Modern Portfolio Optimization using Artificial Intelligence and Machine Learning

Sasha Olinde Vergara

April 30, 2025

1 Introduction

Investing in the stock market can seem a formidable task for anyone just getting started. We are constantly bombarded on television and social media by stories of the stock market's erratic behavior that often leads to personal fortunes either blossoming exponentially or dissolving in an instant.

Many investors are *passive* and simply take the advice of financial planners without expending any effort in understanding the complexities of investing. For example, many employers offer 401(k) retirement plans for which employees simply do whatever the company's mutual fund manager recommends. Other investors are more *active* by strategically allocating a chosen set of equities or assets to try to maximize their returns. Perhaps they have heard of recent growth stocks such as Tesla, Meta, or Nvidia and want to "get in on the action". But how does one actually decide which stocks to buy and, more importantly, how much of each stock should they purchase? This is the basic definition of a portfolio and this is the problem addressed in this project.

Portfolio Optimization is a fundamental approach used by many investors who are looking for an advantage in their stock allocation decisions in support of their financial goals. Perhaps they are trying to increase their net worth to enhance their lifestyle, buy a new house, pay for college for their children, or simply save enough money to retire comfortably. The aim of any investor is to balance the financial risk against their returns. The main difficulty is striking the appropriate balance between possible returns and an acceptable degree of risk. To this end, Portfolio Optimization encompass many basic concepts in mathematics, probability and statistics, linear and non-linear optimization theory, stochastic processes and time series, and of course specific investment domain knowledge. Of note is Harry Markowitz's **Modern Portfolio Theory** (**MPT**) where he pioneered an approach that maximizes returns while controlling risk. He proposed the notion of an **Efficient Frontier** which defines portfolios that provide the best expected return for a given degree of risk. Markowitz turned portfolio construction into a more methodically quantitative process by applying measures that include expected return, variance, and covariance of asset returns which captured *risk* or *volatility.* Though Markowitz's MPT is fundamental to portfolio theory, its shortcomings have been made apparent in its use since it was introduced in 1952. Real-world investing introduces complexity that MPT does not entirely address, including transaction costs, taxes, and borrowing restrictions. Portfolio risk is also influenced by the interactions or correlations among assets in the portfolio and not just by individual asset risks.

Today, the general public is increasingly becoming aware that **Artificial Intelligence** (**AI**) and **Machine Learning** (**ML**) are not just for academic or research scientists but are becoming more mainstream and taking a strategic role in many industries, including the investment industry. As the financial markets evolve and new challenges/constraints emerge, investors are turning to AI/ML to harness their increasing problem-solving abilities. AI/ML is being applied as a set of powerful tools for managing complexities of portfolio optimization. Unlike conventional models, which depend on assumptions about asset return distributions, AI/ML methods can find non-linear connections within data that can lead to improved asset selection, risk forecasting, and advanced portfolio building techniques.

This project looks to build a strong framework for asset allocation in a constantly changing financial scene by combining classical economic theory with modern computational methods. We evaluate conventional and machine learning-based portfolio optimization techniques through theoretical analysis, mathematical modeling, and empirical investigation. The aim is to offer a thorough knowledge of the advantages and drawbacks of every approach, to develop more adaptive and effective portfolio management techniques.

The project assesses the effectiveness of these approaches on two separate stock groups: the **Magnificent Seven** (**MAG7**) and the **Dow Jones 30** (**DJ30**) [3],[1]. We will be using financial metrics including the Sharpe ratio, factor models, and advanced optimization techniques. These stock groups offer different grounds for comparison, representing stable *blue-chip* stocks and *high-growth* technology companies.

This project provides a brief introduction to the historical foundations of portfolio optimization, the mathematical and statistical frameworks behind modern portfolio optimization strategies, the implementation of these optimal strategies in a modern programming language (Python), and the integration of a few prevailing AI/ML techniques to potentially enhance these well-known portfolio optimization strategies. Specifically, in addition to the baseline modern approaches such as the Equal-Weighted or Naïve passive portfolio strategy, the Maximum Sharpe Ratio portfolio, and Minimum Volatility portfolios described below, in this project, we will design optimal portfolio strategies for different asset classes using a **supervised** approach (based on PCA), **unsupervised** approach (based on clustering), and a **Reinforcement Learning** (**RL**) algorithm approach.

