I remember the instant on my 18th birthday I realized I could then legally trade stocks, immediately opening up a whole new world to me. The idea of making passive income is evidently enticing to anybody. For that reason, though my college major of computer engineering is not oriented in finance, I began slowly learning more about the subject. I learned to see it not only as its own industry for bankers and finance majors, but more as an important tool and necessary life skill in gaining wealth in society.

I'm not the only young person intrigued by the stock market. ***Robinhood***, has played the largest role in bringing young people into the market.

More than five years ago, more than 80% of Robinhood's user base belonged to the "Millennial" age demographic, and its average age was 26. In the last three years, I have watched the app become commonplace to many of my peers' phones, with over 10 million total users (equivalent to the population size of Georgia (the state)).

Known as the "Robinhood Effect", there was a huge change in the financial industry pioneered largely by Robinhood, in which broker fees practically became nonexistent. The appeal of instantaneous, fee-less online trades eventually became the new normal for (increasingly younger) investors. By additionally offering users the ability to trade *options* and *cryptocurrencies*, Robinhood has continuously capitalized on a younger, typically riskier investor looking to make a short term flip rather than buying and holding for long durations of time.

Over the last two years, I've watched many videos and read many articles about investing. Additionally, I've noticed more and more of my peers invest in the market to the point where discussing trending stocks is a part of regular small-talk conversations, akin to sports or politics. Anecdotally, an increasing number my peers trade both stocks and options in the short term. Though options trading can be used as a hedge to lower risk, I often find it hard to imagine that so many people I personally come across, whether in real-life or online (YouTube, Reddit, Twitter, etc.), end up profitable this way, especially since none of these people are "experts". From my understanding as an engineering student, some of the top ranking financial institutions with immense research budgets and teams stacked with PhDs in everything from Computer/Data Science to Psychology are constantly improving financial models for the exact purposes of predicting future prices and automating high-frequency trades (with lower latency in data retrieval than laymen) . For this reason, I have always been skeptical whenever I hear of someone making a rather unique or risky trade, as within the two year time period I've been able to purchase stocks, the S&P 500 Index went up about 14%, averaging a 7% increase a year. This is *despite* the recent coronavirus-fueled economic downturn. Those returns are on par with how it has trended historically over large periods of time.

So what about your co-worker who claims to have made a quick 300% ROI buying Hertz stock after they declared bankruptcy, or your cousin who tries to time his trades based on earnings reports? In a social atmosphere it's important to remember people typically tend to only disclose their successful investments. I always find it difficult to believe that the majority of non-institutional, individual stock/options traders are making consistent short run profits over long periods of time.

Last month a friend of mine introduced me to ***Webull*****,** a financial services company very similar to Robinhood. Both companies utilize a similar marketing strategy where if a user recommends the app to a friend, both individuals receive a free stock. This incentive alone along with my friend's insistence led me to download the app. The biggest difference I noticed between the two apps was that Webull gives the user tools for more in depth data analytics. Specifically, there is a list of more than twenty technical indicators at the user's disposal.

-
-
-
-
-

Add alt text


ETHUSD Bitfinex
244.910 +8.800 +3.73%
1 Min
Daily
Weekly
Monthly
Quarterly
Yearly
NONE
359.706
363.300
+21.84%
285.139
-3.42%
-28.68%
136.005
-53.93%
83.000
01/01/2019
06/01
09/01
01/01/2020
06/01
09/01
MACD(12,26,9) MACD:2.224 Signal:2.254 Histogram:-0.030
25.75
-2.71
RSI(14) RSI14:60.682
70
30

I did not know anything about these indicators aside from my slim understanding of day-trading, which was basically that when certain lines on a computer screen cross they buy and when other lines cross they sell. However, after googling "best technical indicators for trading" and doing some reading, I came to the realization that most of these "indicators" are in one way or another related to *moving averages*, a concept I was very familiar with from my math/engineering background in digital controls and signal analysis/filtering. After doing some research to pick popular yet reliable indicators, I chose to use the Moving Average Convergence-Divergence (MACD) along with the Relative Strength Index (RSI) indicators. Other results from my initial search included moving averages (simple and exponential), along with other similar variations such as the Bollinger Bands. In fact, almost all of these indicators are fundamentally based off of moving averages.

