Use Bi-LSTM or Transformer-based architectures.
- Implement Early Stopping and Learning Rate Scheduling.

- Visualize error metrics like RMSE and MAE.
- Deploy the model using Streamlit or Flask for real-time forecasting.