Stock Price Trend Prediction Using LSTM – Project Report

Introduction

In the world of financial markets, predicting stock prices is a challenging yet crucial task. With the rise of machine learning and deep learning, time-series models like Long Short-Term Memory (LSTM) networks have proven effective in modeling sequential data, such as stock prices. This project aims to leverage LSTM networks to predict future stock price trends based on historical data.

Abstract

This project presents a deep learning approach for predicting stock price trends using LSTM networks. The model is trained on historical stock data fetched via the Yahoo Finance API and makes use of technical indicators such as Moving Average (MA) and Relative Strength Index (RSI) to enhance prediction accuracy. After preprocessing and normalizing the data, the model is trained and validated, and its predictions are visualized against actual market trends. The final model is saved and can be integrated into a real-time dashboard using Streamlit.

Tools Used

- **Programming Language:** Python
- **Libraries & Frameworks:**
    - `yfinance` – For historical stock data fetching
    - `pandas`, `numpy` – Data manipulation
    - `matplotlib`, `seaborn` – Visualization
    - `scikit-learn` – Data normalization
    - `TensorFlow/Keras` – Building and training the LSTM model
    - `Streamlit` *(optional)* – For web dashboard deployment

Steps Involved in Building the Project

1. **Data Collection:**
    - Used the `yfinance` API to fetch historical stock data (e.g., AAPL from 2015–2025).
2. **Data Preprocessing:**
    - Selected the `Close` price for prediction.

   - Normalized the data using MinMaxScaler.
   - Created sequences (e.g., 60-day windows) for time-series training.
3. **Model Building:**
   - Built an LSTM model with 2 LSTM layers, dropout layers to prevent overfitting, and a final dense output layer.
4. **Model Training and Evaluation:**
   - Split the dataset into training and testing sets.
   - Trained the model for 20 epochs using MSE loss and Adam optimizer.
   - Evaluated model performance on unseen data.
5. **Visualization:**
   - Plotted actual vs. predicted stock prices to assess performance.
   - Computed and plotted 50-day Moving Average and RSI for added market insight.
6. **Saving the Model:**
   - Saved the trained model weights using `model.save_weights('lstm_stock_model.h5')` for future reuse.
7. **Optional Deployment:**
   - A basic Streamlit app was prepared for visualizing predictions and market indicators interactively.

Live Graph in Streamlit (Optional Deployment Feature)

To enhance user experience and provide real-time stock insights, a live interactive graph was added to the **Streamlit** dashboard. This feature allows users to visualize stock trends dynamically, including actual prices, predicted prices, moving averages, and RSI, all within a user-friendly web interface.

How It Works:

- The Streamlit app fetches live stock data using the `yfinance` API.
- Real-time charts are generated using `matplotlib` or `plotly`.
- The LSTM model loads saved weights and predicts the next price values based on the most recent data.

Conclusion

The project successfully demonstrates the use of LSTM neural networks in predicting stock prices based on historical trends. While the model does not guarantee exact price values (as markets are influenced by numerous unpredictable factors), it effectively captures general trends. Integrating technical indicators like MA and RSI provides additional insights, making this system potentially useful for traders and analysts. Further improvements can include tuning hyperparameters, incorporating news sentiment analysis, or expanding to multivariate input features.