2 Background: The Evolution of Financial Theory

Developed over a century, modern financial theory traces back to the early 1900s with Louis Bachelier's pioneering work. In his dissertation, *The Theory of Speculation*, Bachelier argued that it is impossible to definitely project future prices. Though mostly overlooked at the time, his ideas later shaped the Efficient Market Hypothesis (EMH), which holds that markets use all the information at hand to determine prices, so challenging investors' consistent market beating ability.

Alfred Cowles III investigated professional investors' ability to choose winning stocks in the 1930s and 1940s and discovered little proof they could regularly beat the market. His studies confirmed that known information shapes stock prices, which bolstered the case for passive investing rather than actively trying to "beat the market". Benjamin Graham, stressed an approach to identify undervalued stocks and set the foundation for value investing around the same time. Graham's techniques would come to have impact on significant investors like Warren Buffet. Then 1952 brought a revolution when Harry Markowitz presented Modern Portfolio Theory (MPT) in his paper *Portfolio Selection*. Markowitz demonstrated the link between risk and return, so demonstrating the need of diversification for control of risk. Using a mathematical method for choosing portfolios that maximize return for a given level of risk, he created **Mean-Variance Optimization** (**MVO**).

First, note that for a given portfolio, defined by a weight vector w a set of n assets, the portfolio return is the weighted sum of the individual asset returns:

$$R_p = \sum_{i=1}^{n} w_i R_i \tag{1}$$

where R_i is the return of asset i. When each asset has an expected returns $E[R_i] = \mu_i$, the **expected return** of the portfolio is:

$$E[R_p] = \mu^T w \tag{2}$$

and the **variance** of the portfolio is given by:

$$\sigma_p^2 = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + 2\sum_{i<j} w_i w_j \mathrm{Cov}(R_i, R_j) = w^T \Sigma w$$

where σ_i^2 is the variance of asset i and $\mathrm{Cov}(R_i, R_j)$ is the covariance between assets i and j which is Σ_{ij}.

Given these definitions, Markowitz's MVO model is formally expressed as a *linear optimization problem* with a *quadratic constraint*:

$$\max_{w} f(w) = \mu^T w - \alpha w^T \Sigma w \tag{3}$$

subject to:

$$\sum_i w_i = 1, \quad w_i \geq 0, \quad \forall i \tag{4}$$

where w denotes the proportion of assets (or *weights*) in the portfolio, μ is the vector of expected returns, Σ is the covariance matrix of asset returns, and α (the *Lagrange Multiplier*) is known as the *risk-aversion* parameter [5].

The **Sharpe ratio** (**SR**) compares the portfolio's expected excess return to the volatility of this excess return, measured by its standard deviation. It measures the compensation as the average excess return per unit of risk taken as is defined as:

$$S = \frac{\mu^T w - R_f}{\sqrt{w^T \Sigma w}} \tag{5}$$

where R_f is the risk-free rate. An optimal portfolio that maximizes S would be considered to balance risk and expected return [5].

By developing so-called economic market **factor models**, one could decompose an asset's returns into **systematic** (inherent to the whole market itself) and **idiosyncratic** (inherent to only the asset itself) components. Introduced by **William Sharpe** in the early 1960s, the **capital asset pricing model** (**CAPM**) expands on MPT by adding **Beta** (β), a gauge of the sensitivity of an asset to market fluctuations. Under CAPM, the expected return of an asset is modeled as:

$$E[r_i] = r_f + \beta_i(E[r_m] - r_f) \tag{6}$$

where r_f represents the risk-free rate, r_m denotes the market return, and β_i quantifies the sensitivity of the asset to changes in the wider market. CAPM contends that asset returns are driven by a single risk factor attributed as market risk. Although the CAPM projects a **single risk factor** (market risk), actual stock returns are shaped by several risk factors. Investors analyzing vast amounts of historical return data recognized several factors influencing real-world stock returns outside of just random market swings.

