Senior Capstone Report

Investment Tool: Optimizing Portfolio Allocation with Clustering

Bradford Crosby Washburne Jr [1]

[1] Affiliation 1; Washburne.jr@gmail.com

Abstract

This project presents a portfolio investment tool, focused on optimizing asset allocation through clustering. By leveraging historical stock data, stocks are grouped into distinct clusters to identify patterns in performance and risk profiles. From each cluster, representative stocks are selected to construct diversified portfolios based on three different methods: top average close price, top return, and lowest volatility. From here, these portfolios are evaluated using cumulative return metrics and benchmarked against the S&P 500 and 10-Year Treasury yield to assess performance. The tool aims to assist everyday investors without access to expensive financial advisors by harnessing clustering algorithms to build smarter, more diversified portfolios using only public data.

Keywords: Portfolio Optimization, Clustering Algorithms, Investment Strategy, Machine Learning, Stock Market Analysis

Academic Editor:

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

1. Introduction

Portfolio optimization is a nuanced field that has challenged financial thinkers since the 1950s. Before this period, the goal of the common investor was to simply find a good stock and buy it at the best price – with common sentiment viewing investing as a form of gambling for the wealthy [1]. However, this perception was soon transformed, much to the credit of a 25-year-old economist named Harry Markowitz. Markowitz transformed portfolio optimization, introducing the concepts of risk management to investing. In his 1952 paper *Portfolio Selection*, Markowitz explains that an investor should consider return a desirable thing and variance of return an undesirable thing [2]. This thinking is now popularly coined as Modern Portfolio Theory (or mean-variance analysis), which balances the expected return of an asset and the risk appetite of an investor. The central notion of the MPT states that investment risk can be minimized through diversification or holding a variety of assets whose returns are not highly correlated [3].

While Markowitz laid the foundation for portfolio optimization, his model had two significant drawbacks: the basis of its risk assessment failed to capture many of the nuances in the real world and was not appropriate for the long term. More specifically, the model relies on variance as a measure of risk, which fails to capture non-normal return distributions and real-world noise that contributes to risk. Furthermore, the model does not account for transaction costs, and assumes static returns and risk preferences [4]. These shortcomings have led to many further developments in portfolio optimization – one of which being the Capital Asset Pricing Model (CAPM). Introduced by William Sharpe in his 1964 paper *Capital Asset Prices: A Theory of Market Equilibrium Under*

Conditions of Risk, the model expanded on the MPT by taking a more market-centric view and incorporating the idea of systematic risk [5].

From this work, a plethora of more complex models have been developed in the following decades – mirroring the advancements in computing power and access to data – specifically using machine learning. While powerful, many of these models are reliant on expensive software and data that is not publicly available, creating a gap between the resources of institutions and the common investor. However, this inaccessibility of investing inspired the basis of this project: what if everyday investors could harness clustering algorithms to build smarter, more diversified portfolios using only public data?

This project focuses on three clustering algorithms - K-Means, Self-Organizing Maps, and Fuzzy C-Means – to uncover patterns in historical stock data based on features like close price and trade volume. After clustering, three different techniques are used to select the top performing stocks to build diversified portfolios. These stocks are then compared against the S&P 500 and 10-year Daily Treasury Rate and stressed-tested against volatile markets, including 2008 and 2020 to further evaluate performance. While similar clustering methods have been applied in past research, most focus on single-year returns or complex optimization models [6]. This project instead uses U.S. data across multiple years and emphasizes simple, rule-based portfolio construction. The goal is to show how everyday investors can build resilient portfolios using public data and accessible machine learning tools.

2. Materials and Methods

2.1. Data Collection

To ensure diversification across sector and size, over 120 publicly traded stocks were selected, spanning 11 different industries, and ranging in cap size from large cap (valuation more than $10 billion), medium cap (valuation between $2-$10 billion), and small cap (valuation below $2 billion). Price and volume data was sourced for these companies using AlphaVantage API, which provides daily open, close, high, and low prices as well as trading volume for over 20+ years [7]. In addition to equity data, daily 10-Year U.S. Treasury Rates were obtained from the United States Department of the Treasury to compare with our model as a risk-free rate [8].

2.1. Feature Engineering and Preprocessing

After collecting 20+ years of raw price data, several additional features were engineered. The first of these features is the daily return, or the percentage change in closing price. This feature is the basis for understanding investor gain or loss. Similarly, the second feature, the logarithmic return, was computed as the natural logarithm of the return ratio, which is often preferred in financial modeling due to its ability to be easily added over time periods. From here, 3 more volatility features were computed: the 20-day moving average of the closing price, the 20-day rolling volatility, which measured the standard deviation of the daily returns, and the daily high-low range, which measured the difference between the high and low price, normalized by the opening price for that day.