After learning this, I set out to find my own answer to the question: "How good are technical indicators?". After all, I am not currently employed and have had lots of free time recently due to the global pandemic, so I took this as an opportunity to work on my first truly independent coding project.

My idea: utilize the MACD and RSI indicators in correspondence with each other to maximize profit. Simulate historical data through dual-indicator model to see how an indicator driven robot would have traded over the same period for different stocks and cryptocurrencies.

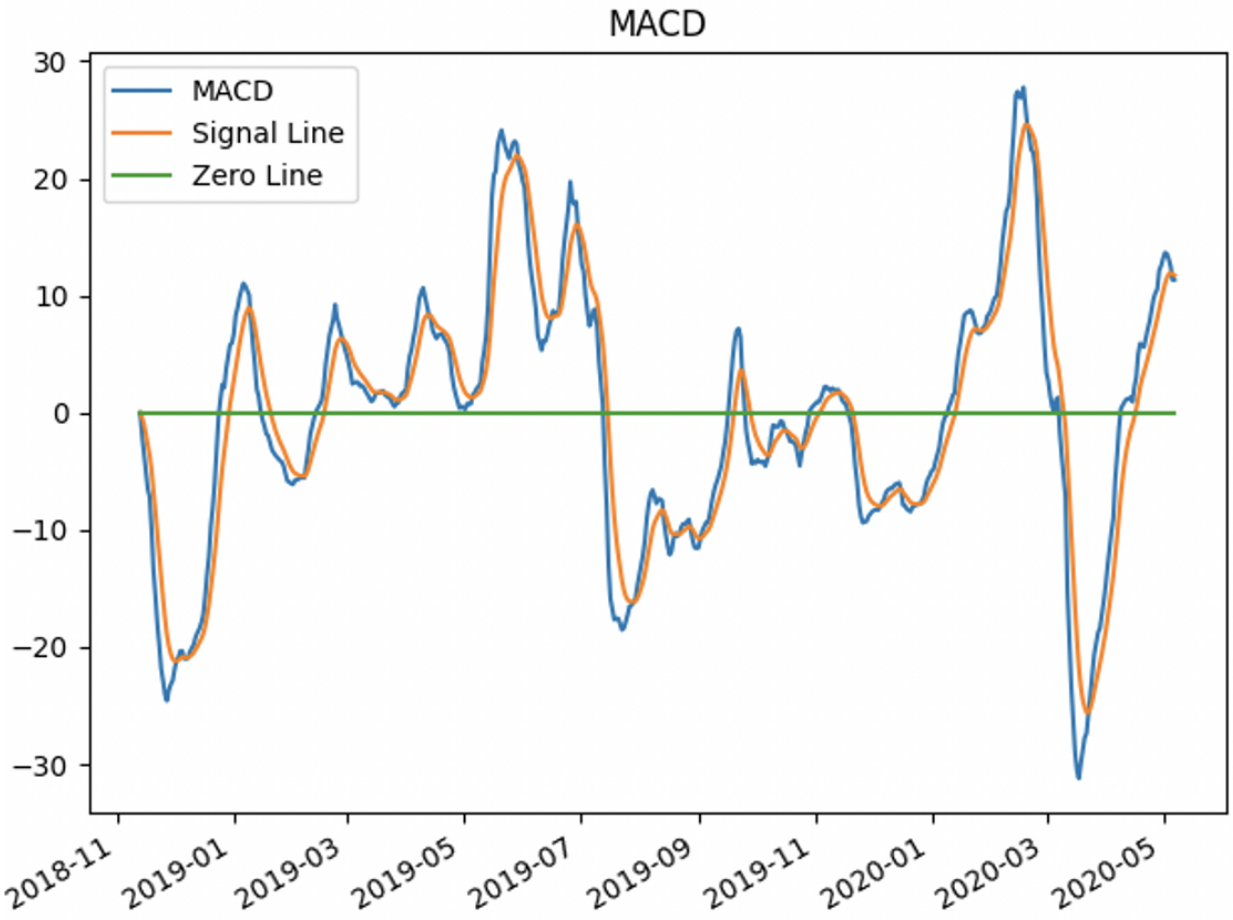
I chose Python as the programming language solely because it's one of the most convenient for working with API's. I had briefly used Python before, but not more than a few times (my background is more-so in MATLAB, C++, and Java). In my implementation, I purposely did not utilize the numpy and pandas libraries efficiently, as I wanted to first just practice generic python syntax altogether. I used a pandas function for retrieving stock data, and briefly used numpy for a few specific calculations.