In 1993, Eugene Fama and Kenneth French enhanced CAPM by including size and value elements, so enhancing the justification of stock return variances. **Eugene Fama** and **Kenneth French** developed a **three-factor model** adding **size and value factors** to better explained cross-sectional stock returns. The Fama-French three factor model is formulated as:

$$E[r_i] - r_f = \alpha_i + \beta_{MKT}\left(E[r_m] - r_f\right) + \beta_{SMB} \cdot SMB + \beta_{HML}. \tag{7}$$

The size premium is represented by SMB (Small Minus Big), which implies that small-cap stocks tend to perform better than large-cap stocks, and the value premium is represented by HML (High Minus Low), which indicates that value stocks typically generate higher returns than growth stocks. By taking into consideration more than just market exposure, these extensions offered a more sophisticated method of risk management and portfolio construction [6]. By taking into consideration a number of variables other than market exposure, these models offered a more sophisticated approach to risk management and diversification.

Using supervised learning and graph theory to group similar assets, **Hierarchical Risk Parity** (**HRP**) presents a more resilient substitute for conventional mean-variance optimization [2]. HRP is less prone to noisy returns and market volatility than more traditional MPT techniques since it does not depend on a direct covariance matrix estimate. Rather, it generates more steady, well-diverse portfolios using hierarchical tree-based clustering methods. Generally speaking, network-based portfolio optimization techniques seem to beat conventional risk management techniques since they offer more resilience and flexibility in evolving financial environment.

Lately, developments in AI/ML have advanced portfolio optimization even more drastically. These days, risk prediction and asset choice are improved using techniques including regression analysis, time series forecasting (akin to ARIMA, LSTMs), RL, and clustering algorithms.

It is expected that as financial markets grow more complicated the integration of quantitative finance, optimization strategies, and AI-driven models will transform portfolio allocation. Future developments in artificial intelligence, RL, and real-time market analytics are expected to make portfolio optimization even more flexible, thus allowing investors to navigate ever-changing market conditions.

3 Portfolio Optimization Techniques

The main purpose of portfolio optimization is to allocate assets such that the expected return is maximized for a certain degree of risk. Or, conversely, risk is minimized for a given level of return. Investors try to optimize the performance of their portfolio while controlling their risk exposure. In this project, we use several portfolio optimization techniques, each providing a different perspective on portfolio building, and investigate their performance.

In this section, we will look at various ways to calculate portfolio weights, including conventional approaches like the (passive) **Equally Weighted** (or **Naive**) portfolio, the **Maximum Sharpe Ratio** portfolio, and the **Minimum Volatility** portfolio [6]. These methods are often used as *benchmarks* in portfolio optimization, and each reflects a particular strategic approach to handling the trade-offs between risk and return. We also include more complex and empirical techniques like **Principal Component Analysis** (**PCA**) and **Hierarchical Risk Parity** (**HRP**), which provide advanced methods for building portfolios utilizing dimensionality reduction and hierarchical clustering [4].

We will also use the Capital Asset Pricing Model (CAPM) and the Fama-French Three-Factor model to help the optimization process by offering insightful analysis of the systematic risk factors driving asset returns. Including these multi-factor models allows us to consider an extra source of risk outside conventional market exposure, enhancing the precision of our expected return computations and enabling more comprehensive portfolio selection [6].

These approaches offer a strong portfolio optimization framework that considers several risk-return profiles and applies both classical and new financial theories. This section provides a mathematical foundation and application of all these approaches.

3.1 Monte Carlo Simulations

In order to visualize the portfolio *feasibility space* and the *Efficient Frontier* (which is simply the upper boundary of the feasibility space), we will construct a Monte Carlo simulation to help in evaluating each of our benchmarks' performances. We will create a large number of portfolio weight vectors w with various random asset allocations and then calculate the corresponding expected returns, volatility, and Sharpe ratio for each of the portfolios. Once each corresponding optimal benchmark portfolio is determined, each benchmark's portfolio's position in the feasibility space will be plotted along with the Monte Carlo set of random portfolios. This plot will provide the reader an excellent view of the expected return and risk associated to each portfolio [6].

