Predictions on the Stock Market

By: Joshua Gibson, Jesse Brents, Andrew Clifft, and Morgan Foge

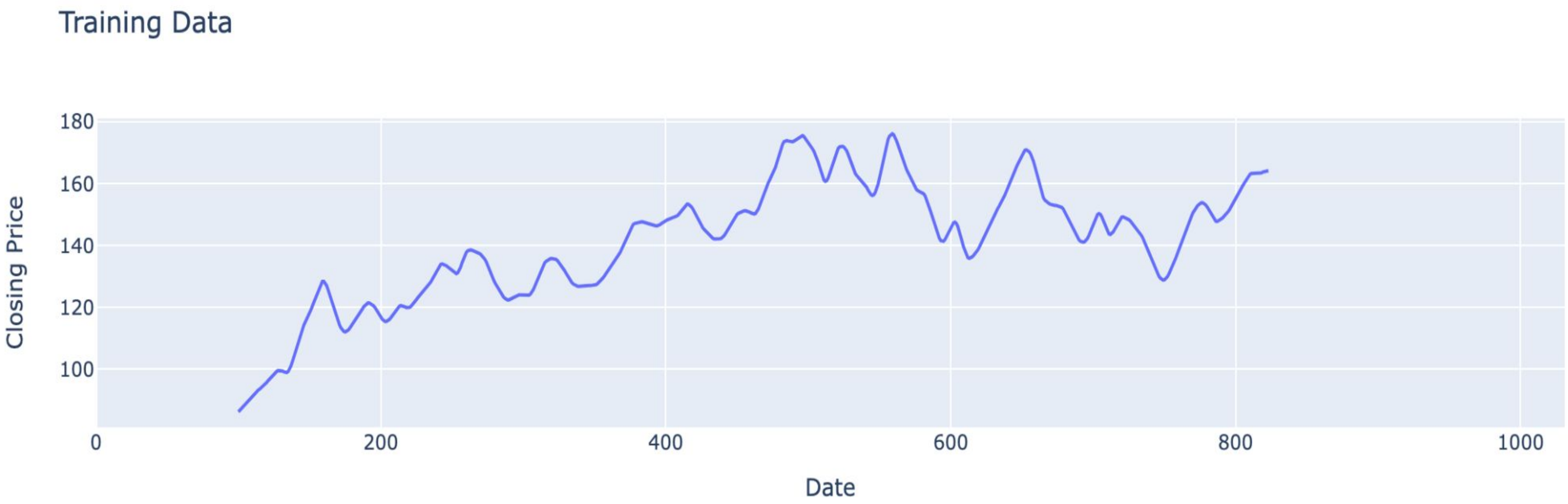
Starting Out

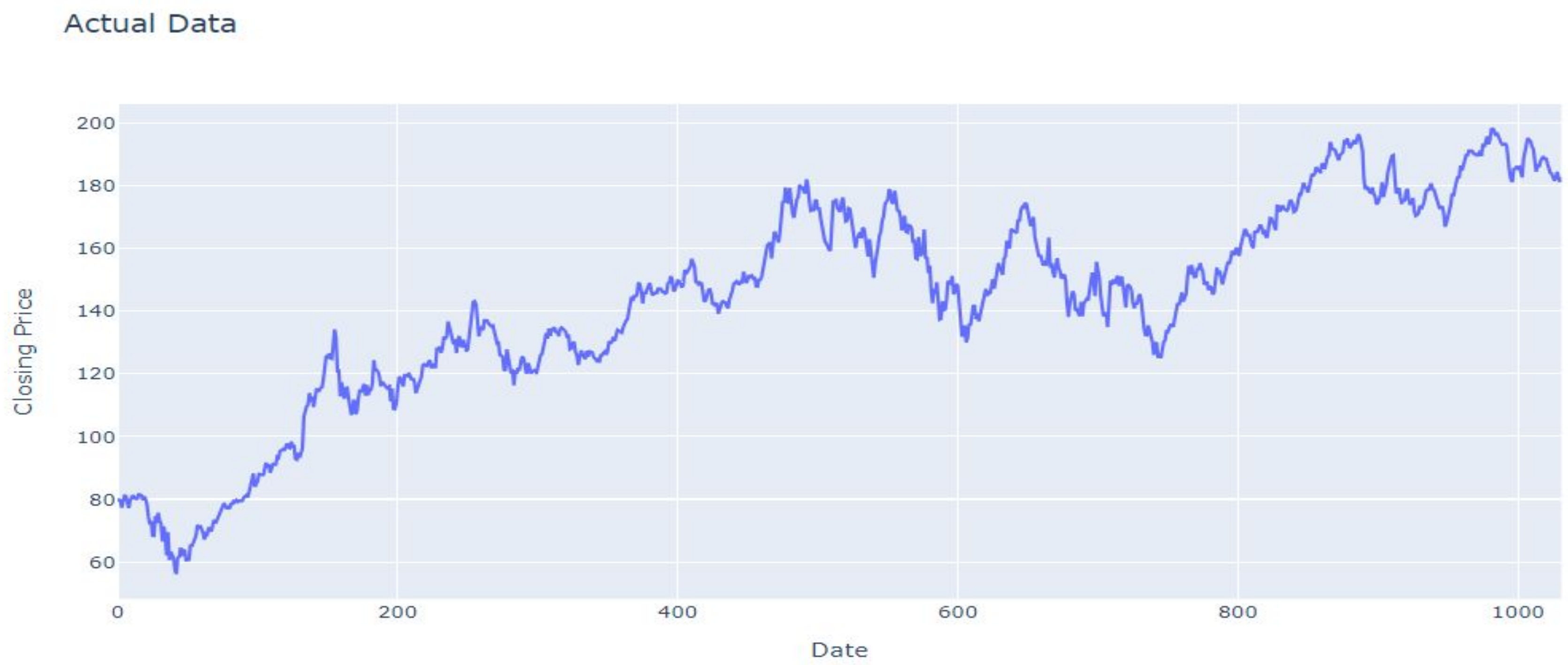
We wanted to use Machine Learning to predict the closing stock prices for stocks because it would offer an opportunity for us to delve into the exciting combination of both finance and machine learning, while