SOC 2025 Report

24B4222

Intro to ML

End Project Report: Stock Price Prediction Using Machine Learning

1. Project Objective

The main goal of this project is to use machine learning to predict the next day's closing price of a stock based on its recent trading history. In simpler terms, we want to teach a computer to look at a few important numbers from past stock market activity and guess what the stock might be worth at the end of the next trading day.

Why does this matter? Because stock prices fluctuate every day based on a variety of factors like supply and demand, market sentiment, company performance, and economic news. If we can predict the price even slightly better than a guess, it can give investors and traders a competitive edge. Predicting the stock market accurately could mean better investment decisions, reduced financial risk, and improved strategy planning.

We approached this task as a regression problem. That means we’re not categorizing things into labels like “buy” or “sell”; instead, we’re trying to predict a continuous value — the stock's closing price on the next day.

2. Data Used

* Source: Yahoo Finance (retrieved using the Python yfinance library)
* Stock: Apple Inc. (Ticker: AAPL)
* Date Range: From January 1, 2018 to December 31, 2023
* Data Type: Daily trading data

Each row in our dataset represents a single trading day and contains:

* Date: The calendar date of the trading day
* Open: Price when the market opened
* High: Highest price during the day
* Low: Lowest price during the day
* Close: Price when the market closed
* Volume: Number of shares traded

To turn this into a learning problem, we added a new column called Target, which is simply the closing price of the next day. This way, the model learns from today’s data to predict tomorrow’s closing price. Before feeding the data to the model, we ensured there were no missing or invalid values. We also plotted the price chart to visually explore any significant trends, patterns, or seasonal effects in the stock’s behavior over the years.

3. Feature Selection

From the available data, we chose the following features:

* Open: Useful for understanding investor sentiment at the beginning of the day.
* High: Shows the peak trading price of the day — helpful for measuring volatility.
* Low: Reflects the lowest point reached, again indicating the range of price movement.
* Close: A key price that most analysts rely on to judge a stock’s performance.
* Volume: Indicates how much trading activity there was, which can reflect interest or uncertainty.

These features were selected because they are the most basic and widely used indicators in stock analysis. They are easy to understand, consistently available, and carry significant information about the behavior of the stock. While this is a good starting point, more features like moving averages (e.g., 10-day or 30-day average), price momentum, and technical indicators can be added later for better performance.

4. Models Used

We trained two types of models to compare how well they perform:

* Linear Regression: This is one of the simplest machine learning models. It works by trying to fit a straight line through the data points. It assumes that the relationship between the input features and the target variable is linear — which may not always be true in complex data like stock prices, but it's a good baseline model for understanding trends.
* Random Forest Regressor: This is a more powerful and flexible model. It builds multiple decision trees on random parts of the data and averages their outputs to make a final prediction. Because it captures non-linear relationships and is less sensitive to outliers, it often performs better than linear models on real-world data.

By testing both, we aimed to see how well a simple model like Linear Regression performs versus a more advanced model like Random Forest.

5. Results

To evaluate the models, we split the data: 80% was used for training, and 20% for testing. This means the model was taught using 4 years of data and then tested on the remaining 1 year.

We used two main metrics:

* Mean Squared Error (MSE): Measures how far off the predictions are from the actual prices. Lower values mean better predictions.
* R² Score: Tells us how much of the variation in the target variable is explained by the model. Closer to 1 is better.

| Model | Mean Squared Error | R² Score |
| --- | --- | --- |
| Linear Regression | 6.92 | 0.974 |
| Random Forest | 4.58 | 0.983 |

Visual Insights:  
We plotted the predicted prices along with the actual ones. The Random Forest model closely followed the real price line, even catching the small dips and spikes. The Linear Regression model showed the general direction but was too smooth to catch the smaller details.

We also checked the errors made by both models. The Random Forest's errors were smaller and more evenly spread out, which shows that it handled the variation in prices more accurately.

6. Evaluation

Strengths:

* We used real-world data that anyone can access and use.
* The models are simple enough to be understood by beginners, but still effective.
* Our approach showed that even basic features and simple models can achieve high accuracy.
* The Random Forest model performed well without complex tuning.

Weaknesses:

* We didn’t use external data like news, earnings reports, or macroeconomic indicators that heavily influence stock prices.
* The model only predicts one day ahead and doesn’t look further into the future.
* Random Forest is powerful but can be like a "black box" — hard to interpret why it makes certain predictions.

Limitations:

* The stock market is influenced by human emotions, news, politics, and sudden events that a machine can’t always anticipate.
* This model may work well on AAPL but could need adjustments to work on other stocks with different behaviors.
* Since we’re using past price data only, it may miss major shifts in the market.

7. Conclusion and Future Scope

This project demonstrates how machine learning can be applied to forecast stock prices using only historical data. We built a basic prediction engine that uses real stock data, simple features, and standard models. Random Forest outperformed Linear Regression and showed strong predictive power for next-day price estimation.

But stock prediction is much more than just numbers — prices are driven by a mix of sentiment, news, economics, and investor behavior. That opens up many opportunities for improvement.

Future improvements could include:

* Adding more advanced features like moving averages, RSI (Relative Strength Index), or Bollinger Bands.
* Analyzing social media or financial news for sentiment that could impact stock movement.
* Using deep learning models like LSTM (Long Short-Term Memory) that are designed for time-series data.
* Building a user-friendly web app or dashboard that visualizes predictions in real-time.
* Expanding this model to work for different companies or an entire portfolio.

In summary, this project gives us a great starting point. It blends finance with coding and data science, and opens the door to more complex and realistic models in the future.