3.2 Equally-Weighted or Naive Portfolio

The Equally-Weighted or Naive Portfolio is one where each asset receives the same allocation percentage or weights. In this case, the weight w_i for each asset i is simply:

$$w_i = \frac{1}{n} \tag{8}$$

where n is the number of assets in the portfolio.

3.3 Maximum Sharpe Ratio Portfolio

The Max Sharpe Ratio Portfolio aims to determine which portfolio has the highest risk-adjusted return. The Sharpe ratio measures the excess return that an investor received for each unit of risk assumed [5]. The Sharpe ration (S) of a portfolio can be calculated using the following formula:

$$S = \frac{E[R_p] - R_f}{\sigma_p} \tag{9}$$

where $E[R_p]$ is the expected return of the portfolio, R_f is the risk-free rate (which is often assumed to be zero), and σ_p is the standard deviation (volatility) of the portfolio's return.

3.4 Minimum Volatility Portfolio

The Minimum Volatility Portfolio seeks to reduce the risk (volatility) of the portfolio. The goal in this situation is to locate the group of portfolio weights $w = (w_1, w_2, \ldots, w_n)$ that minimizes the portfolio variance:

$$\text{Minimize } \sigma_p^2 = w^T \Sigma w \tag{10}$$

where Σ is the covariance matrix of the asset returns. The constraint is that the sum of the weights must equal 1:

$$\sum_{i=1}^{n} w_i = 1 \tag{11}$$

The Monte Carlo simulation will create random portfolios with different weights, computes their volatilities, and then one can easily determine the portfolio corresponding to the least volatility. It will be clear from the feasibility plot that the portfolio with the least risk will sit at the leftmost point on the Efficient Frontier.

3.5 Portfolio Feasibility Space and Efficient Frontier

The Monte Carlo simulations allow one to plot the Efficient Frontier, which is represented by the upper boundary curve of the highest return for every degree of risk. The Efficient Frontier is the curve where investors can choose portfolios with the best trade-offs between risk and return. Portfolios below the efficient frontier are deemed "suboptimal" since they provide lesser returns for the same risk as portfolios on the frontier. The Efficient Frontier is represented by the blue dashed line in the plot of the Monte Carlo simulations which also shows the scatter of the randomly produced portfolios points representing the feasibility region. Overall, portfolio optimization using Monte Carlo simulations is a strong approach for visualizing the feasibility space, Efficient Frontier, and where in this space optimal portfolios lie. Every one of the optimization techniques outlined—Max Sharpe Ratio, Equally Weighted, and Minimum Volatility—provides a unique approach to asset allocation and by using these simulations, the investor can select a portfolio that most closely fits their risk tolerance and return goals.

3.6 Capital Asset Pricing Model (CAPM)

Introduced by Sharpe (1964), the Capital Asset Pricing Model (CAPM) is a basic one-factor model that predicts an asset's expected return depending on its correlation with the general market. The CAPM holds that the only risk that cannot be diversified away is the market risk (systematic risk), which is reflected by the return of the market portfolio. Where the asset's sensitivity to the market is reflected by the beta coefficient, the model offers a linear connection between the return of an asset and the market return.The CAPM can be expressed as:

$$E(r_i) = r_f + \beta_i \left(E(r_m) - r_f\right) \tag{12}$$

Where $E(r_i)$ is the expected return on asset i, r_f is the risk-free rate, $E(r_m)$ is the expected return on the market portfolio, and β_i is the asset's beta (a measure of the asset's sensitivity to the market).

The beta coefficient β_i measures the asset's risk in relation to the market factor. If $\beta_i = 1$, the asset moves in line with the market. If $\beta_i > 1$, the asset is more volatile. If $\beta_i < 1$, the asset is less volatile. The beta coefficient is estimated using historical return data via a linear regression. The beta is the slope of the regression line when the asset's returns are regressed against the market's returns:

$$r_i = \alpha + \beta_i r_m + \epsilon \tag{13}$$

where r_i is the return of asset i, r_m is the return of the market, α is the intercept (known as *Jensen's alpha*), and ϵ is the error term. The beta coefficient is the ratio of the covariance between the asset and the market returns to the variance of the market returns:

$$\beta_i = \frac{\operatorname{cov}(r_i, r_m)}{\operatorname{var}(r_m)}$$

Portfolio optimization depends on the CAPM since it provides a means to evaluate the anticipated returns depending on an asset's systematic risk exposure to the market. Understanding how an asset moves with the market helps investors build portfolios that maximize the trade-off between risk and return. Often, the market portfolio—made up of all accessible assets—serves as a benchmark for assessing the performance of single assets and portfolios.

