An algorithmic trading strategy based on Robinhood user data

David Nagy

Introduction

Anecdotal evidence suggests that most retail accounts at Robinhood (an online zero-commission brokerage) lose money. This may be due to reliance on market orders (crossing the bid-ask spread with every trade) or bad predictions regarding the future direction of asset prices. Unlike in options space (where Robinhood directs its order flow to firms like Citadel, who are happily milking clients by providing mile-wide spreads[1]), the first reason might be weaker in the case of equities traders because of the higher liquidity of the asset class. Hence is could be possible to outperform the market by doing the opposite of what they do. That is what this project investigates.

Fortunately Robinhood publishes some data about its user accounts. (At least up until now.[2]) The Robintract project[3] has historical data on the number of accounts holding each stock, sampled hourly. I used this data, and stock prices from an Eikon terminal (remote access, courtesy of Cambridge Judge Business School and the Cambridge Marshall Library of Economics) to backtest the strategy of buying stocks after Robinhood users sold them and vice versa.

More specifically, I restricted the asset universe to the current constituents of the S&P500 index (for reasons of computational capacity and asset liquidity), for all of which Robintrack data is available back until May 2018.[4] Then I constructed a popularity measure for each stock (at each moment, the number of accounts holding that stock relative to the number of account-stock pairs such that the account holds the stock). Each day, I find a couple of stocks whose popularity decreased (increased) the most from the previous market close to the current market close and buy (sell) them for a day (execution is assumed to happen at mid close price).

I uploaded my code on GitHub: https://github.com/dpn29/Robintrack-project/blob/master/Robintrack%20project.ipynb.

---

[1] https://themargins.substack.com/p/robinhood-and-how-to-lose-money

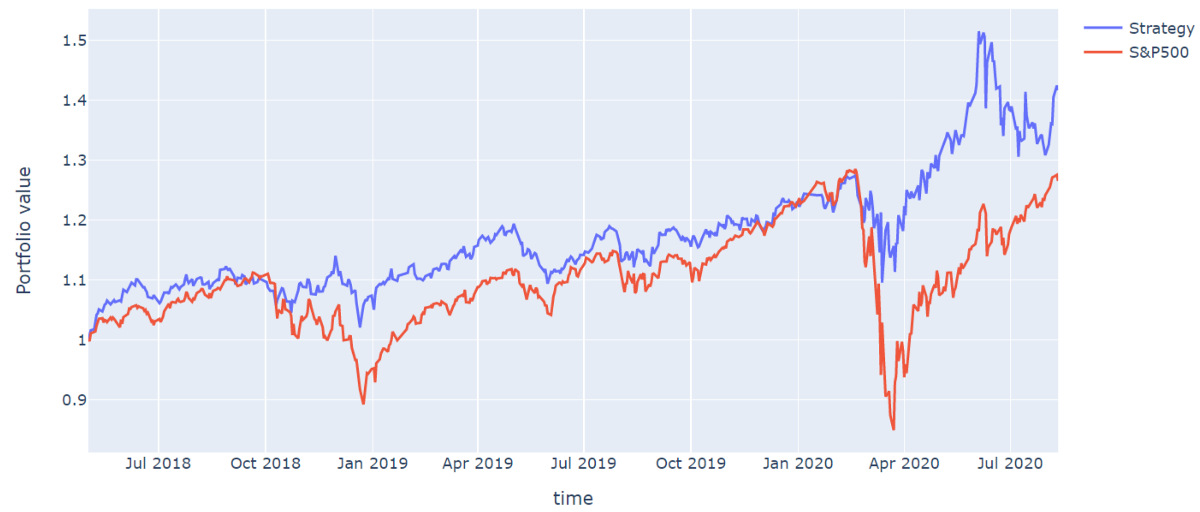
[2] https://fortune.com/2020/08/10/robinhood-popularity-data-robintrack-stock-market-trading-tracker/

[3] https://robintrack.net/

[4] Note that this restriction introduces look-ahead bias. Stock A being part of the index today is a piece of information that 1) helps predict the return on A between May 2018 and today and 2) is not known in May 2018. Hence using a stock universe restricted in this way assumes that we could have perfectly predicted which stocks will be selected into the index; since the S&P500 consists of the largest companies this implies that these stock are more like to have outperformed the market in the past 2 years. I overcome this issue by comparing the performance of my backtests to the performance of holding this *fixed* set of stocks over the timeframe.

