Stock Price Trend Prediction with LSTM

Project Report

October 26, 2025

Abstract

This report details the design, implementation, and deployment of a stock price prediction system using a Long Short-Term Memory (LSTM) neural network. The objective was to forecast future stock price trends by learning from historical data. The project utilized Python and the Keras/TensorFlow library to build a stacked LSTM model. Historical stock data for Apple (AAPL) from 2015 to 2024 was sourced using the `yfinance` API. To provide the model with richer context, features were engineered to include not only the closing price but also the 20-day Moving Average (MA) and the 14-day Relative Strength Index (RSI). The complete data pipeline—from fetching and preprocessing to model training and evaluation—was developed. The final, trained model was deployed as an interactive web application using Streamlit Community Cloud, demonstrating a successful end-to-end data science project.

1. Introduction

Financial markets are notoriously complex and volatile, making stock price prediction one of the most challenging tasks in data science. Traditional statistical methods often fall short as they struggle to capture the non-linear, sequential, and long-term dependencies inherent in stock market data. This project confronts this challenge by employing a Long Short-Term Memory (LSTM) network, a specialized type of recurrent neural network (RNN) explicitly designed to recognize patterns in sequential data over long time horizons.

The primary goal of this project was not to predict an exact future price, but to forecast the *trend* of a stock's movement. The scope extended beyond simply building a model; it encompassed the entire data science lifecycle, including:

- Sourcing and cleaning real-world financial data.
- Engineering meaningful technical indicators as model features.
- Building, training, and evaluating a robust deep learning model.
- Deploying the final model as an accessible, interactive web application for real-world use.

2. Tools Used

The project was built using a combination of powerful open-source Python libraries, each chosen for a specific task in the data science pipeline.

Tool / Library	Purpose in Project
Python 3	The core programming language for the entire project.
TensorFlow & Keras	For building, training, and evaluating the LSTM neural network.
Streamlit	To build and deploy the interactive web dashboard.
Pandas	For data loading, manipulation, and cleaning.
yfinance	To fetch historical stock data directly from the Yahoo Finance API.
TA-Lib	For calculating technical analysis indicators (RSI, MA).
Scikit-learn	Used for data normalization with `MinMaxScaler`.
GitHub	For version control and as the bridge to deploy on Streamlit Cloud.

Caption: Key technologies and their roles.

3. Steps Involved in Building the Project

The project was executed in a systematic, end-to-end process, broken down into the following key stages:

1. **Data Collection & Feature Engineering:** Historical stock data for Apple (AAPL) from 2015 to 2024 was downloaded using the `yfinance` library. To enrich the dataset, two critical technical indicators were calculated and added as features: the 14-day Relative Strength Index (RSI) and the 20-day Simple Moving Average (MA).
2. **Data Preprocessing:** The initial dataset contained `NaN` (blank) values created by the indicator calculations. These rows were dropped. The resulting features (Close, MA_20, RSI) were then normalized using `MinMaxScaler` to scale all values between 0 and 1. This step is critical for the stable training of a neural network.
3. **Sequence Generation:** The data was transformed from a daily list into overlapping sequences. A "look_back" period of 60 days was chosen, meaning the model was trained to use the features from the past 60 days (X) to predict the closing price of the 61st day (y).
4. **Model Architecture:** A stacked LSTM model was constructed using Keras. It consisted of two `LSTM` layers with 50 neurons each, followed by `Dropout` layers to prevent overfitting. A final `Dense` layer with a single neuron was used to output the singular, predicted price.
5. **Model Training & Evaluation:** The sequential data was split into an 80% training set and a 20% test set. The model was trained for 25 epochs. Predictions were made on the test set and then inverse-transformed back into dollar values to be plotted against the actual prices for visual evaluation.
6. **Deployment:** A Streamlit application script (`app.py`) was written to load the saved model (`.h5`) and scaler. A (`requirements.txt)` file was created to list all dependencies. The project was pushed to a GitHub repository, which was then linked to Streamlit Community Cloud for automated, continuous deployment.

4. Conclusion

This project successfully demonstrated the complete lifecycle of a deep learning project, from data acquisition to a publicly deployed web application. The resulting LSTM model is capable of tracking the general trend of the stock price, validating its ability to learn from historical time-series data.

The model's predictions, while not perfectly accurate, capture the directional movements, though often with a slight lag. This is a common and expected characteristic of models trained on price action alone. The project serves as a robust foundation for future work, which could include adding more features (e.g., trading volume, Bollinger Bands), performing hyperparameter tuning, or integrating sentiment analysis from news headlines to provide the model with external context.