Before clustering, the dataset was thoroughly cleaned and transformed. First, an initial cleaning of the data inspected for missing values, formatting inconsistencies, and outliers. From here, the sector and market cap information were merged with the historical stock data to create a unified data frame indexed by ticker and time. This main data frame was further broken down into 3 specific years to make 2024, 2020, and 2008 data

frames. These years were selected to represent different market environments: a post-pandemic recovery phase, the COVID-19 pandemic, and the Global Financial Crisis. In terms of further cleaning, tickers with extensive missing data, such as companies that were not traded in 2008 or 2020 —were excluded entirely from the analysis to preserve the validity of year-over-year performance comparisons. For remaining stocks, short gaps in data were filled using forward and backward fill.

The final data frame used for clustering excluded the S&P 500 and encoded both sector and market cap to be used in the clustering. The features were standardized using StandardScaler from sklearn. From here, the models were ready to be fit and stress tested.

2.2. Clustering Algorithms

As previously mentioned, three unsupervised techniques were applied: K-Means, Fuzzy C-Means, and Self-Organizing Maps (SOM). Each of these algorithms have slightly different approaches and parameters, which were all tuned and evaluated using validity metrics.

2.2.1 K-Means

K-Means is a popular clustering algorithm which groups observations into K clusters by minimizing the intra-cluster distance from each point to its centroid. In simpler terms, the algorithm organizes the data into K groups of similar observations, based on their distance from the center of their group. The algorithm first initializes K centroids, then assigns each point to its nearest centroid, and the centroids are updated based on the positions of the points within each cluster. This process repeats until convergence or a maximum number of iterations. For this algorithm, the two main parameters to tune were the number of clusters, and the initialization method.

2.2.2 Fuzzy C-Means

While similar to K-Means, Fuzzy C-Means allows each data point to belong to multiple clusters with different weights of membership. This is particularly useful for grouping data that exhibits traits of more than one cluster. The algorithm assigns each point a probability of belonging to a certain cluster, and the centroids are all updated based on a weighted average of these probabilities for each point. For Fuzzy C-Means, the two parameters to tune were the number of clusters and fuzziness exponent, which controls how much overlap is allowed between clusters.

2.2.2 Self-Organizing Maps

Self-Organizing Maps are slightly different from the first 2 algorithms as they are a type of neural network. SOMs map high dimensional data to a lower dimension, with an input layer representing the features, and an output layer of a 2D grid of neurons representing a cluster in the data. To implement, the weights are randomly initialized, and for each input, SOM aims to minimize the distance between the input and weight vectors. The neuron that minimizes this distance is updated to move closer to the input vector, and this process repeats until convergence. In my project, SOM was implemented from the MiniSom library, which had several parameters that needed to be tuned: the number of clusters; the grid size, which determined the number of neurons; the learning rate,

which determined how quickly weights updated; and sigma, which represented the neighborhood radius.

2.3. Portfolio Construction

For each of the three clustering algorithms, three portfolio strategies were implemented, each based on a different performance metric: average close price, cumulative return, and volatility. As mentioned previously in the introduction, this methodology aligns with similar clustering-based approaches explored in past research. For example, a 2010 paper, *Cluster analysis for portfolio optimization*, applied hierarchical clustering to group assets by correlation, while a 2023 paper, *Portfolio construction with K-means clustering algorithm based on three factors,* applied K-Means using risk and return to improve portfolio performance [6,9]. Within each cluster, the top three stocks were selected based on the respective metric. For the average close price strategy, stocks with the highest mean daily closing price were chosen, which characterizes companies with high valuation. For the top return strategy, stocks were ranked by cumulative return, capturing stocks that had the most growth. For the low volatility strategy, the three stocks with the lowest standard deviation of daily returns were selected, favoring more stable prices.

The goal of using 3 strategies was to compare different investment objectives: value, growth, and risk management. Furthermore, to ensure diversification, each final portfolio was required to have at least 10 unique tickers. In scenarios where there were fewer than 10 – often due to small clusters or overlap – additional tickers were chosen from the biggest clusters until the minimum was met. Once selected these stocks were combined into a single portfolio with equal weighting.

As a result, there were 9 portfolio combinations for each year, with three algorithms and 3 strategies for choosing the stocks from the clusters. Each of these portfolios were benchmarked against the S&P 500 and 10-Year Treasury Rate and included calculations of cumulative returns and the Sharpe Ratio to assess model performance.

3. Results

This section presents the outcomes for the portfolio construction across 2024, 2020, and 2008. This evaluation includes the clustering metrics, the stocks that make up each portfolio, and financial performance.

