Active Portfolio Management with Mean Variance Optimization

Magnus Miller

Optimizations in Finance

Date: December 12, 2024

1 Introduction

Active portfolio management aims to outperform benchmark indices by leveraging strategic asset allocation and security selection. This project focused on applying portfolio optimization techniques to construct and evaluate portfolios under different constraints, including long-only, long-short, and 130/30 strategies. By solving mean-variance optimization problems, the study incorporates practical constraints such as turnover limits and beta alignment with the S&P 500 benchmark. The analysis evaluates portfolio performance through cumulative returns, utilizing both standard models and rolling window approaches to capture the adaptability of the strategies in dynamic market conditions.

2 Methodologies

This project was completed in five parts. The first portion of the project pertained to accessing historical stock data ranging from volume traded and market capitalization to daily close prices. The second portion of this project was built around developing three different portfolio optimization strategies. The next portion dealt with handling the historical data and segregating it into in-sample calibration data and out-of-sample testing data. Next, the project shifted to creating visualizations. The last portion of the project was focused on constructing a portfolio of stocks to be analyzed.

2.1 Historical Data

There were many choices to access historical stock data whether it be using Microsoft Excel or Bloomberg. Due to my previous experience, ease of use, and easy integration in Python, I chose to utilize Yahoo Finance. Yahoo's stock data API is widely supported and known for its easy integration and ease of use in Python with the help of libraries such as *yfinance*, *pandas*, and *numpy*. Using these libraries, I was easily able to filter and select stocks that met any given requirements from volume traded to market capitilization.

2.2 Optimization Strategies

At the heart of this project lies the optimization strategies. In total, there were three optimization strategies implemented: Long-Only, Long-Short, and 130/30 Long-Short. In general, the optimizations are different versions of the Mean-Variance Optimization found below.

$$
\begin{aligned}
\text{minimize:} \quad & \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} - \mathbf{q}^T\mathbf{x}, \\
\text{subject to:} \quad & \begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b}, \\ \mathbf{G}\mathbf{x} \leq \mathbf{h}, \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{cases}
\end{aligned}
$$

In the above quadratic program, $\mathbf{x}$ denotes the portfolio weights, $\mathbf{P}$ denotes the covariance matrix of the asset returns, and $\mathbf{q}$ is the negative vector of expected returns. The remaining variables contribute to different constraints which will be explained later.

From this equation, I was able to build out the Long-Only optimization by imposing the following constraints:

1. Fully invest the portfolio: $\mathbf{1}^T\mathbf{x}^T = 1$
2. Implement portfolio beta: $\boldsymbol{\beta}^T\mathbf{x} = 1$
3. Long-Only: $\mathbf{x} \geq 0$

With these constraints, we have the following quadratic program.

$$\text{minimize:} \quad \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} - \mathbf{q}^T\mathbf{x},$$
$$\text{subject to:} \quad \begin{cases} \mathbf{1}^T\mathbf{x} = 1, \\ \boldsymbol{\beta}^T\mathbf{x} = 1, \\ \mathbf{x} \geq 0. \end{cases}$$

To develop the Long-Short optimization, the same quadratic program was used with a small modification. Instead of limiting $\mathbf{x}$ to positive values (Long-Only), we add the constraint $-1 \leq \mathbf{x}_i \leq 1$ for every i. The above modification gives the following.

$$\text{minimize:} \quad \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} - \mathbf{q}^T\mathbf{x},$$
$$\text{subject to:} \quad \begin{cases} \mathbf{1}^T\mathbf{x} = 1, \\ \boldsymbol{\beta}^T\mathbf{x} = 1, \\ -1 \leq \mathbf{x}_i \leq 1 \quad \forall i. \end{cases}$$

Finally, the 130/30 Long-Short strategy implies that, at any point, a given portfolio is invested 130% in long positions and 30% in short positions. To implement this constraint, we alter the long-short bounds from above, and subject the portfolio weights to different values of $\mathbf{G}$ which can be seen below.

$$\text{minimize:} \quad \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} - \mathbf{q}^T\mathbf{x},$$
$$\text{subject to:} \quad \begin{cases} \mathbf{1}^T\mathbf{x} = 1, \\ \boldsymbol{\beta}^T\mathbf{x} = 1, \\ \sum \mathbf{x}_i^+ \leq 1.3, \\ \sum |\mathbf{x}_i^-| \leq 0.3, \\ -0.3 \leq \mathbf{x}_i \leq 1.3 \quad \forall i. \end{cases}$$

It is important to note that each of the aforementioned optimization strategies were implemented in Python using the *qpsolvers* library. This library serves as an optimizer interface for Python users as it allows the use of various quadratic programming solvers written in many different projects. For the sake of this project, I used *OSQP* as the solver algorithm for each optimization due to its technical support within the library documentation. Although the *OSQP* solver worked well, I did have to increase the maximum number of iterations to 800 to guarantee that every program was solved.