After beginning with the API/data retrieval, I then wrote functions to calculate exponential moving averages, a vital calculation for both MACD and RSI, from scratch. Eventually, I developed functions for calculating both of those indicators for a given dataset for a given time delay. I followed convention and made the default for MACD time parameter tuning strategy (12, 26, 9) (day-delay for the three exponential moving averages that make up MACD) and RSI to 14 days.

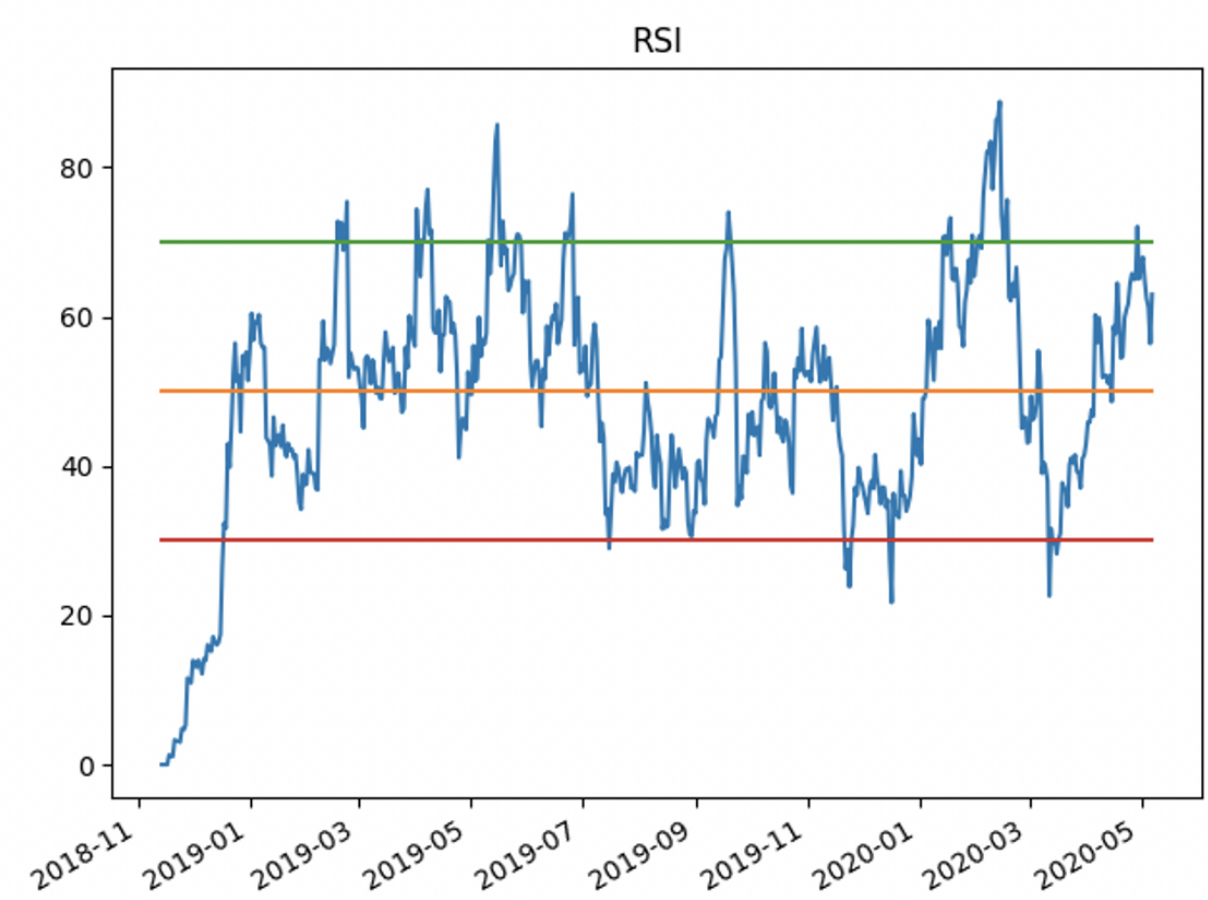
The below data is my ~1.5 year long analysis of Ethereum, the 2nd most popular cryptocurrency (behind Bitcoin) in the world, compared to a screenshot from the Webull app.