3.1. Clustering Evaluation

Each algorithm was evaluated based on three metrics: the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz score. These scores were used to tune the algorithms to find the optimal parameter configurations.

3.1.1 K-Means

For K-Means, several configurations were tested by varying the number of clusters and the different initialization methods. The highest observed Silhouette Score was 0.4427 for the algorithm with 12 clusters and random initialization. Furthermore, this configuration achieved the lowest Davies-Bouldin score, and a relatively high Calinski-Harabasz score, indicating the best K-Means model had parameters of k = 12 and init = random.

3.1.2 Fuzzy C-Means

Fuzzy C-Means was again evaluated by varying the number of clusters as well as the fuzziness exponent. The highest observed Silhouette Score was 0.127; however, this was achieved at 2 clusters which would only provide 6 stocks (2 clusters, top 3 chosen from each), thus the minimum number of clusters was determined to be 3. Therefore, we look the next best Silhouette Score, which was 0.033, and achieved with a configuration of 3 clusters and fuzziness = 2.

3.1.3 SOM

SOM was tuned by adjusting the grid size, neighborhood radius (σ) and learning rate. Like Fuzzy C-Means, the highest observed Silhouette Score was again for a configuration with 2 clusters. Thus, the next best Silhouette Score was 0.2978, which was achieved with 3 clusters and a configuration of σ = 0.3 and learning rate = 0.5.

Table 1. Metrics for K-Means clustering.

No. of Clusters	2	3	4	5	6	7	8	9	10	11	12
Silhouette	0.334000	0.136200	0.163800	0.189800	0.187700	0.239700	0.242400	0.259400	0.374800	0.333000	**0.442700**
Davies-Bouldin	1.232700	2.171400	1.957800	1.818400	1.630300	1.683500	1.479000	1.517400	1.458100	1.269900	**1.062100**
Calinski-Harabasz	5456.680400	4578.029800	4175.503300	3673.346700	3531.336600	3284.316500	3558.592600	3393.151300	3504.422600	3887.416600	**3920.821600**
Config	k-means++	k-means++	random	k-means++	random	random	k-means++	k-means++	random	k-means++	**random**

Table 2. Metrics for Fuzzy C-Means clustering.

No. of Clusters	2	3	4	5	6	7	8	9	10	11	12
Silhouette	0.127400	**0.033000**	-0.033200	-0.037500	-0.048900	-0.053100	-0.055500	-0.057700	-0.122700	-0.127100	-0.138800
Davies-Bouldin	2.491500	**2.631700**	4.807600	7.111000	8.468900	7.542900	9.893200	8.728000	7.290400	6.395800	7.176300
Calinski-Harabasz	3937.802100	**1978.326900**	1318.249700	991.123200	795.141100	663.870700	569.345600	498.868500	444.086400	445.262200	446.937300
Config	m=1.5	**m=2.0**	m=1.5	m=1.5	m=1.5	m=1.5	m=1.5	m=1.5	m=1.5	m=1.5	m=1.5

Table 3. Metrics for SOM clustering.

No. of Clusters	2	3	4	5	6	7	8	9	10	11	12
Silhouette	0.431100	**0.297800**	0.103100	0.114300	0.126800	0.166200	0.181600	0.152200	0.164800	0.232500	0.256100
Davies-Bouldin	1.105700	**1.494100**	2.590800	2.360500	2.062300	2.100700	2.202100	2.104700	1.973400	1.815600	1.885100
Calinski-Harabasz	681.663000	**703.298100**	2545.714800	2001.261300	2442.110700	2518.577900	1791.198600	1445.078200	1675.533100	1587.993200	2365.934300
Config	σ=0.3, lr=0.5	**σ=0.3, lr=0.5**	σ=0.3, lr=0.3	σ=0.3, lr=0.3	σ=0.3, lr=0.5	σ=0.3, lr=0.1	σ=0.3, lr=0.3	σ=0.3, lr=0.1	σ=0.7, lr=0.1	σ=0.3, lr=0.5	σ=0.3, lr=0.1

3.1.4 Comparing Clustering Algorithms

When comparing the clustering quality across the three algorithms, K-Means clearly was the best performing model. It achieved the highest Silhouette Score and Calinski Harabasz Score as well as the lowest Davies-Bouldin Score. In comparison, the Fuzzy C-Means algorithm had much weaker performance, with its best configuration achieving a Silhouette Score around 0. This weaker performance likely suggests that the soft

assignments that Fuzzy C-Means allows resulted in poorly defined clusters, compared to the hard clusters imposed by K-Means. Finally, SOM delivered stronger performance than Fuzzy C-Means, but still lagged behind K-Means.