In addition to the quadratic program optimizations, the covariance matrices, α vectors, and β vectors were calculated. I calculated the covariance using the *numpy* library's *cov()* function. To calculate the β

vectors, I used the *scikit-learn* library's *LinearRegression()* function to estimate values. Finally, I decided to use a 2-month, or 60-day momentum average to calculate and rank the α values.

2.3 Data Segregation and Handling

For this project, two model constructions were used. The first constructed called for the first 60% of the 5 year historical dataset to be used for the calibration of the model while the last 40% was saved for testing. The second construction required implementing a rolling window model. For this, I assumed that portfolio weights would be adjusted on a quarterly, or roughly 90-day, time frame. Using that, I then calculated the total whole number of windows withing the 5 year time frame of the data. I then iterated over every window in the dataset chronologically using the $n^(th)$ window to calibrate and the $(n+1)^(th)$ window to test. Each window of testing data was then combined chronologically to calculate returns.

2.4 Visualizations

In order to visualize the portfolio return data, especially in comparison to the S&P 500 returns over the same time period, I chose to use the *matplotlib* library's *pyplot* class to create and design the figures seen later. No massive creative liberties were taken here.

2.5 Stock Choices

Besides the optimization strategies, the second most important portion of this project is the choice of stocks to be used. The project called for 25-50 stocks to be chosen which meet the following criteria.

1. 5 years of historical data available.
2. Market Capitalization $\geq$ \$2B.
3. Volume Average $\geq$ \$20M

When it came time for me to swap from my test files to actual stocks, I had no real intuition or insight into what to choose. In short, the process I followed to select stocks to put into my portfolio consisted of me putting the listed requirements into an online stock parser and selecting companies that I either (a) heard of in my personal experiences, (b) had very impressive returns recently, or (c) had cool sounding names. I tried to implement some variation and sector diversification in my choices, but at the end of the day, you probably could have thrown a dart at the wall of possible choices and ended with my portfolio. That being said, the full portfolio can be found below in Table 1.

Ticker	Company Name
MU	Micron Technology Inc.
BAC	Bank of America Corp
AMZN	Amazon.com Inc.
AAPL	Apple Inc.
PFE	Pfizer Inc.
SNAP	Snap Inc.
VALE	Vale SA
ABEV	Ambev SA
GRAB	Grab Holdings Ltd.
F	Ford Motor Co.
TSLA	Tesla Inc.
MARA	MARA Holdings Inc
PCG	PG&E Corp.
INTC	Intel Corp.
NVDA	Nvidia Corp.
SMCI	Super Micro Computer Inc.
RIOT	Riot Platforms Inc.
META	Meta Platforms Inc.
KGC	Kinross Gold Corp.
HL	Hecla Mining Co.
CRM	Salesforce Inc.
CELH	Celsius Holdings Inc.
ENPH	Enphase Energy Inc.
AXSM	Axsome Therapeutics Inc.
AEHR	Aehr Test Systems
LSCC	Lattice Semiconductor Corp.
ELF	e.l.f Beauty Inc.
MSFT	Microsoft Corp.

Table 1: Stock Tickers and Company Names

3 Results

The figures below illustrate the cumulative returns of the optimized portfolios compared to the benchmark S&P 500 index. Each plot displays the performance of the long-only, long-short, and 130/30 portfolios, highlighting their ability to outperform or under-perform relative to the market over time. The x-axis represents time, while the y-axis shows the growth of $1 invested, allowing for a clear comparison of the portfolio strategies against the benchmark.