Moreover, CAPM can help choose assets with the best risk-adjusted returns and offer knowledge on capital costs. But, CAPM is constrained by its dependence on one market element and its assumption of market efficiency, which holds that only systematic market risk is essential. Clearly, there exist other ecomomic factors/drivers for portfolio performance than simply the market and the next section addresses this issue.

3.7 Fama-French Three-Factor Model

Proposed by Fama and French (1993), the Fama-French Three-Factor Model adds two more elements—the size factor (SMB) and the value factor (HML)—to the CAPM, extending it. The model was created to solve empirical CAPM anomalies, including the finding that, over time, small-cap stocks and value stocks tend to outperform large-cap and growth companies. The Fama-French Three-Factor Model expands the CAPM by including the size and value factors:

$$E(r_i) - r_f = \alpha_i + \beta_{MKT} \left(E(r_m) - r_f\right) + \beta_{SMB} \cdot SMB + \beta_{HML} \cdot HML$$

Where $E(r_i) - r_f$ is the excess return of asset i (the return above the risk-free rate), SMB (Small Minus Big) is the size factor that captures the difference in returns between small-cap and large-cap stocks, HML (High Minus Low) is the value factor that captures the difference in returns between value stocks (high book-to-market ratio) and growth stocks (low book-to-market ratio), α_i is the intercept term, which represents the asset's alpha or abnormal return.

The Market Factor (MKT) are the return on the market portfolio minus the risk-free rate, similar to the CAPM. The Size Factor (SMB) are the excess return of small-cap stocks over large-cap stocks. Historically, smaller companies tend to have higher returns, albeit at higher risk. The Value Factor (HML) are the excess return of value stocks over growth stocks. Value stocks, which have a high book-to-market ratio, often outperform growth stocks over the long term.

The Fama-French Three-Factor Model is estimated through multiple linear regression. The excess returns of an asset or portfolio are regressed against the three factors: market returns, size (SMB), and value (HML). The coefficients β_{MKT}, β_{SMB}, and β_{HML} represent the asset's exposure to each factor.

Useful in portfolio optimization, the Fama-French Three-Factor Model offers a more complex perspective on asset returns than CAPM. The model catches additional risk not explained by the market alone by including size and value factors. The capacity to include several elements lets one manage risk more effectively and forecast asset returns more precisely.

The model lets investors determine which assets are vulnerable to these stated risk factors—e.g., small-cap stocks or value stocks—and build factor-adjusted portfolios. Furthermore, by clarifying how exposure to size and value elements affects asset pricing, the model can help one grasp the cross-section of anticipated returns.

3.8 PCA and Eigenvalue Decomposition

PCA is a dimensionality reduction technique that transforms correlated asset returns into fewer uncorrelated components. These can be used to derive the principal components of a portfolio. PCA finds the directions (principal components) that maximize data variance; each corresponds to an eigenvector and eigenvalue. Regarding financial portfolios, the data reflects the returns of a group of assets. PCA helps us keep the most important characteristics—those elements that account for the most variance in asset returns—while lowering the data's dimensionality. These main components can then be used to build portfolios matching the latent empirical risk factors that explain the most volatility in the return data of the chosen set of assets.

PCA starts with calculating the covariance matrix S of the asset returns. The covariance matrix shows the relationships between the assets, suggesting how their returns co-vary over time. The *sample covariance matrix* S is computed as follows:

$$S = \frac{1}{N} \sum_{n=1}^{N} (R_n - \bar{R})(R_n - \bar{R})^T$$

where R_n is the return vector for the n-th observation, $\bar{R}$ is the mean return vector, and N is the number of observations. The sample covariance matrix S is symmetric and captures the variance of each asset along the diagonal and the covariances between pairs of assets off the diagonal.