3.2 Portfolio Composition

As seen below, Table 4 outlines the stocks selected for each clustering algorithm across the three portfolio methods: Top Close, Top Return, Low Volatility. K-Means portfolios have the most diversification, with 36 total tickers (12 clusters, three chosen from each cluster). In contrast, both SOM and Fuzzy C-Means have 10 tickers each, which makes sense because of the minimum constraint put in place to ensure 10 tickers (both used 3 clusters, so the maximum selection using the top 3 from each cluster would be 9 tickers).

Furthermore, K-Means portfolio includes a broad mix of sectors and market cap from large-cap tech leaders like Microsoft and Apple, to small-cap companies like American Resources. This diversity further suggests that K-Means was relatively effective at clustering companies into distinct groups. On the contrary, both SOM and Fuzzy C-Means captured much less variation amongst the stocks, in favor of large, well-established firms like Blackrock, Amazon, and Berkshire Hathaway. This strengthens the above validity metrics for the clustering algorithms, suggesting both SOM and Fuzzy C-Means weak clustering resulted in more homogenous portfolios.

Table 4. Tickers in each Portfolio.

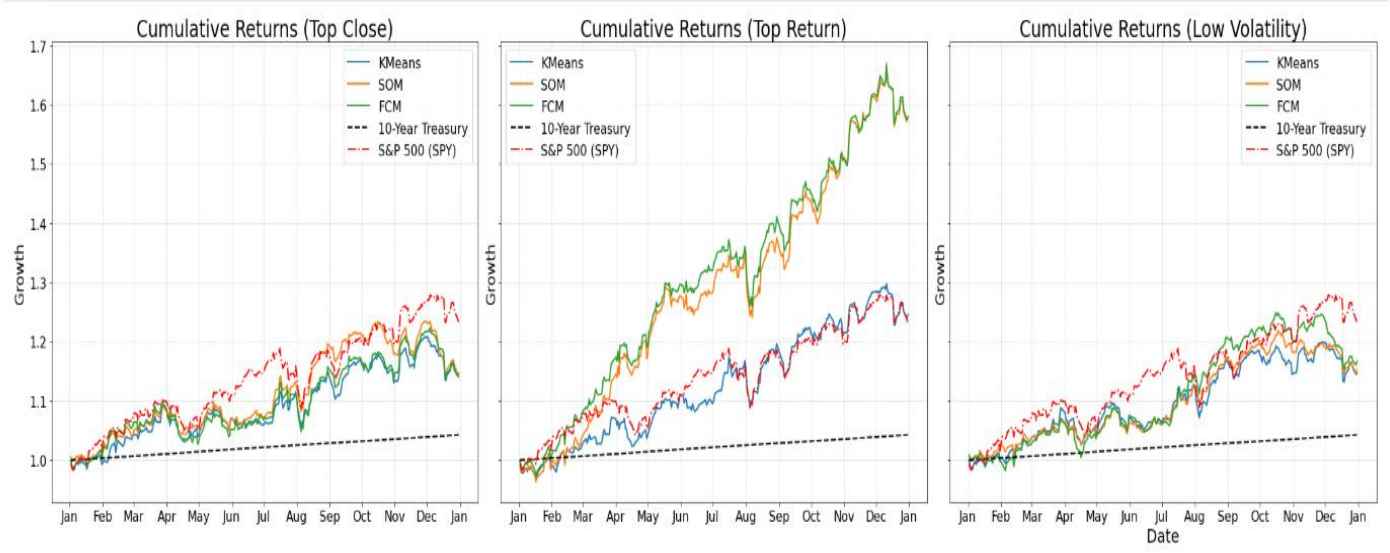
	KMeans	SOM	Fuzzy C-Means
Top Close	SHW, GE, X, ALL, NKE, SYY, SPG, AXS, REX, JNJ, LAMR, STT, LLY, CI, XOM, MSFT, WELL, NRP, MCD, AMZN, SIGI, BRK.B, DE, CLH, AAPL, GOOGL, AFL, BLK, UNH, JPM, WSM, SLAB, AOS, KWR, MAA, ON	BLK, LLY, UNH, ASA, BRK.B, MSFT, DE, SHW, CI, MCD	BLK, LLY, UNH, BRK.B, MSFT, SHW, DE, AMZN, KWR, SPG
Top Return	SHW, HMC, X, ALL, GCO, ASA, AXS, SPG, CSIQ, BAC, STT, LAMR, WMT, LLY, PSMT, DAL, WELL, BRC, AMZN, MCD, SIGI, CLH, BRK.B, RGLD, GOOGL, AAPL, AFL, JPM, BLK, NYT, FUN, SLAB, AOS, CDE, IMAX, NHI	CDE, IMAX, WMT, ASA, AXS, DAL, AMZN, JPM, WELL, GOOGL	IMAX, WMT, AXS, CDE, JPM, GOOGL, DAL, AMZN, WELL, LLY
Low Volatility	SHW, HMC, GE, SYY, ALL, MFA, SPG, AXS, REX, JNJ, STT, LAMR, XOM, CI, PSMT, WELL, MSFT, AVA, MCD, SIGI, AMZN, BRK.B, NGG, DE, KO, AFL, AAPL, BLK, NYT, JPM, AOS, FUN, SLAB, MAA, NHI, CSR	KO, BRK.B, JNJ, ASA, MCD, BLK, WELL, SYY, XOM, MSFT	BLK, WELL, XOM, KO, BRK.B, JNJ, AFL, AVA, LAMR, SPG