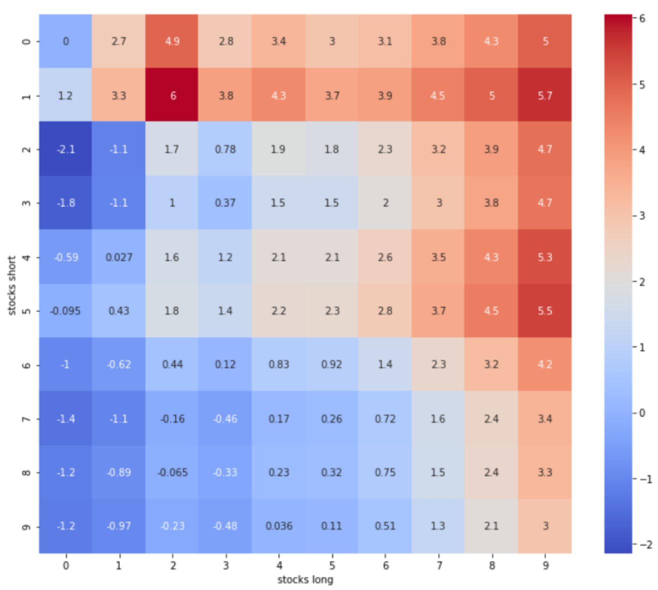
Results

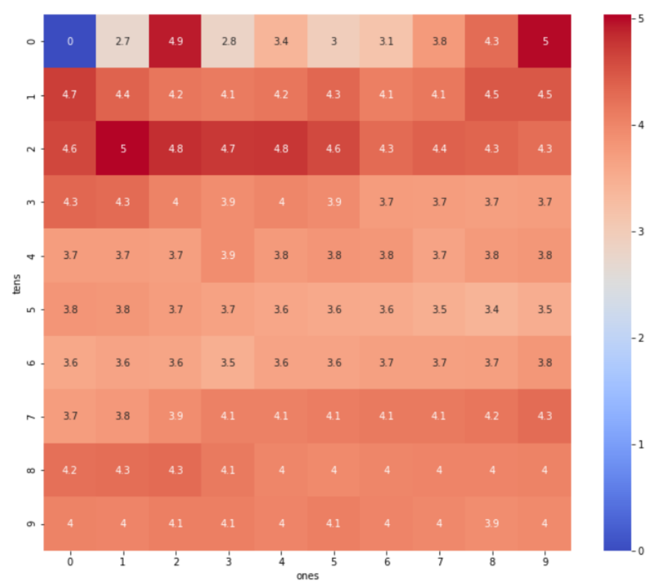
The below chart shows the performance of the strategy of buying the 20 stocks losing most popularity and shorting the 5 stocks gaining the most popularity each day, compared to the S&P500.



The Sharpe ratio of the strategy is 5.44% against the S&P500's 3.66%. So even though the strategy finishes just a bit above the baseline, it would be possible to trade it with higher leverage and have the same volatility as the S&P500 – so if we were to equalise the risk, the strategy would have a substantially higher return than the index.

It is possible with my code to optimise the number of stock held/shorted and the length of holding days for each stock. The below heatmaps show the Sharpe ratio (a much





better measure of strategy performance than the return achieved) for different numbers of longed/shorted stocks and for a long-only strategy up to 99 stocks each day.

However, as past returns are not necessarily indicative of future returns, one should avoid over-optimising. The strategy of long 20 stocks and short 5 stock makes sense based on the data (shorting more than 5 starts to decrease performance, the range of optional longs looks like 8-30 based on the heatmaps) and also intuitively: with a 25-stock portfolio, the idiosyncratic risk from each stock averages out enough and the average signal is stronger if we pick fewer stocks. So we trade-off between low portfolio volatility and high expected return via the number of stocks held; it makes sense to long more than we short as stocks go up most of the time.

Evaluation

What is the rationale behind the idea that this signal might work? It seems unlikely that people trading on Robinhood move the market. However, Robinhood makes money from selling its order book data – and the buyers (many of them HFTs), can run their own quant strategies based on that data (e.g. frontrunning), thereby amplifying the impact of retail trades.[5] A second reason might be overreaction to bad news. If a retail trader reads an article about Facebook's Cambridge Analytica scandal, chances are the news is already incorporated into the price by the time he gets to sell his FB stocks. (Or other kinds of trend following, which my analysis provides some, but not strong, evidence for.) Hence the strategy discussed above will buy FB when the price is already low – then, if the negative news were overacted to by the market,[6] the price will rebound on average.

The Sharpe ratio looks promising, but it's statistical significance is hard to evaluate (even if it was statistically significant, there is no guarantee that it will work like it did in the past). Furthermore, having to pay any commission or a non-negligible bid-ask spread would be detrimental to profitability as the whole portfolio of 25 stocks can turn over completely on a daily basis. Regarding the data, there might be a lag for the publication of popularity data, which would definitely take away from performance.

[5] https://blockworksgroup.io/blog/robinhood-sells-your-data-but-does-that-matter
[6] Note that this would be a violation of the EMH...

Disclaimer