•

Add alt text


MACD
MACD
Signal Line
Zero Line
30
20
10
0
−10
−20
−30
2018-11
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05

Add alt text


RSI
80
60
40
20
0
2018-11
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05

Add alt text

Day-traders use the RSI indicator to buy when RSI is less than 30 (undervalued asset) and sell when it is greater than 70 (overvalued asset). Similarly, the MACD indicator can be used to predict changes in price direction based when the MACD-Line intersects with the Signal-Line, indicating a *crossover*. The direction of the intersection indicates whether to buy or sell. Using the framework I developed to grab the historic equity price-data and calculate RSI and MACD, I then developed a model to simulate hypothetical buying and selling over the given time-period based on both indicators and a specific trading strategy. I also fed the model data from before the start date to properly tune the RSI before transactions begin (RSI starts at extreme due to calculation). At every RSI triggered transaction (RSI just became lower than 30 or greater than 70), the model automatically bought/sold 100% of a constant, hypothetical amount (in dollars) of equity, we can call x. At every MACD crossover, the relative level of RSI at the same date is checked to determine what percent of x to stake. I did so by linearly interpolating from 50% to 100% between RSI values 50 and 70 (and 50 and 30) such that when a MACD crossover occurs *and* the RSI is at an 'extreme' value, we stake 100% of x. But if we have a MACD crossover and the RSI is exactly 50, RSI is not helpful in making a determination of the future of the equity price and we would stake instead 50% of x (MACD yes but RSI no).

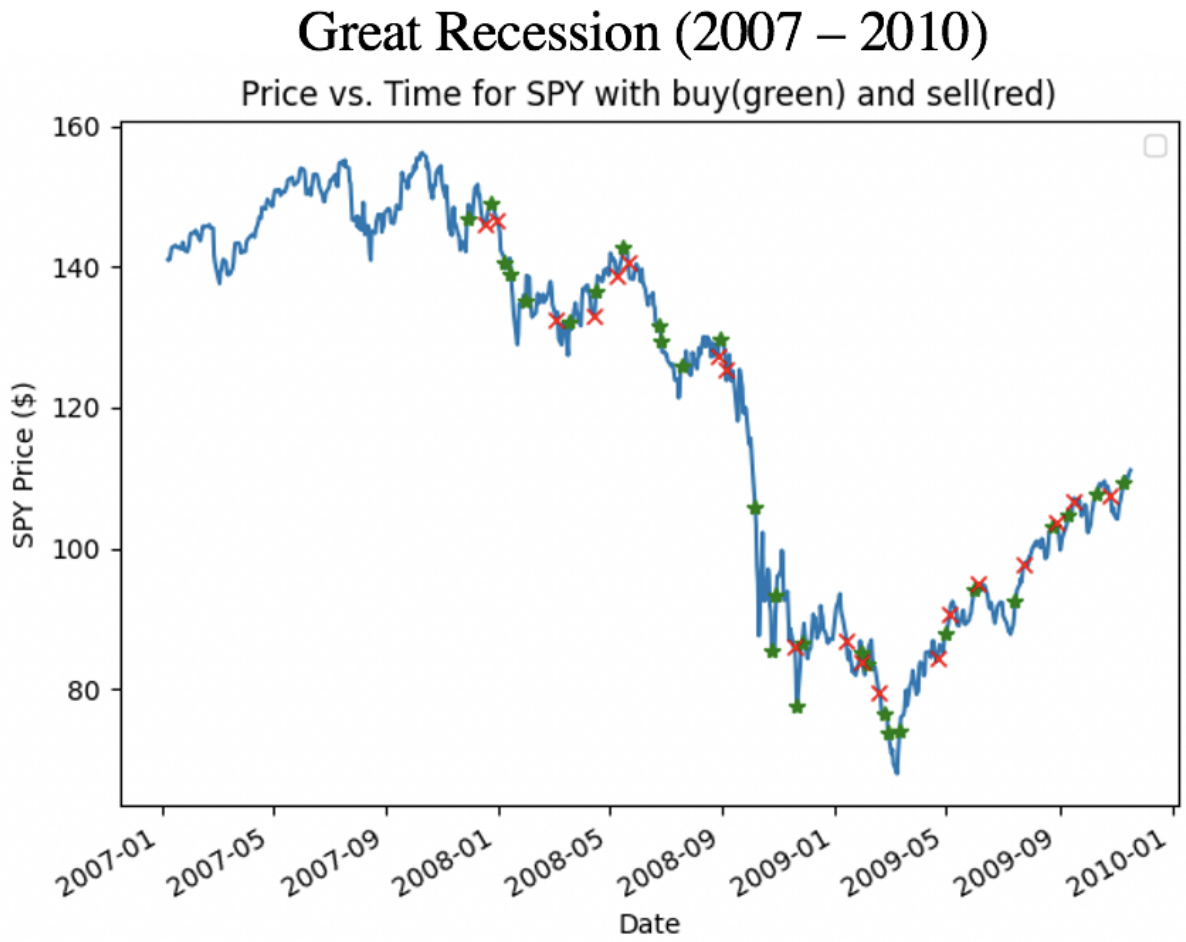
It might seem complex and abstract, but in reality, all the math behind it is just a few lines of algebra. The only thing I really came up with was the (very simple) staking strategy; the calculation and interpretations of MACD and RSI are well-known and documented. Before ever researching these indicators, I assumed that a trader who is well equipped with technical indicators would have an edge on other traders.