3.3 2024 Results

The 2024 performance results, depicted in figure 1 and figure 4, uncover key differences in outcomes based on the stock selection strategy. Looking at cumulative returns, it is apparent that the Top Return strategy clearly outperformed the other strategies – specifically for the SOM and Fuzzy C-Means algorithms, which outperformed the S&P 500 benchmark. In contrast, the Top Close and Low Volatility had more modest results, staying right around the S&P 500 benchmark. Furthermore, the Sharpe ratios in figure 6 further support this trend, as Fuzzy C-Means and SOM Top Return led with the highest ratios, while K-Means Top Return was almost identical to the S&P 500. In any case, all these strategies outperformed the risk-free rate.

Overall, these results suggest that clustering portfolios based on return may be particularly effective during strong markets. In context, the 2024 market saw strong equity gains, and is considered a banner year for U.S. stocks because of the high performance [10].

Figure 1. *Performance for each algorithm by selection strategy in 2024.*



3.4 Stress Testing

To evaluate each portfolio, stress testing was performed using historical data from 2020 and 2008. These years were specifically chosen as examples of volatile market conditions, with the COVID-19 pandemic crashing markets in 2020 and the U.S. housing bubble leading to the Global Financial Crisis of 2008 – the most severe economic downturn since the Great Depression.

3.4.1 2020 Results

In 2020, all portfolios sharply dropped in March, before recovering later in the year. As seen in figure 2 and figure 4, the best strategy was the Top Close Price for the SOM and Fuzzy C-Means algorithms. Furthermore, all of the algorithms using the Top Return strategy outperformed the S&P 500 as well. The only portfolios that did not outperform the S&P 500 during 2020 was K-Means Top Close and all of the Low Volatility combinations.

After the market crash in March, 2020, sentiment shifted towards stocks with high growth and valuation, particularly in technology. In fact, the best performing tickers were from food, healthcare, and software stocks [11]. With this context, it makes sense that the Top Close and Top Return strategies were the best performing as they are biased towards stocks with strong growth and high valuations. Furthermore, as mentioned before, SOM and Fuzzy C-Means are composed with less variety across sector and market cap, and have a higher concentration of these large growth, high valuation companies.

3.4.2 2008 Results

In 2008, all portfolios saw significant drops, with the largest drop starting around mid September. As seen in figure 3 and 4, there were no strategies that produced positive returns on the year. Even so, the Top Return strategy was the best performing, followed by Low Volatility, and finally Top Close. In fact all of the algorithms with each respective strategy outperformed the S&P 500 during 2008.

These results align with the historical context of 2008. One paper by Jozef Barunik found that "the overall intra-market connectedness of U.S. stocks increased substantially with the increased uncertainty of stock market participants during the financial crisis," or in other words, the effectiveness of diversification diminished during the 2008 financial crisis [12]. This explains in part why, although more effective in comparison to other years, the low volatility strategy did not perform as strongly as expected. Regardless, the 2008 stress test highlights the limitations of not only the clustering-based portfolio, but the general volatility of investing and the risk of extreme downturns.

Figure 2. Performance for each algorithm by selection strategy in 2020.