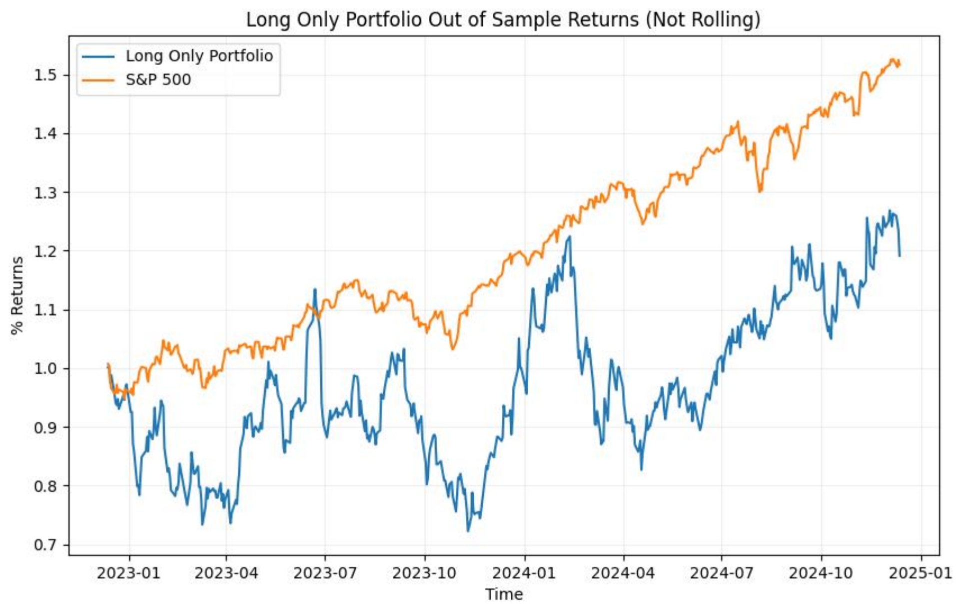

Figure 1: Cumulative returns for the long-only portfolio compared to the S&P 500 benchmark. The portfolio is optimized using the entire in-sample data and tested on the out-of-sample data, demonstrating its overall performance without periodic re-balancing.

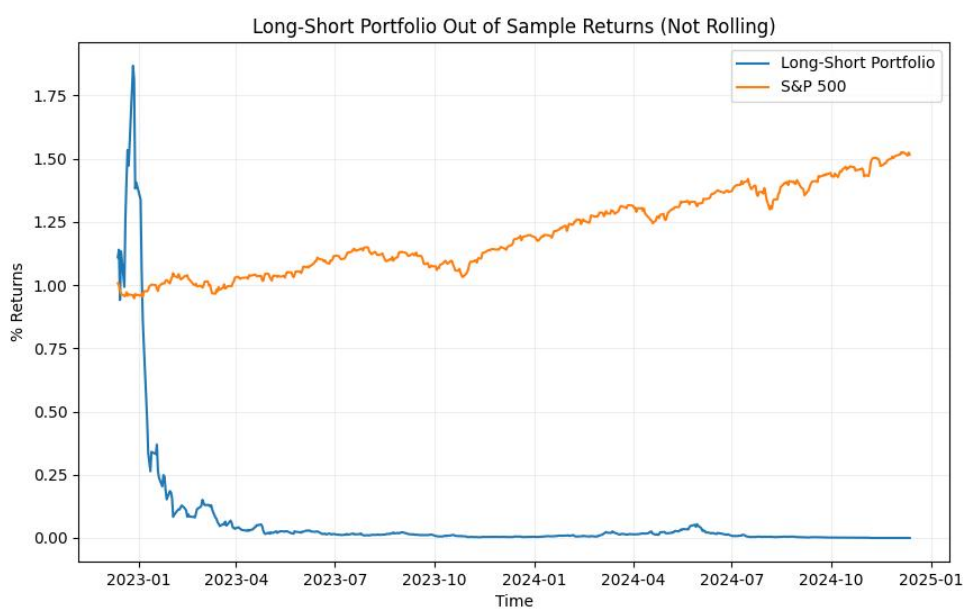

Figure 2: Cumulative returns for the long-short portfolio compared to the S&P 500 benchmark. The portfolio is optimized to allow both long and short positions, and its overall performance based on a single in-sample optimization.

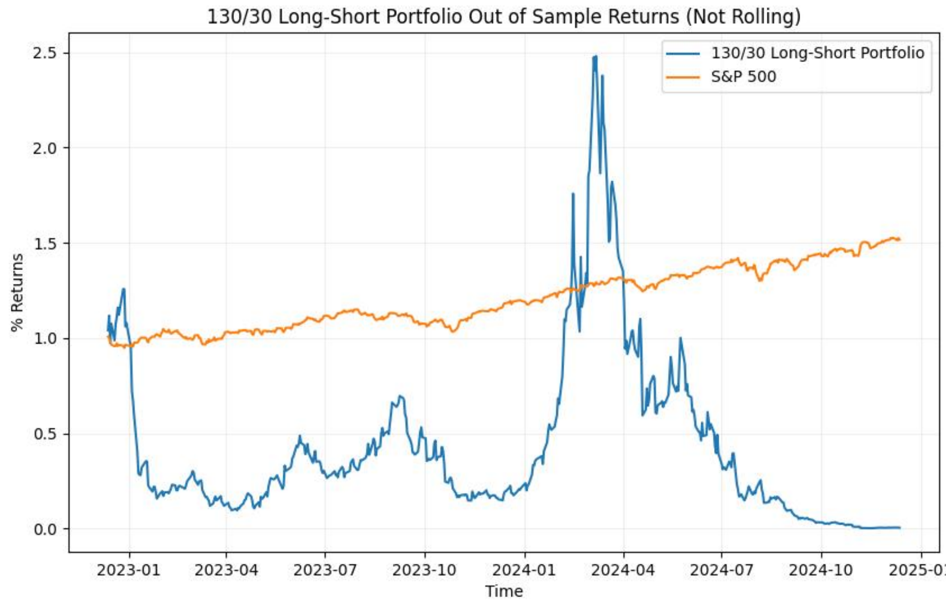

Figure 3: Cumulative returns for the 130/30 portfolio compared to the S&P 500 benchmark. The portfolio combines a 130% allocation to long positions and 30% to short positions, with results reflecting performance from a single in-sample optimization applied to out-of-sample data.