potentially gaining insights into the dynamics of financial markets.

Dataset found?

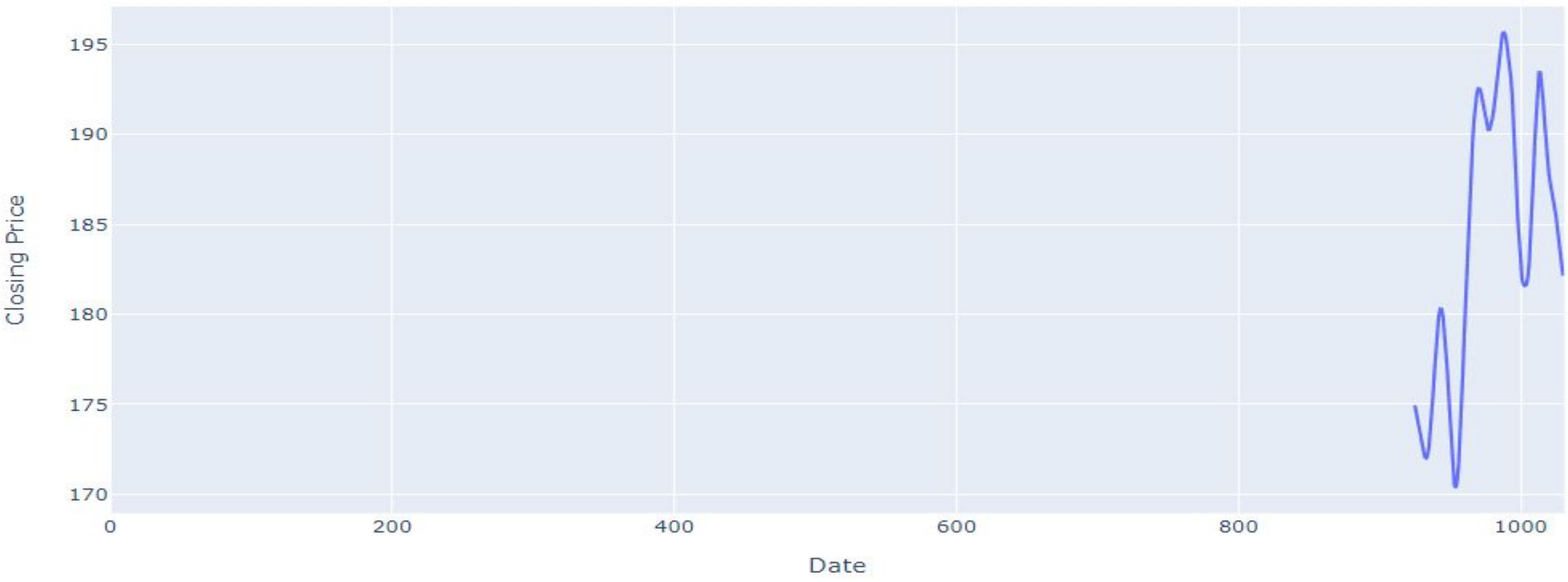
On Kaggle we found a Stock Dataset for the S&P 500 that the creator updated and kept clean. We used 4 years of data to help train our Machine so that it would learn and capture trends, so it would be able to accurately predict the outcome of the stock prices.

Visuals of Testing and Training

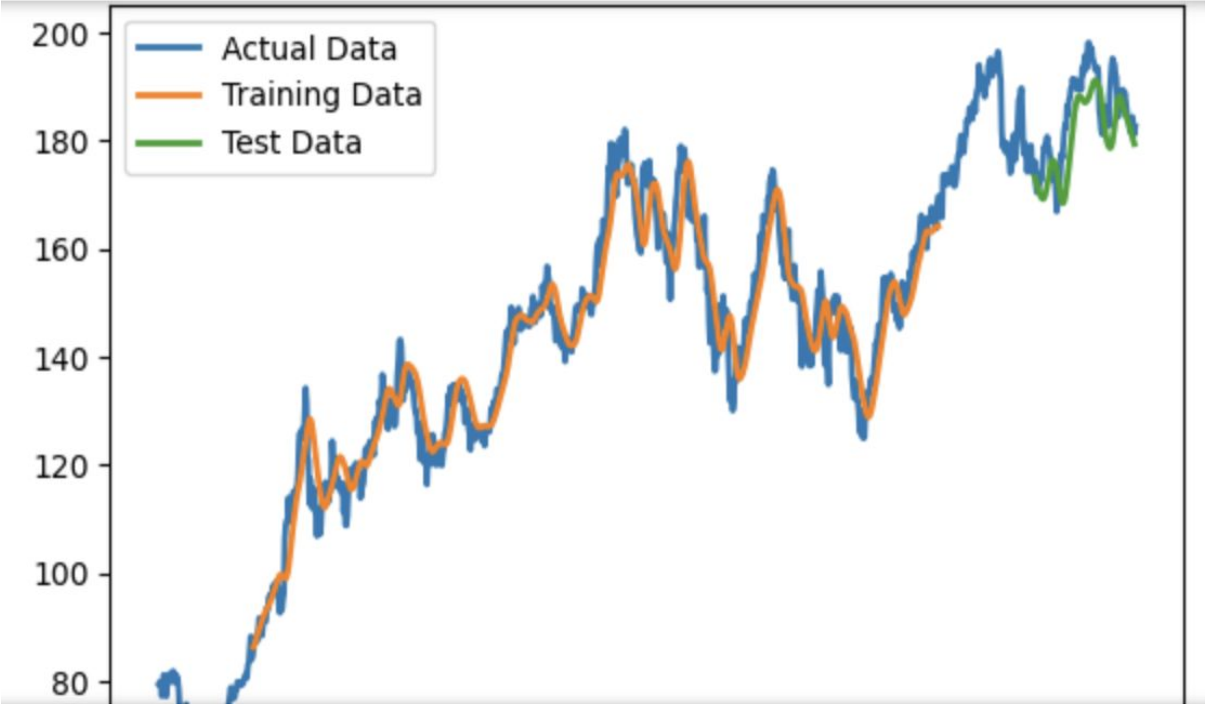



Actual Data
200
180
160
140
120
100
80
60
Closing Price
0
200
400
600
800
1000
Date

Test Data


195
190
185
180
175
170
Closing Price
0
200
400
600
800
1000
Date

Training



When we trained and created our Machine, we also created a graph to see the data and keep track of the test data, training data, and actual data to know if it was successful.

Issues?

We ran into issues after creating our machine and training that we did not foresee happening. Our model was not giving accurate readouts and would become stagnant after a few readouts. After some research and conversations, we reached the conclusion that the more days out you try to predict, the less accurate your prediction will become. This is because stocks have a number of incalculable variables from day to day.

Debugging and figuring out the issue

We started looking at all sorts of possible reasons and checking our R^2 score. After some research and a conversation with Ahmad, we reached the conclusion that the farther from the current day, the less accurate your prediction will become due to multiple factors like social media, industry news, and company news.

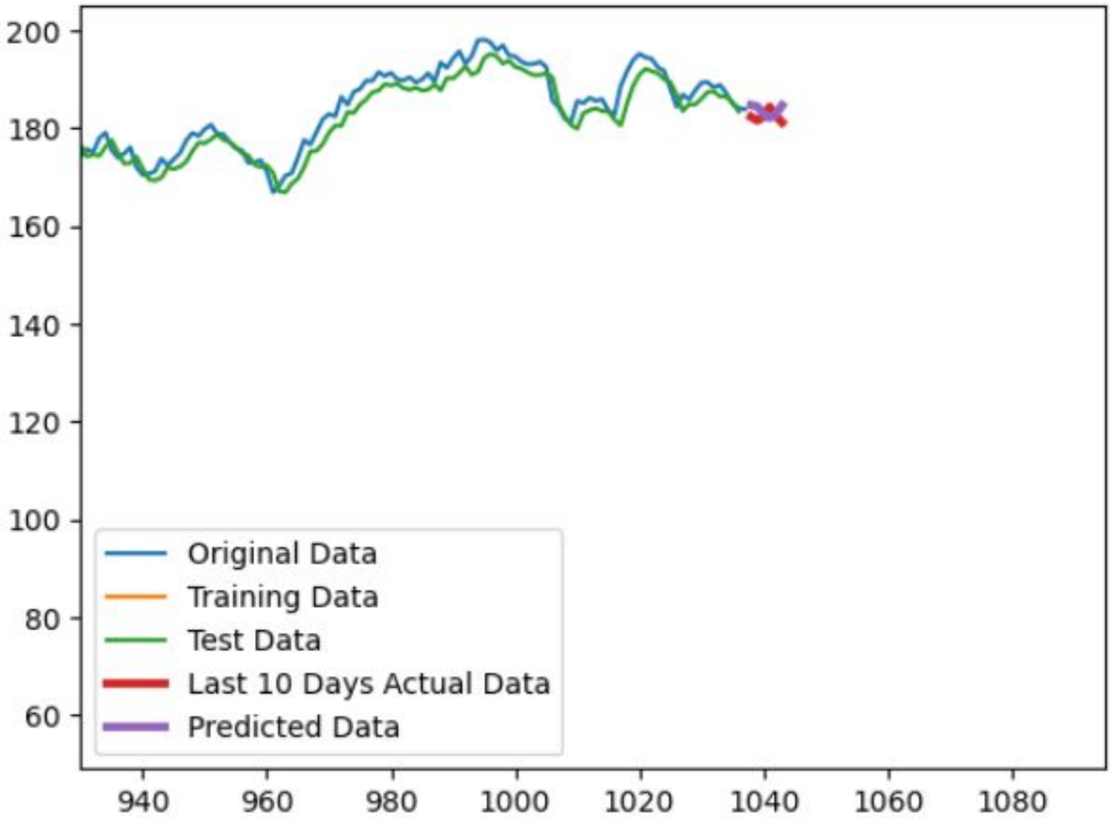
```
Mean Absolute Error (MAE): 0.01651371386995071
Mean Squared Error (MSE): 0.0004658930621188948
R^2 Score: 0.9797903625873398
```

Pivoting our plan

Finding out this information put us in a weird predicament; we were unsure how to move forward with our project. However, because our data was being updated daily, we thought that we could take the updated data in the dataset and create a model that kept track and compared our models predictions for the following day. By doing this, we could feed actual daily data to make the next day's predictions. This led to a better way for us to track if the data trend is being followed correctly.

Failure?

After redirecting our goal, our graph was still unable to accurately guess the upcoming trends. In this instance, it was predicting almost the exact opposite of the actual closing price.



Using the YFinance library

We tried using the “yfinance” library. This library contained formulas to add moving averages, strength index, and a couple of ratios.This allowed us to add more features for the model to go off of, and enabled the machine to get a more accurate 1-day prediction for what would be yesterday’s closing price.

Stock Technical Indicators for LTSM Prediction Model

We added a loop for "calculate_features" that calls the data for the stock symbol and computes various technical indicators and OHLC ratios.

Each dataset within the 'stocks_data' dictionary is augmented with the calculated features.

In a LSTM (Long Short-Term Memory) stock prediction model for Keras, incorporating relevant technical indicators can enhance the model's ability to capture patterns and trends in stock price data. The three commonly used technical indicators used in our final model include the following: moving average, relative strength index (RSI), and OHLC (Open, High, Low, Close) Ratios.

In [8]:

```python
# Calculate technical indicators and OHLC ratios for stock
def calculate_features(data):
    # Moving Average
    data['MA_10'] = data['Close'].rolling(window=10).mean()
    # Exponential Moving Average
    data['EMA_10'] = data['Close'].ewm(span=10, adjust=False).mean()

    # Relative Strength Index (RSI)
    delta = data['Close'].diff()
    gain = (delta.where(delta > 0, 0)).rolling(window=14).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=14).mean()
    rs = gain / loss
    data['RSI'] = 100 - (100 / (1 + rs))

    # OHLC Ratios
    data['HLC_Ratio'] = (data['High'] - data['Low']) / data['Close']
    data['OC_Ratio'] = (data['Open'] - data['Close']) / data['Close']

    # Drop NaN values generated by moving averages and RSI
    data.dropna(inplace=True)
    return data
```

```python
# Apply feature engineering to stocks
for symbol in stocks_data:
    stocks_data[symbol] = calculate_features(stocks_data[symbol])
```

Out[9]:

Date	Open	High	Low	Close	Adj Close	Volume	MA_10	EMA_10	RSI	HLC_Ratio	OC_Ratio
2020-01-22	79.644997	79.997498	79.327499	79.425003	77.279770	101832400	78.309250	78.255756	70.868434	0.008436	0.002770
2020-01-23	79.480003	79.889999	78.912498	79.807503	77.651939	104472000	78.710250	78.537892	71.902570	0.012248	-0.004104
2020-01-24	80.062500	80.832497	79.379997	79.577499	77.428146	146537600	78.927250	78.726911	75.401481	0.018253	0.006095
2020-01-27	77.514999	77.942497	76.220001	77.237503	75.151337	161940000	78.892751	78.456110	59.513444	0.022301	0.003593
2020-01-28	78.150002	79.599998	78.047501	79.422501	77.277336	162234000	78.911001	78.631817	67.412481	0.019547	-0.016022
...	...	...	...	...	...	...	...	...	...	...	...
2024-02-21	181.940002	182.889999	180.660004	182.320007	182.320007	41529700	185.297000	184.512015	43.772500	0.012231	-0.002084
2024-02-22	183.479996	184.960007	182.460007	184.369995	184.369995	52292200	184.792999	184.486193	42.357254	0.013560	-0.004827
2024-02-23	185.009995	185.039993	182.229996	182.520004	182.520004	45074500	184.212999	184.128704	40.280199	0.015396	0.013642
2024-02-26	182.240005	182.759995	180.649994	181.160004	181.160004	40867400	183.443999	183.588940	30.432210	0.011647	0.005962

Why Use a Moving Average?

A moving average helps cut down the amount of noise on a price chart. Look at the direction of the moving average to get a basic idea of which way the price is moving. If it is angled up, the price is moving up (or was recently) overall; angled down, and the price is moving down overall; moving sideways, and the price is likely in a range.

