Stock Sight

Machine Learning for Apple Stock Price Prediction

A PROJECT REPORT

Submitted by

Suraj Gangwar Registration Number: 25BCE11388

[College Name] *Department of Computer Science and Engineering*

GitHub Repository: Stock-Price-Prediction-Using-LSTM-Neural-Networks

1. Abstract

Stock market prediction is a classic problem in finance and computer science due to the volatile and non-linear nature of financial time-series data. **StockSight** is a machine learning pipeline developed to predict the daily closing price of Apple Inc. (AAPL) stock. By utilizing a Random Forest Regressor and historical market data (Open, High, Low) retrieved via the yfinance API, the project demonstrates how ensemble learning techniques can estimate future stock trends with reasonable accuracy. The model is evaluated using Root Mean Squared Error (RMSE) and R^2 score, and results are visualized to compare predicted values against actual market performance.

2. Introduction

2.1 Background

The ability to predict stock prices can provide significant value to investors and analysts. While stock prices are influenced by a myriad of external factors—including macroeconomic indicators, news sentiment, and global events—historical price action (technical analysis) remains a fundamental component of predictive modeling.

2.2 Problem Statement

Manual analysis of stock charts is time-consuming and prone to human bias. The challenge is to create an automated system that can ingest historical data and generate quantitative predictions for stock closing prices, providing a baseline for decision-making.

2.3 Objectives

- To automate the retrieval of financial data using Python APIs.
- To preprocess and clean time-series data for machine learning tasks.
- To train a regression model (Random Forest) to predict the 'Close' price based on daily 'Open', 'High', and 'Low' prices.
- To evaluate the model's accuracy and visualize the results for intuitive analysis.

3. Methodology

3.1 Data Acquisition

The project utilizes the yfinance library to fetch real-time and historical market data from Yahoo Finance.

- **Target Asset:** Apple Inc. (Ticker: AAPL)
- **Time Period:** Past 5 years.
- **Features Extracted:** Date, Open, High, Low, Close, Volume.

3.2 Data Preprocessing

- **Cleaning:** The raw data is checked for missing values (NaN).
- **Feature Selection:** The model uses Open, High, and Low prices as input features (X) to predict the Close price (y).
- **Train-Test Split:** Since stock data is a time series, a random split is inappropriate. The data is split **chronologically**:
    - **Training Set:** First 80% of the data (used to teach the model).
    - **Testing Set:** Last 20% of the data (used to validate the model).

3.3 Model Selection: Random Forest Regressor

A **Random Forest Regressor** was chosen for this task. It is an ensemble learning method that constructs a multitude of decision trees at training time.

- **Why Random Forest?** It handles non-linear relationships better than simple linear regression and is less prone to overfitting than individual decision trees.
- **Hyperparameters:** n_estimators=100 (The model builds 100 decision trees).

4. Implementation Details

The project is implemented in **Python 3.8+** using the following technology stack:

- **Pandas:** For data manipulation and DataFrame management.
- **NumPy:** For numerical operations and array handling.
- **Scikit-Learn:** For model creation, training, and evaluation metrics.
- **Matplotlib:** For plotting the comparison graphs.
- **yfinance:** For data fetching.

Core Algorithm Steps

1. **Fetch:** Download 5 years of AAPL history.
2. **Split:** Divide into chronological train/test sets.
3. **Train:** Fit the Random Forest model on the training features.
4. **Predict:** Generate predictions for the test set.
5. **Evaluate:** Calculate error metrics.
6. **Visualize:** Plot the results and save to prediction_chart.png.

5. Results and Analysis

5.1 Evaluation Metrics

The model's performance is quantified using two primary metrics:

1. **Root Mean Squared Error (RMSE):** Measures the average magnitude of the error. A lower RMSE indicates a better fit.
2. R^2 **Score (Coefficient of Determination):** Represents the proportion of variance in the dependent variable that is predictable from the independent variables. An R^2 score close to 1.0 indicates high accuracy.

5.2 Visualization

A generated graph (prediction_chart.png) displays three key elements:

1. **Training Data (Green):** The historical data the model learned from.
2. **Actual Test Data (Blue):** The real closing prices for the test period.
3. **Predicted Data (Red, Dashed):** The model's predictions overlaid on the actual prices.

Observation: The visualization typically shows that the Random Forest model closely tracks the trend of the actual stock price, capturing major dips and rises effectively.

6. Conclusion and Future Scope

6.1 Conclusion

StockSight successfully demonstrates the application of supervised machine learning in finance. The Random Forest Regressor proved effective in predicting the closing price of AAPL based on daily trading ranges (Open, High, Low). The automated pipeline significantly reduces the effort required for data gathering and initial technical analysis.

6.2 Future Scope

To further enhance the project, the following improvements are proposed:

- **Feature Engineering:** Incorporating technical indicators like Moving Averages (SMA/EMA), RSI, or MACD.
- **Sentiment Analysis:** Integrating news headlines or social media sentiment as input features.
- **Deep Learning:** Experimenting with LSTM (Long Short-Term Memory) networks, which are specifically designed for sequence prediction problems.
- **Multi-Stock Support:** Expanding the tool to compare multiple tickers simultaneously.

7. References

1. Scikit-Learn Documentation: https://scikit-learn.org/
2. Yahoo Finance API (yfinance): https://pypi.org/project/yfinance/
3. Pandas Documentation: https://pandas.pydata.org/