# LSTM-Based Stock Price Prediction

**Introduction**

Stock market forecasting has long been a domain of intense interest for investors, analysts, and researchers due to the immense economic value tied to accurate predictions. However, traditional models often fail to adequately capture the complexity and randomness of financial time series data. With the rise of deep learning, particularly neural networks designed to process sequential data, the potential for more sophisticated prediction models has grown. This project focuses on utilizing a Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN), to predict stock prices based solely on historical closing values. The aim is not to provide investment advice or guaranteed predictions, but to explore the viability of LSTM in capturing time-based dependencies and trends in stock data.

**Abstract**

This project presents a machine learning approach to predicting stock prices using historical time series data and LSTM neural networks. LSTM is especially suitable for sequential data because it retains long-term dependencies better than traditional RNNs, making it ideal for financial forecasting tasks. In this implementation, we retrieve historical stock price data from Yahoo Finance using the yfinance library. After preprocessing the data through normalization and sequence generation, we train the LSTM model on 80% of the data and test its predictive capabilities on the remaining 20%. The model is evaluated through visual comparison between actual and predicted prices. While stock prices are influenced by numerous unpredictable external factors, the LSTM model demonstrates promising results by learning underlying trends and producing reasonably close estimates of future stock movements.

**Tools Used**

A variety of Python libraries and frameworks were employed to build and evaluate the stock prediction model. The project was implemented in Python due to its wide adoption in the data science community and the availability of robust libraries. The core tools include:

* **Pandas** and **NumPy**: Used for handling time series data, cleaning, reshaping, and numerical operations.
* **yFinance**: A powerful and easy-to-use Python wrapper that enables downloading historical stock data from Yahoo Finance directly into a DataFrame.
* **Scikit-learn**: Specifically, MinMaxScaler is used to scale stock prices between 0 and 1, which improves the training performance of neural networks.
* **TensorFlow and Keras**: Used to build and train the LSTM model. Keras simplifies the process of creating complex deep learning architectures.
* **Matplotlib**: Used to plot actual vs. predicted stock prices, allowing for visual performance evaluation.

**Steps Involved in Building the Project**

The development of this stock prediction system can be broken down into several key stages:

1. **Data Collection**: Using the yfinance.download() function, we retrieve stock data (e.g., Apple Inc. - AAPL) from 2015 to 2025. The model uses only the "Close" price, which reflects the final traded value of a stock at the end of each trading day.
2. **Data Preprocessing**: The closing prices are first normalized using the MinMaxScaler to scale values between 0 and 1. A sliding window of 60 time steps is used to create input sequences where each input sample consists of 60 consecutive prices and the label is the price immediately following the sequence. This approach helps the LSTM learn temporal dependencies.
3. **Model Building**: The architecture consists of two LSTM layers with 50 units each, separated by dropout layers to reduce overfitting. After the LSTM layers, a fully connected layer is added to generate the final prediction. The model is compiled using the Adam optimizer and trained using mean squared error (MSE) as the loss function.
4. **Model Training and Evaluation**: The model is trained on 80% of the dataset, while the remaining 20% is used for testing. After training, the model is evaluated by predicting stock prices for the test period. The predicted values are then inverse transformed back to their original scale using the scaler to enable comparison with actual prices.
5. **Visualization and Interpretation**: A line chart is plotted using Matplotlib to compare actual and predicted stock prices over the test period. Additionally, the last ten predictions are printed in tabular format to examine how closely the model tracks the real stock prices.

**Conclusion**

In conclusion, this project successfully demonstrates how LSTM networks can be used to forecast stock prices based on historical trends. The results show that while LSTM cannot account for all the complexities and external influences of the real-world stock market (such as news, politics, and market sentiment), it can effectively learn and replicate patterns in historical data. This approach lays the foundation for more sophisticated financial models that may incorporate additional features such as volume, technical indicators, or even news sentiment. The project serves as a practical introduction to time series forecasting with deep learning and illustrates how machine learning techniques can aid in financial analysis. Future improvements could involve tuning hyperparameters, using bidirectional LSTMs, or integrating external data sources to further enhance prediction accuracy.

Top of Form

Bottom of Form