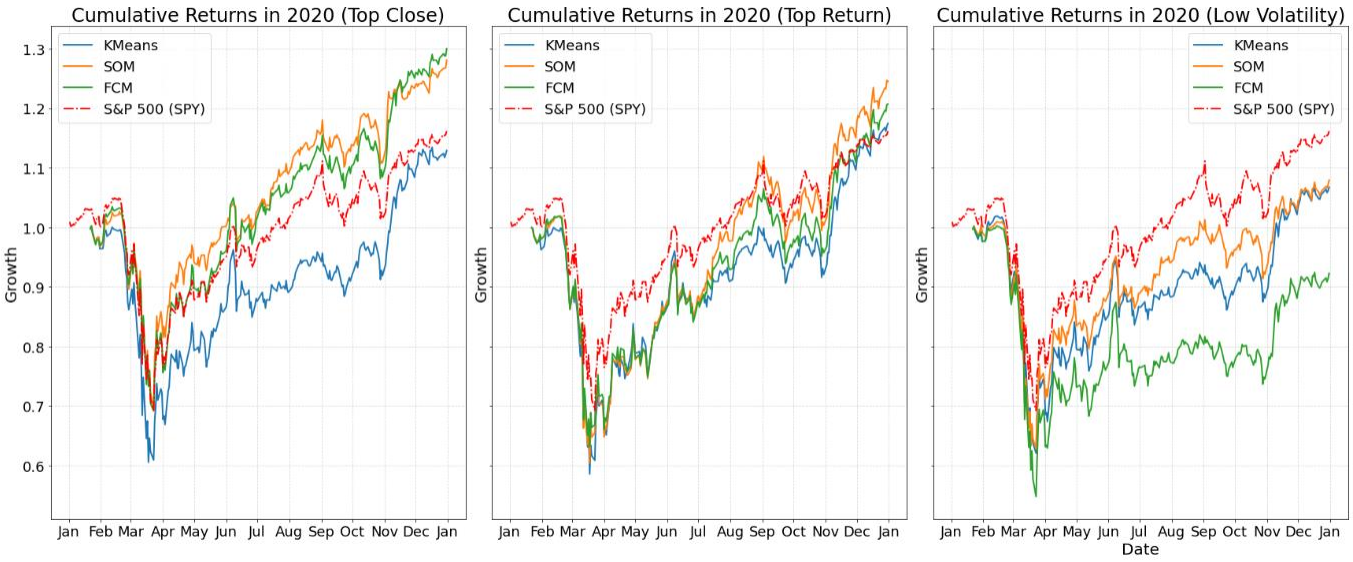


Figure 3. Performance for each algorithm by selection strategy in 2008.

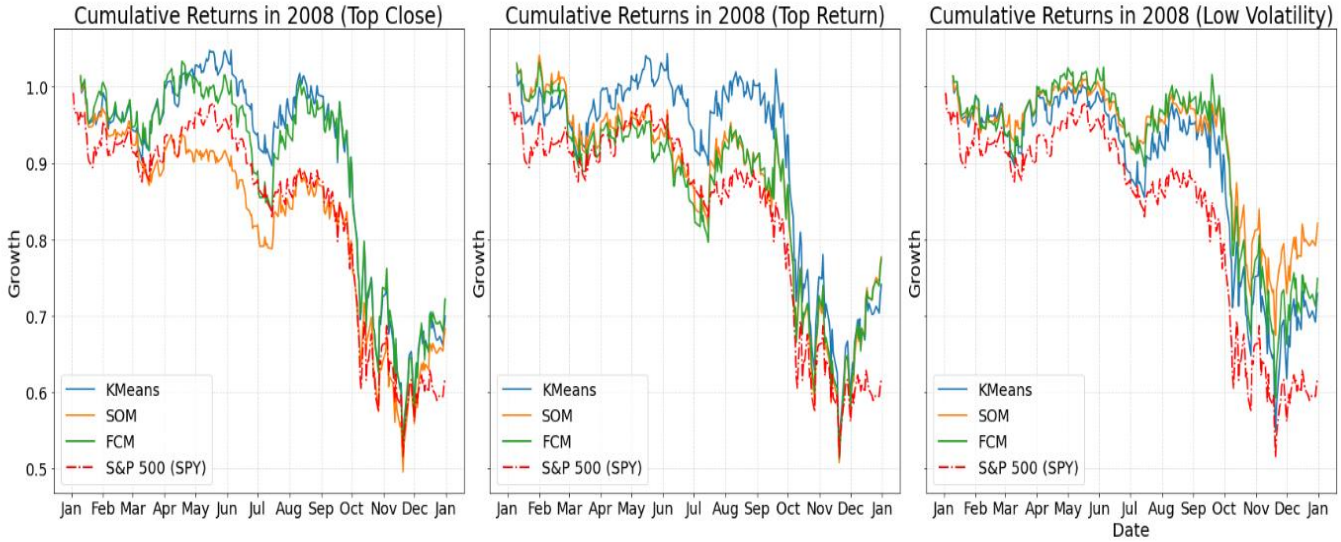
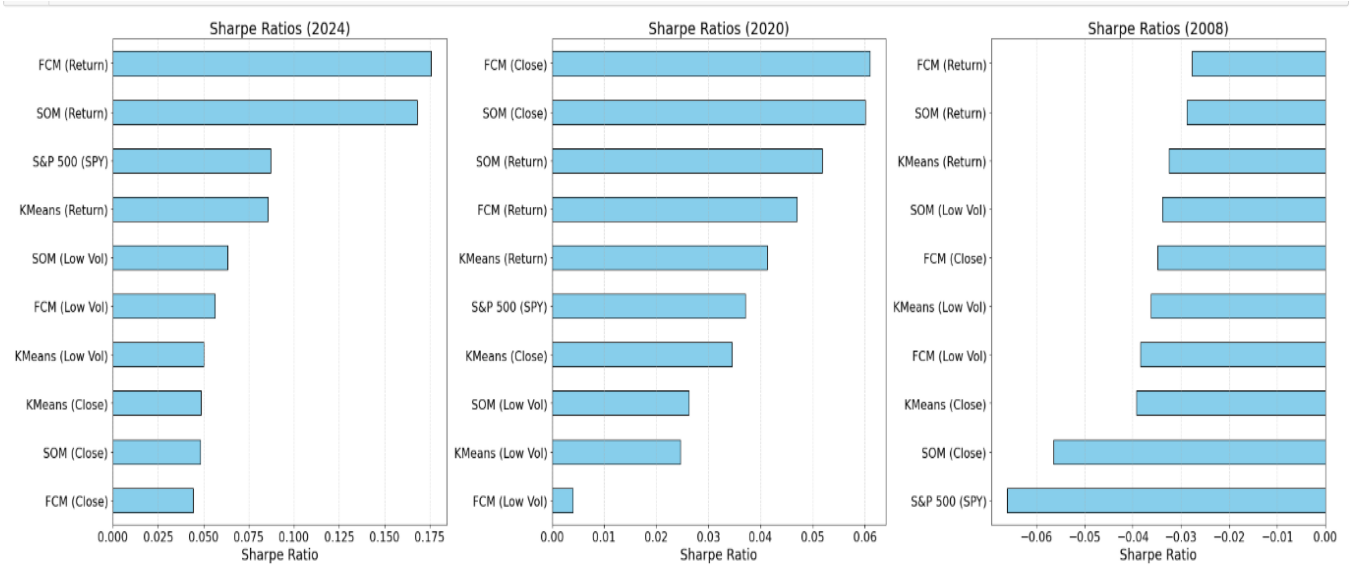


