Stock Price Prediction Using LSTM

March 9, 2025

1 Introduction

Stock price prediction is a challenging task due to the dynamic nature of financial markets. In this project, we develop a Long Short-Term Memory (LSTM) model using TensorFlow to predict stock prices based on historical data.

2 Dataset Description

The dataset contains historical stock price data with columns including:

- **Date**: The date of the stock record
- **Open Price**: Stock's opening price
- **High Price**: Highest price of the day
- **Low Price**: Lowest price of the day
- **Close Price**: Stock's closing price
- **Volume**: Number of shares traded

Before training the model, we handled missing values and normalized the dataset to improve model performance.

3 Data Preprocessing

To prepare the dataset:

- **Missing Data Handling**:
  - Used forward fill (`ffill`) and backward fill (`bfill`) to fill NaN values.
  - Used interpolation (mean of previous and next rows) to handle remaining NaNs.
- **Feature Scaling**:

  - Applied `MinMaxScaler` to normalize stock prices between 0 and 1.
- **Data Reshaping**:
  - Converted the dataset into a time series format suitable for LSTM.

4 Model Architecture

We implemented an LSTM model using TensorFlow with the following structure:

- **Input Layer**: LSTM layers to capture time-dependent patterns
- **LSTM Layers**: Two stacked LSTM layers with 50 units each
- **Dense Layer**: Fully connected layer to output stock price predictions
- **Activation Function**: ReLU for hidden layers, Linear for output
- **Loss Function**: Mean Squared Error (MSE)
- **Optimizer**: Adam with gradient clipping

5 Training Evaluation

- **Train-Test Split**: 80% training, 20% testing.
- **Training**:
  - 50 epochs with a batch size of 32.
  - Early stopping to prevent overfitting.
- **Evaluation Metrics**:
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)

6 Results Future Improvements

- The model successfully learned stock price patterns, but performance varied depending on market volatility.
- **Future Improvements**:
  - Use additional technical indicators (e.g., Moving Averages, RSI, MACD).
  - Train on a larger dataset for better generalization.
  - Experiment with hybrid models (CNN + LSTM) for feature extraction.

7 Model
Data Saving

- **Saved Model**: `model.save("lstm_stock_model.keras")`
- **Saved Scaler**: `joblib.dump(scaler, "scaler.pkl")`
- **Load Model**: `model = load_model("lstm_stock_model.keras")`

This project demonstrates the feasibility of LSTM models in stock price prediction, providing insights for financial forecasting and investment strategies.