Next, we perform an eigen decomposition of the sample covariance matrix S. This step is crucial because PCA seeks to identify the directions (or principal components which correspond to the eigenvectors of S, of maximum variance (corresponding to the cumulative sum of the eigenvalues) in the returns. Mathematically, we solve the following eigenvalue problem:

$$Su_i = \lambda_i u_i$$

where u_i is the i-th eigenvector and λ_i is the corresponding eigenvalue. The eigenvectors represent the directions in the data space along which the variance is maximized, while the eigenvalues represent the amount of variance explained by each eigenvector. The eigenvectors are orthogonal, meaning they are uncorrelated, which is a key property of PCA.

The first eigenvector u_1 corresponds to the direction of maximum variance in the data. In contrast, the second eigenvector u_2 is orthogonal to u_1 and corresponds to the next most significant variance, and so on. The eigenvalue λ_1 linked to u_1 reflects the degree of variance caught by the first principal component; likewise, λ_2 relates to the second principal component.

PCA is used in portfolio optimization to identify the key elements influencing asset returns. These elements may be applied to create so-called eigen-portfolios, portfolios from the principal components. Serving as the foundation for latent empirical risk factors in the portfolio, the first principal component explains the most variance in asset returns. The eigenvalues show the degree of the variance each principal component accounts for, therefore reflecting the relevance of every principal component in characterizing the behavior of the data.

Mathematically, the portfolio weight for the i-th asset can be represented as the components of the eigenvector corresponding to the first principal component. The weight w_i for asset i is proportional to the i-th component of u_1 if the eigenvector u_1 denotes the first principal component. Therefore, by normalizing the parts of the eigenvector u_1, the portfolio weights can be known, therefore guaranteeing that the total of the weights is one:

$$w_i = \frac{u_{1i}}{\sum_{i=1}^{D} |u_{1i}|}$$

where u_{1i} is the i-th component of the first eigenvector u_1, and D is the number of assets. This method results in a portfolio that is highly sensitive to the direction of maximum variance in the data, which is typically the most important risk factor.

Essentially, the main component analysis may be applied to several dimensions. If the goal is to project the data onto a M-dimensional subspace, where $M < D$, the portfolio weights can be constructed by selecting the eigenvectors corresponding to the largest M eigenvalues. The portfolio is built as a combination of the principal components, with the weights given according to the size of the eigenvalues, these eigenvectors defining the principal subspace.

The general approach for determining portfolio weights using multiple principal components involves the following steps:

1. Compute the covariance matrix S of the asset returns.

2. Perform eigenvalue decomposition on S to obtain the eigenvectors $u_1, u_2, \ldots, u_M$ corresponding to the largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_M$.

3. Construct the portfolio weights by normalizing the eigenvectors $u_1, u_2, \ldots, u_M$ and combining them based on their respective eigenvalues.

Larger eigenvalues cause higher weights for the relevant components in the weighted mix of the principal components, therefore determining the final portfolio weights. This produces a portfolio that shows the most important elements influencing the returns of the assets.

3.9 Hierarchical Risk Parity

This part examines the HRP method from a mathematical point-of-view], focusing on its use in portfolio optimization. Introduced by López de Prado (2016), HRP seeks to solve problems including volatility clustering, risk concentration, and instability in conventional portfolio optimization techniques, such as mean-variance optimization [2]. By combining hierarchical clustering with risk parity ideas, HRP improves portfolio building to produce robust and varied portfolio allocations.

The three primary phases of HRP are hierarchical asset clustering, quasi-diagonalization of the covariance matrix, and recursive bisection of the portfolio into sub-portfolios. The following parts explain the mathematical theory underlying each phase and how they support the general optimization process.