Figure 4. Sharpe Ratio across algorithm and strategy for each year.



3.5 Comparison Between Algorithm and Strategy

To compare different algorithms and strategies, one variable must be held constant. That is, when comparing different algorithms, the strategy is held constant, and when comparing different strategies, the algorithm is held constant.

Across all of the metrics and years, Fuzzy C-Means with the Top Return strategy showcased the strongest performance, with a return of 1.580 and Sharpe Ratio of 0.176 in 2024 – the highest of any combination.

K-Means appeared to be the most consistent algorithm across all of the selection strategies. More specifically, there is less variation between the different strategies for each year for the K-Means portfolios. Although it did not outperform either Fuzzy C-Means or SOM, this supports the earlier assessment of the high diversification within the K-Means portfolios due to the stronger clustering. Even so, it is apparent that the Top Return strategy did outperform both Top Close and Low Volatility in every year for K-Means.

Both SOM and Fuzzy C-Means had very similar performances – given their similar portfolio compositions. In 2024, the Top Return strategy was the best performing, in 2020 the Top Close strategy had the highest returns, and in 2008 Fuzzy C-Means had the best returns with the Top Return strategy while SOM had the best returns with Low Volatility strategy.

The variation in optimal strategy across algorithm and years depicts the influence of market conditions on portfolio performance. Despite their outperformance, both SOM and Fuzzy C-Means had greater variance between strategy choice compared to K-Means. K-Means may never have been a top performer, but it demonstrated more stability across each strategy, which is rooted in the stronger clustering results. This suggests that while effective, SOM and Fuzzy C-Means may require more awareness of market conditions, while K-Means may be a better long-term position.

Overall, these results indicate that the Top Return strategy was generally the most effective across all algorithms. Furthermore, Fuzzy C-Means and SOM provide more returns, yet are more volatile than K-Means.

Figure 5. Returns by strategy for SOM in 2024, 2020, and 2008

	2024 Return	2024 Sharpe	2020 Return	2020 Sharpe	2008 Return	2008 Sharpe
SOM (Close)	1.146	0.048	**1.302**	**0.060**	0.678	-0.059
SOM (Return)	**1.578**	**0.168**	1.270	0.051	0.760	**-0.029**
SOM (Low Vol)	1.152	0.063	1.084	0.025	**0.820**	-0.035

Figure 6. Returns by strategy for Fuzzy C-Means in 2024, 2020, and 2008

	2024 Return	2024 Sharpe	2020 Return	2020 Sharpe	2008 Return	2008 Sharpe
FCM (Close)	1.141	0.044	**1.329**	**0.061**	0.715	-0.036
FCM (Return)	**1.580**	**0.176**	1.234	0.047	**0.758**	**-0.028**
FCM (Low Vol)	1.167	0.056	0.933	0.004	0.744	-0.039

