Stock Market Price Prediction using Long Short-Term Memory (LSTM) Neural Networks

Abstract

The stock market is known for its volatility and complex patterns, making it challenging to predict price movements accurately. This project aims to forecast the next-day closing price of a stock using Long Short-Term Memory (LSTM) — a type of recurrent neural network capable of learning long-term dependencies in sequential data.

By utilizing historical stock price data from Yahoo Finance, this project builds a deep learning model that learns patterns from past prices and predicts future trends. The proposed model demonstrates how machine learning and data science techniques can be applied to financial forecasting, providing valuable insights for traders, investors, and researchers.

Introduction

Stock market prediction is one of the most popular applications of data science and machine learning. Traditional statistical models, such as ARIMA or linear regression, struggle to capture the non-linear and time-dependent nature of financial data.

Deep learning models, especially **LSTM networks**, are more suitable because they can remember information from previous time steps and handle complex temporal relationships.

In this project, we:

- Collected 10 years of stock data for Apple Inc. (AAPL) from Yahoo Finance.
- Processed and normalized the data.
- Trained an LSTM model to predict future stock closing prices.
- Compared actual and predicted prices to evaluate performance.

Objective

The main objectives of this project are:

1. To build a data-driven model that predicts the **next-day closing price** of a stock.
2. To understand how LSTM models can learn from sequential time-series data.
3. To visualize and compare actual vs. predicted stock prices.
4. To explore the potential and limitations of AI-based financial forecasting.

Tools and Technologies Used

Category	Tools/Technologies
Programming Language	Python
Libraries Used	yfinance, pandas, numpy, matplotlib, scikit-learn, tensorflow, keras
Data Source	Yahoo Finance
Development Environment	Jupyter Notebook / Google Colab

Dataset Description

- **Stock Symbol:** AAPL (Apple Inc.)
- **Duration:** 1st January 2015 – 1st January 2025
- **Attributes Collected:**
    - Date
    - Open Price
    - High Price
    - Low Price
    - Close Price
    - Adjusted Close
    - Volume

For modeling, only the **‘Close’** price was used, as it represents the final traded price of a day.

Methodology

The entire process can be divided into several stages:

Step 1: Data Collection

Data was downloaded using the yfinance library directly from Yahoo Finance.

Step 2: Data Preprocessing

- Extracted only the 'Close' column.
- Scaled the values between 0 and 1 using MinMaxScaler to improve model training.
- Created sequences of **60 previous days** to predict the **next day**.

Step 3: Train-Test Split

- 80% of the data was used for training and 20% for testing.
- The data was **not shuffled** to maintain time order (to prevent future information leakage).

Step 4: Model Building

A **Sequential LSTM model** was built using TensorFlow/Keras:

- Two LSTM layers with 50 neurons each.
- One Dense (fully connected) layer with 25 neurons.
- Output layer with 1 neuron for predicting the next day's price.
- Optimizer: *Adam*
- Loss function: *Mean Squared Error (MSE)*

Step 5: Model Training

The model was trained for **10 epochs** with a **batch size of 32**.

Step 6: Prediction and Evaluation

- The trained model was tested on unseen (future) data.
- Predicted values were converted back to original scale using inverse transformation.
- Model accuracy was evaluated using **Root Mean Squared Error (RMSE)**.

Step 7: Visualization

- Actual vs Predicted prices were plotted using Matplotlib.
- This visually demonstrates how close predictions are to real stock movements.

Results and Discussion

Model Performance:

- The model successfully captured the overall trend of Apple's stock price.
- Predicted values closely followed the actual price curve, especially during stable market periods.
- Minor deviations occurred during high volatility periods, which is **expected in financial time-series prediction.**

Evaluation Metric:

- Root Mean Squared Error (RMSE): *(example)* 4.52 USD *(Note: RMSE will vary slightly each time you train the model.)*

Next-Day Price Prediction Example

For the last 60 days of data, the model predicted the next day's closing price as:

Predicted Price: $202.37 (example)

Graphical Output

A plot was generated showing:

- Blue Line → Actual Stock Prices
- Orange Line → Predicted Stock Prices

The graph showed strong overlap, proving the LSTM's ability to learn temporal patterns effectively.

Advantages of LSTM in Stock Prediction

- Remembers long-term dependencies (past patterns).
- Handles non-linear relationships better than traditional models.
- Automatically learns time-based features (no need for manual feature engineering).

Limitations

- LSTM predictions are sensitive to market volatility and sudden events (e.g., news, earnings, political events).
- Model doesn’t consider external factors like company performance, sentiment, or macroeconomic indicators.
- Increasing prediction horizon (e.g., 7 days ahead) reduces accuracy.

Future Scope

- Include technical indicators like RSI, MACD, Moving Averages, etc.
- Use Transformer-based models for improved long-term understanding.
- Incorporate sentiment analysis of financial news and social media data.
- Deploy the model as a web app using Streamlit or Flask for real-time prediction.

Conclusion

This project demonstrates how LSTM neural networks can be effectively used for stock market price prediction. By training on historical data, the model learns underlying trends and produces realistic forecasts.

While perfect prediction in financial markets is impossible, LSTM-based systems can provide valuable insights and trend directions to assist in decision-making.

The project highlights the power of deep learning in time-series forecasting and forms a strong foundation for future improvements in AI-driven financial analytics.

References

- Yahoo Finance API Documentation – https://finance.yahoo.com/
- TensorFlow Documentation – https://www.tensorflow.org/
- Scikit-learn Documentation – https://scikit-learn.org/
- “Sequence Modeling with LSTM Networks” – Goodfellow et al. (Deep Learning Book)