HRP's first stage is hierarchical asset clustering using their correlation matrix. This stage aims to group assets that behave similarly, so lowering the number of pairwise correlations to be taken into account. A distance matrix generated from the correlation matrix starts the clustering process. The distance between two assets i and j is calculated as:

$$d_{i,j} = \sqrt{\frac{1}{2}(1 - \rho_{i,j})}$$

Where $\rho_{i,j}$ is the correlation between assets i and j. A linkage criterion, namely single linkage clustering, is then applied to this distance measure to carry out hierarchical clustering. Single linkage clustering defines the distance between two clusters as the least pairwise distance between their parts. The assets or clusters that are most similar are combined as the clustering moves forward; the process goes on until all assets are in one cluster.

A dendrogram visualize the clustering process by means of a hierarchical structure showing asset relationships. Capturing the whole merging process, the linkage matrix holds information on which assets or clusters were combined and the distances separating them. This hierarchy is significant since it shows the natural groupings of assets, therefore guaranteeing that assets inside the same cluster are viewed as complementary instead of interchangeable.

The covariance matrix's quasi-diagonalization follows asset clustering. The covariance matrix Σ contains the covariances between the returns of assets in the portfolio. We reorder the rows and columns of the covariance matrix to position assets belonging to the same cluster closer to the diagonal, hence improving the efficiency of the portfolio allocation. A permutation matrix P is used to carry out this reordering, which produces a new covariance matrix Σ_{QD}:

$$\Sigma_{QD} = P \Sigma P^T$$

Here, Σ_{QD} is the quasi-diagonalized covariance matrix, and P is the permutation matrix derived from the hierarchical clustering. While assets with lesser correlations are placed further away, this reordering puts highly correlated assets (within the same cluster) close to each other.

Quasi-diagonalization's main objective is simplifying the covariance matrix for the following HRP stage, inverse-variance allocation. With assets inside the same cluster having their covariance values positioned closer to the diagonal, the reorganization of the covariance matrix guarantees that the covariance structure reflects the clustering outcomes and helps to lower the complexity of the portfolio optimization process.

The next step in the HRP process, which employs inverse-variance allocation, is supported by this covariance matrix reorganization. HRP guarantees that the portfolio risk is more evenly distributed through this alignment of the covariance matrix, maximizing diversification's advantages.

Recursive bisection—the last phase in HRP—involves splitting the portfolio into smaller sub-portfolios and assigning weights depending on the inverse-variance of every sub-portfolio. This recursive process weights all assets. The secret to this approach is making sure the portfolio evenly distributes risk (variance).

The portfolio is first split into two sub-portfolios, P_1 and P_2, based on the hierarchical structure. Let Σ_1 and Σ_2 be the covariance matrices for the two sub-portfolios. Each sub-portfolio's variance is computed as follows:

$$\mathrm{Var}(P_1) = w_1^T \Sigma_1 w_1, \quad \mathrm{Var}(P_2) = w_2^T \Sigma_2 w_2$$

Where w_1 and w_2 are the weight vectors for sub-portfolios P_1 and P_2 andΣ_1 and Σ_2 are the covariance matrices of the respective sub-portfolios.

The variance of every sub-portfolio is computed using the inverse-variance allocation technique, whereby the allocation for each asset is inversely related to its variance. This guarantees an even distribution of the portfolio's risk, with more weight assigned to assets or sub-portfolios with lower variance.

The weight $w_i^{(j)}$ of asset i within sub-portfolio j is given by the inverse of the variance of the sub-portfolio, as expressed by the formula:

$$w_i^{(j)} = \frac{1}{\mathrm{diag}[\Sigma_j]^{-1}} \cdot \frac{1}{\mathrm{trace}(\mathrm{diag}[\Sigma_j]^{-1})}$$

Where $w_i^{(j)}$ is the weight of asset i in sub-portfolio j, Σ_j is the covariance matrix of the assets in sub-portfolio j, $\mathrm{diag}[\Sigma_j]^{-1}$ is the inverse of the diagonal elements of Σ_j (the variances of the assets), $\mathrm{trace}(\mathrm{diag}[\Sigma_j]^{-1})$ is the sum of the inverse variances, which normalizes the portfolio weights.