Figure 4: Cumulative returns for the long-only portfolio compared to the S&P 500 benchmark. Returns are based on rolling window optimizations and demonstrate the portfolio's performance relative to the market.

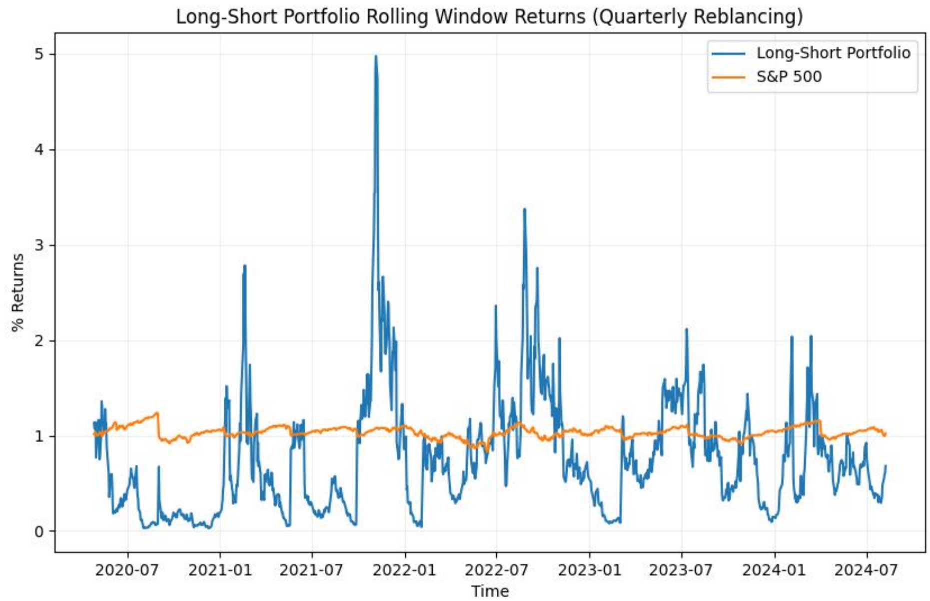

Figure 5: Rolling window cumulative returns for the long-short portfolio compared to the S&P 500 benchmark. The figure showcases the portfolio's performance when rebalanced quarterly with periodic adjustments based on rolling optimizations.

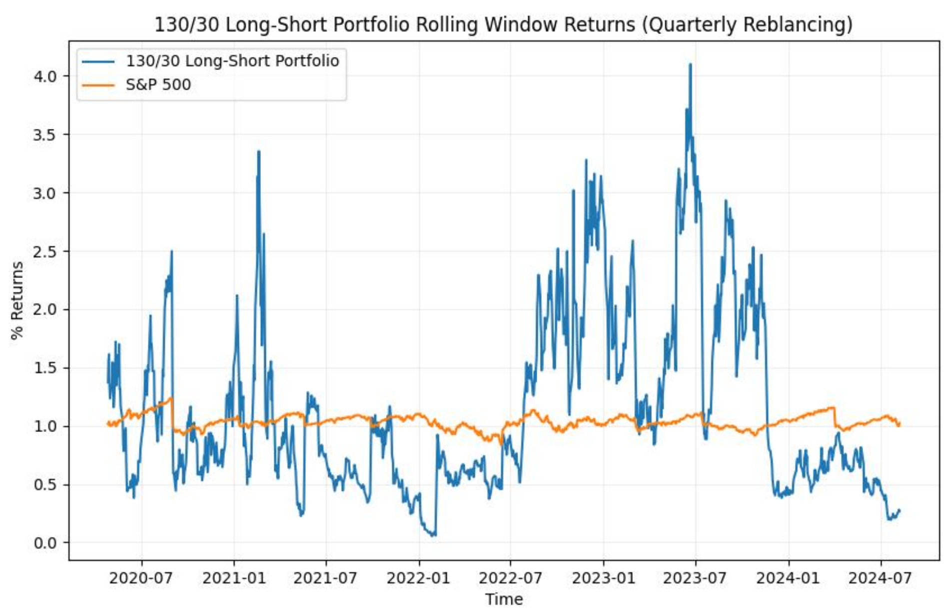

Figure 6: Cumulative returns for the 130/30 portfolio compared to the S&P 500 benchmark. Performance is evaluated using rolling window optimizations.