Figure 7. Returns by strategy for K-Means in 2024, 2020, and 2008

	2024 Return	2024 Sharpe	2020 Return	2020 Sharpe	2008 Return	2008 Sharpe
KMeans (Close)	1.146	0.049	1.152	0.035	0.697	-0.041
KMeans (Return)	**1.245**	**0.086**	**1.204**	**0.042**	**0.735**	**-0.034**
KMeans (Low Vol)	1.147	0.050	1.082	0.025	0.728	-0.038

4. Discussion

4.1 Key Takeaways

There are 3 main takeaways to highlight from this analysis. First, survivorship bias was an inherent limitation with stress testing. Because there was missing data from 2008 and 2020, the final portfolios were constructed with stocks from companies that

survived these markets and traded through 2024. However, this filtering likely overestimated growth by biasing more historically resilient companies. Secondly, Fuzzy C-Means and SOM outperformed K-Means in almost every instance – despite significantly lower clustering results – highlighting the importance of strategy-algorithm alignment in short-term optimization. Lastly, return-based portfolios almost always outperformed the other strategies.

4.1 Future Work

In terms of future work, there are several areas for improvement given extended time and resources. For one, sourcing more features – specifically company specific data like revenue, profit margins, etc. – would likely increase the effectiveness of clustering. One of the main limitations of my data is its lack of information on market conditions and company metrics. Another improvement is to implement a more advanced method of handling missing data, such as imputation or dynamic reweighting, instead of simply excluding these stocks. Lastly, to better reflect real-world conditions, future models should incorporate transaction costs, and other asset classes, further enhancing the applicability of the tool.

5. Conclusions

Overall, Fuzzy C-Means and SOM delivered the strongest performance when paired with return-based strategies, outperforming K-Means even when their clustering metrics appeared weaker. This highlights the importance of aligning strategy with algorithm rather than relying solely on technical scores. K-Means, while more stable, was generally outpaced by SOM and Fuzzy C-Means in raw returns – especially in strong markets.

However, these results are not without limitations. Survivorship bias and the exclusion of delisted or missing tickers may have overinflated results. Furthermore, the portfolio construction ignores the reality of transaction costs, liquidity constraints, and general macro-economic conditions. To build a more robust tool, future work should incorporate sourcing more financial features, handling missing data without simply eliminating it, and simulating real-world conditions.

More broadly, these results highlight the potential of using machine learning models to improve portfolio construction in an accessible way. With just public data and open-source tools, investors can still build well-performing, diversified portfolios that compete with market benchmarks. In a world dominated by large institutions, clustering may not completely level the playing field, but it brings the average investor much closer.

References

1. Beattie, Andrew. (2025, August 25). *Understanding the History of the Modern Portfolio.* Investopedia. https://www.investopedia.com/articles/07/portfolio-history.asp#citation-2.
2. Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x
3. Fabozzi, Frank J., Francis Gupta, and Harry M. Markowitz. (2002) The legacy of modern portfolio theory. *The journal of investing*, 11(3), 7-22.
4. Zanjirdar, Majid. (2020, July 09). Overview of portfolio optimization models. *Advances in mathematical finance and applications*, 5(4), 419-435.
5. Sharpe, W.F. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *The Journal of Finance*, 19, 425-442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x
6. Tola, Vincenzo, Fabrizio Lillo, Mauro Gallegati, and Rosario N. Mantegna. (2010). Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control*, 32(1), 235-258.
7. Alpha Vantage. (n.d.). *Alpha Vantage API documentation*. https://www.alphavantage.co/documentation/
8. U.S. Department of the Treasury. (2024). *Daily treasury yield curve rates*. https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value=2024
9. Aslam, B., Bhuiyan, R. A., & Zhang, C. (2023). Portfolio construction with K-means clustering algorithm based on three factors. *MATEC Web of Conferences, 377*, 02006. https://doi.org/10.1051/matecconf/202337702006
10. Sonders, L. A. (2025, January 6). *It was a very good year*. Schwab Brokerage. https://www.schwab.com/learn/story/it-was-very-good-year
11. Mazur, M., Dang, M., & Vega, M. (2021). COVID-19 and the March 2020 stock market crash: Evidence from S&P 1500. Finance Research Letters, 38, 101690. https://doi.org/10.1016/j.frl.2020.101690
12. Barunik, J., Kocenda, E., & Vacha, L. (2013). Asymmetric connectedness of stocks: How does bad and good volatility spill over the US stock market?. *arXiv preprint arXiv:1308.1221*.