

Nvidia Stock Price Predictor

I built the Nvidia Stock Price Predictor to forecast Nvidia's future stock prices based on historical data and technical analysis. The model analyses past stock prices, 50-day and 200-day moving averages, and key momentum indicators such as the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) to predict the stock's future movements. The goal was to develop a data-driven approach to understanding potential trends in Nvidia's stock price, offering insights into this dynamic and volatile market.

My motivation for this project stemmed from my personal interest in the stock market. As a frequent trader, I have achieved profits through my own technical analysis, which inspired me to explore how a machine learning model could replicate this process. The data for this model was imported directly from Yahoo Finance using their Python package.

For the model, I chose to use a Random Forest algorithm due to its ability to handle complex relationships within the data and provide robust, reliable predictions. The dataset was multivariate, with multiple independent variables used to predict the target variable, which was the closing stock price.

To evaluate the model's accuracy, I focused on three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). Initially, the model produced highly inaccurate predictions, prompting me to experiment with the number of trees in the Random Forest. I gradually increased the number of trees in increments of 50, starting from 100, with the idea that more trees would improve accuracy by reducing variance and leading to more reliable predictions through averaging.

The final results of the model were:

- Mean Absolute Error: 6.17
- Mean Squared Error: 46.55
- R-squared: -0.82

While these results suggest the model's predictions were still inaccurate, with a negative R-squared indicating that it is not capturing patterns in the data effectively, they provide valuable insight into areas for improvement. Predicting the stock price of a volatile asset like Nvidia is inherently challenging, as factors beyond just technical indicators influence price movements. However, in the future, the model's accuracy could be enhanced through better feature engineering, hyperparameter tuning, or by experimenting with other machine learning models better suited for time-series data, such as ARIMA or LSTM.

I had a lot of fun working on this project and it greatly improved my understanding of the uses of machine learning in the real world. This has me to want to increase my understanding of machine learning.

Setting up the environment

The project relies on several essential Python libraries: yfinance for fetching stock market data, pandas and numpy for data manipulation, scikit-learn for building and evaluating the machine learning model, and matplotlib for visualizing results. These libraries can be installed using the following command:

```
pip install yfinance pandas numpy scikit-learn matplotlib
```

After installing the libraries, ensure your system has a stable internet connection to allow the script to fetch historical Nvidia stock data from Yahoo Finance. Once all dependencies are installed, navigate to your project folder in the terminal and run the script with: