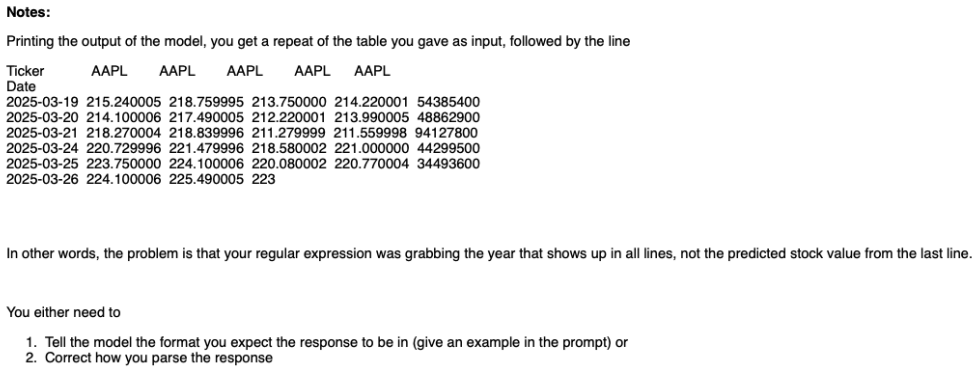
Proj2Comp399V2 is the updated version of project 2 that is a working version with an accurate future stock prediction.

Correction of project 2 – which was the AI powered stock predictor using LLMs and Python

1. In the email sent regarding my project 2 was to correct the response from the LLM as I was getting 0.0 or 2025 as my outputs after inputting Apple's stock data from yahoo finance which is between 200-300 dollars so both outputs make no sense. (Screenshot via email on project 2)



Notes:

Printing the output of the model, you get a repeat of the table you gave as input, followed by the line

Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2025-03-19	215.240005	218.759995	213.750000	214.220001	54385400
2025-03-20	214.100006	217.490005	212.220001	213.990005	48862900
2025-03-21	218.270004	218.839996	211.279999	211.559998	94127800
2025-03-24	220.729996	221.479996	218.580002	221.000000	44299500
2025-03-25	223.750000	224.100006	220.080002	220.770004	34493600
2025-03-26	224.100006	225.490005	223		

In other words, the problem is that your regular expression was grabbing the year that shows up in all lines, not the predicted stock value from the last line.

You either need to

1. Tell the model the format you expect the response to be in (give an example in the prompt) or
2. Correct how you parse the response

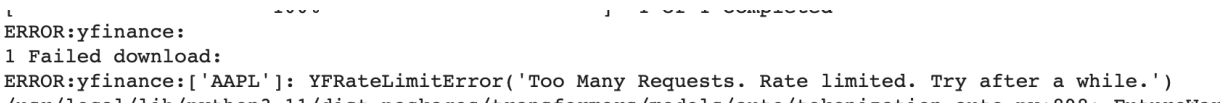
2. I corrected the output from the LLM to get an actual number that made sense in the 200 – 300 range. It first retrieves the opening prices for the last five trading days of the specified stock(Apple). This historical data is then formatted into a text prompt, asking the model for the opening price on the following day. The prompt is tokenized and fed to the language model, which generates a textual response. A regular expression is used to extract a numerical price from the model's output string. Finally, the extracted predicted price is printed to the console.

Extension of Project 2 that we talked about during the stars presentation:

1. I thought the idea that you gave regarding taking the front page of the New York Times and feeding this into the language model as well for more accurate data was a cool idea and I wanted to see if it would help improve price predictions.
2. The idea I settled on was to import the Google Search python API and implement queries regarding the Apple stock, which I will then use the built in search method and feed into my stock predictor, in my predict_stock_price method.

Problems I encountered:

1. The yfinance library (Yahoo Finance API which I used to download the Apple stock ticker data from) has a limited amount of queries it will allow before temporarily blocking your IP address. This was a profound problem as I had trouble finding another stock data library that integrated easily with my code.



2. The google search API did not integrate well with my code and I had a lot of trouble integrating this into the stock price generator. Below is a screenshot of an error that I had a lot of trouble being able to fix.

Conclusion:

Overall this project was a lot of fun and I really enjoyed it. I learned that I will need much more significant hardware to run a larger version of an LLM and I would need to implement a time series model to get a much more accurate predicted price that I would need to use for personal gain in investing.