Upon completing a working model, I was very interested in the results of the model when applied to different types of equities over different lengths of time. Below are

some results of the S&P 500 (SPY) over both the short and long run, along with Ethereum.

-
-
-
-
-

Add alt text



ROI over given duration: -26.6%

ROI with model: -2.2%

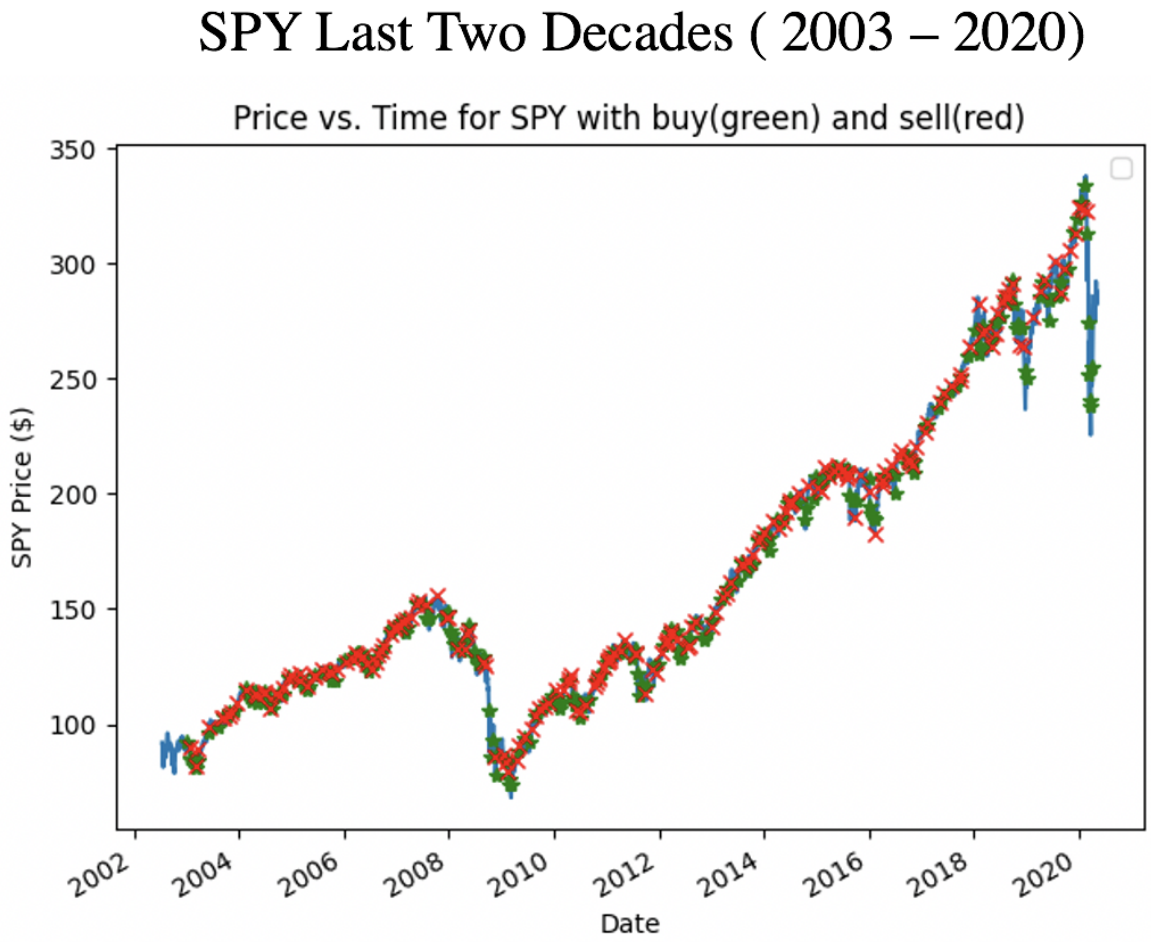
*ROI over given duration - exact percent of change from start date to end date