In the financial markets, noise can include small price corrections in the market as well as price fluctuations–called volatility–that distorts the overall trend. However, market noise can make it challenging for investors to discern what's driving the trend and whether a trend is changing or merely experiencing short-term volatility.

Moving averages work quite well in strong trending conditions but poorly in choppy or ranging conditions. Adjusting the time frame can remedy this problem temporarily, though at some point, these issues are likely to occur regardless of the time frame chosen for the moving average(s).

A moving average simplifies price data by smoothing it out and creating one flowing line. This makes seeing the trend easier. Exponential moving averages react quicker to price changes than simple moving averages.

A five-day simple moving average (SMA) adds up the five most recent daily closing prices and divides the figure by five to create a new average each day. Each average is connected to the next, creating the singular flowing line.

The one used in our model is a popular type of moving average called the exponential moving average (EMA). The calculation is more complex, as it applies more weighting to the most recent prices. Since our data depends on more recent movement, we used this metric.

If you plot a 50-day SMA and a 50-day EMA on the same chart, you'll notice that the EMA reacts more quickly to price changes than the SMA does, due to the additional weighting on recent price data.

What Is the Relative Strength Index (RSI)?

The relative strength index (RSI) is a momentum indicator used in technical analysis. RSI measures the speed and magnitude of a security's recent price changes to evaluate overvalued or undervalued conditions in the price of that security.

$$RSI = 100 - \left[\frac{100}{1 + \frac{n_{up}}{n_{down}}} \right]$$

where:

n_{up} = average of n-day up closes

n_{down} = average of n-day down closes

(most analysts use 9 - 15 day RSI)

The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100.

Traditional interpretation and usage of the RSI are that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

Disadvantages of Momentum Trading

Just like any other trading style, there are risks that come with momentum trading. By using this technique, you should know that you are trading on the backs of other people in the market, and price trends are never guaranteed. And always be prepared for unexpected reversals or corrections that take place. This can happen because of unexpected news or changes in investor sentiment in the market.

Simply put, the RSI forecasts sooner than almost anything else an upcoming reversal of a trend, either up or down.

Understanding an OHLC Chart and How to Interpret It

An OHLC chart is a type of bar chart that shows open, high, low, and closing prices for each period. OHLC charts are useful since they show the four major data points over a period, with the closing price being considered the most important by many traders.

When the close is above the open, the bar is often colored black. When the close is below the open the bar is often colored red. OHLC and candlestick charts show the same amount of information, but they show it in a slightly different way. While OHLC charts show the open and close via left and right facing horizontal lines, candlesticks show the open and close via a real body.

```
Mean Absolute Error (MAE): 0.03078462652590841
Mean Squared Error (MSE): 0.0015270573601939S3
R^2 Score: 0.925594527603979
```

Official Machine Learning Prediction

Below is the machine’s prediction for 2/28/24, and the actual closing price via Google:

```
           Close  Predictions
1031  182.630005   179.352173
Predicted Closing Price for February 28th, 2024:[[179.35217]]
```

Date	Open	High	Low	Close
Feb 28, 2024	182.51	183.12	180.13	181.42

Incalculable Factors For Yesterday’s Prediction

Below are a couple of the news stories from 2/27 and early morning 2/28 regarding Apple that had an impact in yesterday’s closing price:

Bloomberg.com

Apple Cancels Its Electric Car in Favor of AI, Vision Pro

17 hours ago

Yahoo Finance

Apple Stock Could Notch New Record High

1 day ago

Conclusion

In the end we tried many different libraries and used a lot of resources and research to look into the problems that we were having. We know that our Machine does not give a 100% accuracy readout, but it does give a fairly good estimate. We think that if we had more access to more features, we could make our predictions more and more accurate for each upcoming day.

Works Cited

Data: https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks

Reference Material:

- https://data-flair.training/blogs/stock-price-prediction-machine-learning-project-in-python/amp/
- https://theaiquant.medium.com/forecasting-stock-returns-with-deep-learning-a-technical-approach-caccee231051

AI Help at times from ChatGPT. :)