*ROI with model - simulated return on investment from model

During a recession like we experienced in 2008-2010, the indicator-based trading robot *beats the market*! This is mostly due to the chosen window of time (only three years, and they are three specifically bad years). In relatively short term times of uncertainty, the model is safer because it never holds that much equity in the market for that long. However, this causes us to lose out on the benefits of compound interest over longer periods of time. Take a look at the SPY data over a ~20 year long period below:

-
-
-
-
-

Add alt text

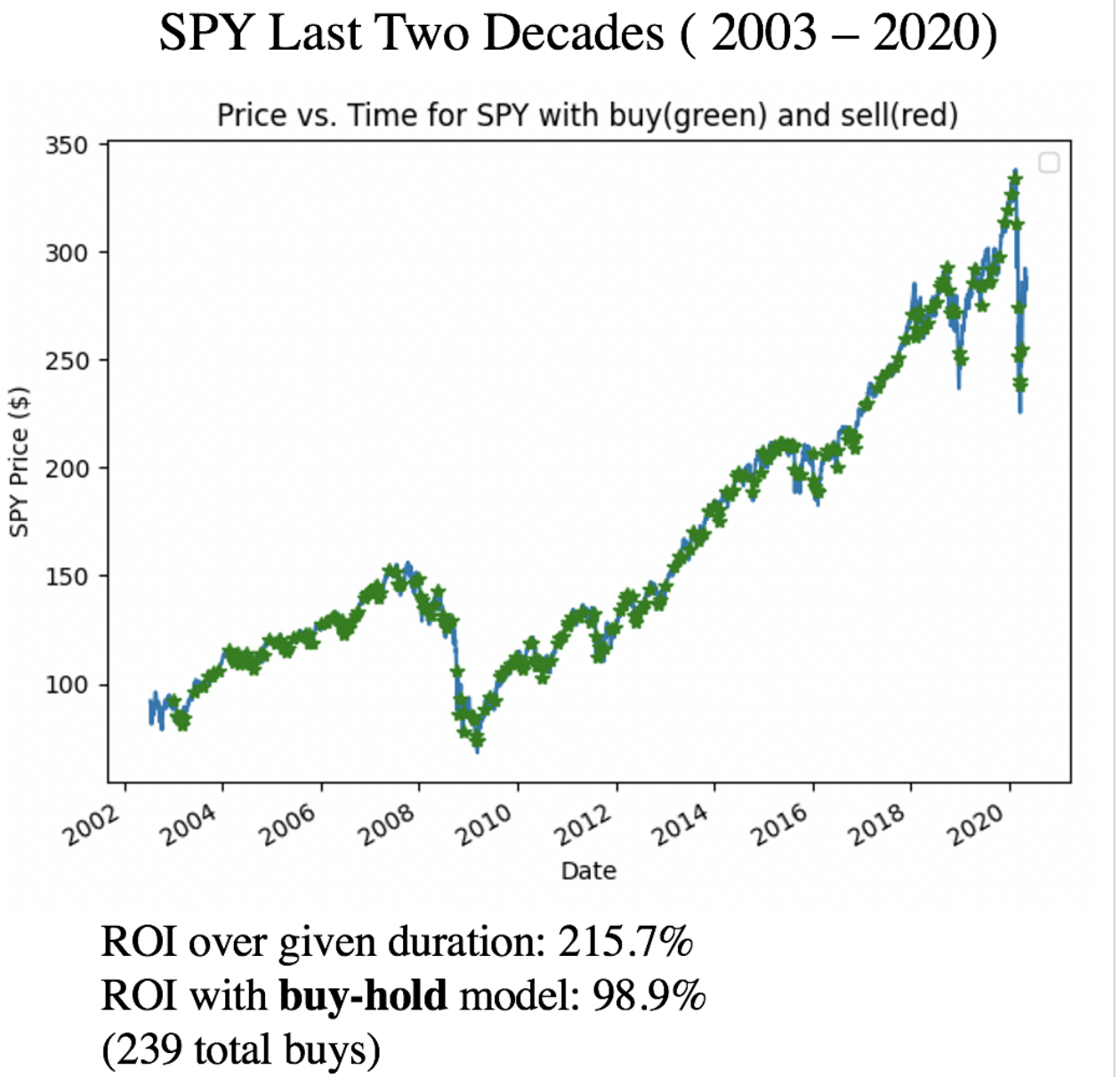


ROI over given duration: 215.7%
ROI with **buy-sell** model: 46.3%

Simulated over 17 years, this model produces positive returns! So, it seems like this model predicts correctly more than 50% of the time. However, as you can tell from the respective potential returns on investment, we miss out on the full benefits of buying and holding. This then led me to wonder, how would the same model work if we only bought shares and held them? The results were quite eye-opening:

-
-
-
-
-

Add alt text

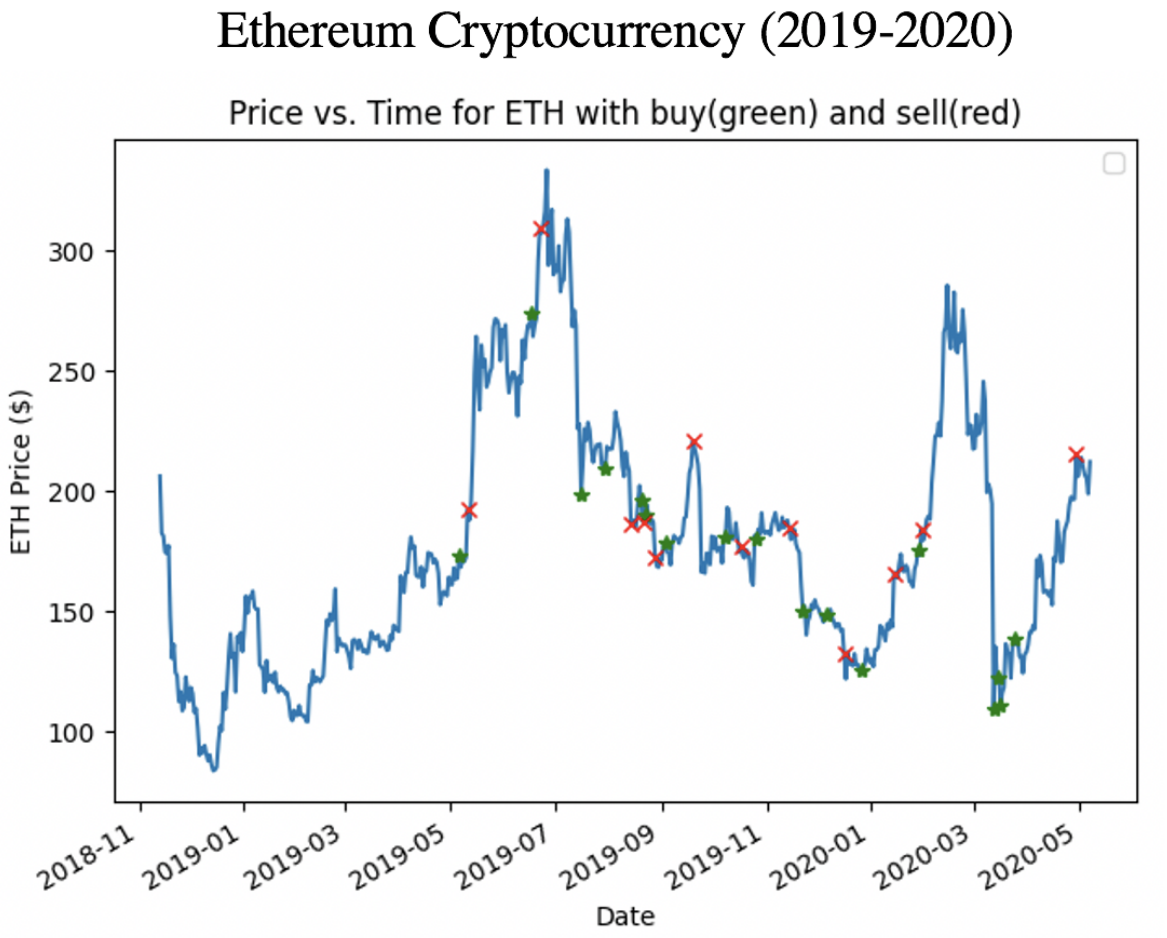


With the same interval of time and same buying strategy (just without selling), this robot averaged slightly more than one buy-per month. Over 17 years, by *only buying and holding* the S&P 500 at the indicator-predicted lows, we actually simulate better returns over time than the buy-sell robot did by buying and selling. For the S&P 500, at least, we would have been far better off buying and holding than buying and selling. This is because the S&P 500 trends upwards over time. Though this automated trading scheme is better than flipping a coin, it is not better than purely buying and holding the S&P 500.

However, the S&P 500 is not at all volatile. I tested the model on a more volatile equity, Ethereum, to see how the model would work for cryptocurrencies.

-
-
-
-
-

Add alt text



ROI over given duration: 31.6%
ROI with model: 79.6%

Unlike the S&P 500, this equity does not necessarily consistently trend upward overtime, and was rather volatile. For volatile equities that don't necessarily have room for long term growth, this buy-sell model is likely more profitable than strictly buying and holding. That is, however, a very specific type of equity.

It is hard for me to make any bold claims or determinations from all this information. Especially since I have not exhaustively tested this model. In reality, I should run thousands of these individual simulations, varying the chosen asset, the time-duration, the aggressiveness of the trading strategy, the number of indicators used in the strategy, their respective weights, etc. But for now I will leave that for the folks on Wall Street. I am more interested in the key takeaways from this endeavor. The model can confirm that there is definitely some validity in the indicators, considering it seems to yield an increasingly positive return after simulating trades for longer periods of time. However, it likely would have been much more profitable if it simply bought and held over this time.

In the days where automated, high-frequency trades driven by complex neural-networks and massive amounts of data dominate the stock market and Wall Street firms, I did not expect these indicators to be particularly successful. Especially in the age of the AI revolution, the idea that anyone with widely accessible real-time indicator data can compete with Wall Street using these fundamentally lagging (moving averages have a delay inherently) indicators did not seem realistic to me. **After seeing the results first-hand, it makes me wonder why companies like Webull offer their users these tools in the first place. Is it to give their users the best edge in trading? Or is it to give them the false-confidence that they're being "smart" in their trades?**

This was really just an idea for a simple project I was able to see myself being able to complete and made it a reality. I'd like to stress again that my relationship with the financial world is rather new, and though I did just write a whole article on it, I am certainly far from an expert. I did, however, learn a ton about working with python that I did not know previously despite my previous times using it. Much of what I learned regarding the indicators was from Investopedia and Wikipedia, but most of that was the generic calculation and interpretation. In reality, this was nothing more than some much needed python practice and an experiment that left me with even more evidence that the best investing strategy, as even Warren Buffett says, is to just buy and hold index funds.

My next step? I now plan on diving deep into an online Python course and hopefully become fluent in using the numpy and pandas libraries.

If you have the free time, pursue that shower thought you had the other night. I never thought that the model would ever beat the S&P 500, but I'm glad I still made it because it gave me something to do for the last few weeks and taught me a lot about python, APIs, moving averages, and even LinkedIn articles.

Stay safe friends!