This equation guarantees that assets with less variance get more weight in the sub-portfolio. Every sub-portfolio goes through this process again, guaranteeing equal distribution of risk at every hierarchical level.

plotting every sub-portfolio further into smaller sub-portfolios continues the recursive bisection process. The variance of every sub-portfolio is recalculated at every hierarchical level, and the inverse-variance allocation is used. This recursive approach reduces risk concentration by guaranteeing that the portfolio is diversified at every level of the hierarchical tree.

From a mathematical standpoint, the recursive bisection can be shown as:

$$P_j \quad \rightarrow \quad \text{split into two sub-portfolios} \quad P_{j,1}, P_{j,2}$$

The portfolio is split at every stage into two sub-portfolios according to the hierarchical clustering structure; weights are given based on the inverse variance. All assets are weighted and assigned in this manner.

Finally, the total weight w_i of asset i in the entire portfolio is the weighted sum of its contributions from all-recursive bisections:

$$w_i = \sum_{j=1}^{K} w_i^{(j)}$$

Where $w_i^{(j)}$ is the weight of asset i in sub-portfolio j, and K is the total number of sub-portfolios.

3.10 Reinforcement Learning

Because of its capacity to adjust to changing market conditions and maximize asset allocation techniques over time, Reinforcement Learning (RL) has been applied as a AI tool for portfolio allocation. Unlike conventional approaches, which depend on static models or forecasts, RL models interact with the asset returns within the market to learn directly from historical performance. In terms of portfolio management, this interaction means changing the asset weights of the portfolio depending on the observed market conditions and related rewards.

RL's fundamental idea is that an agent, constantly evolving to new data, learns how to decide by taking actions that maximize total rewards over time. Often assessed using metrics like the Sharpe ratio or total return, the agent in portfolio allocation *dynamically changes* the asset weights to maximize the portfolio's risk-adjusted returns which is known as **portfolio rebalancing** [7].

Several elements in RL set the framework for acquiring optimal tactics. These elements are as follows:

1. Agent: The decision-maker is a portfolio manager, a robo-advisor, or an algorithmic trading system—the agent's objective is to understand how to distribute the portfolio properly depending on the market conditions.

2. State: Typically, the agent's view of the market defines the state, which is the present condition or environment. This could be the financial assets' correlation matrix for portfolio allocation, which reflects the relationships among the portfolio's instruments. Reflecting the changing state of the market, this matrix is dynamically updated.

3. Action: The action is the choice made by the agent in reaction to the condition. Actions for portfolio management are the allocation and rebalancing of portfolio weights, which can change with time as the agent interacts with the surroundings. Models like Deep Q-Networks (DQN) provide Q-values that help one choose the actions.

4. Reward: The reward function measures how well an action succeeds. Often in portfolio allocation, this is the Sharpe ratio (or another financial performance measure), which balances risk and returns. The incentive drives the agent to act to maximize risk-adjusted returns. Depending on the desired goal, other measures such as maximum drawdown or cumulative returns could be used instead.

5. Environment: The environment is the market or financial system that the agent engages with. The environment gives the agent's actions feedback—rewards. For instance, in a cryptocurrency portfolio, the environment would comprise elements including liquidity conditions, market volatility, and asset price fluctuations.

The RL process's foundation comprises these elements, which enable the agent to maximize the total cumulative reward by optimizing a policy directing which action to take in any given state.

The RL agent aims to find the best policy π^* maximizing the expected cumulative reward over time. Mathematically, the goal is to identify the policy π that maximizes the return, defined as the sum of discounted rewards. At time t, the total reward G_t is expressed as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Where R_t is the immediate reward at time t and γ is the discount factor (ranging from 0 to 1), which controls the importance of future rewards relative to immediate rewards.

Using the value function $V(s)$, which forecasts the long-term reward of being in a given state s, the agent learns to maximize the expected cumulative return by assessing its actions.

$$V(s) = \mathbb{E}[G_t | S_t = s]$$

The action-value function, or Q-value, $Q(s, a)$, offers a measure of the long-term reward expected from executing a specific action a in state s:

$$Q(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a]$$

The aim is to discover the ideal Q-value function $Q^*(s, a)$, which directs the agent in choosing the best feasible action in every state.

The Bellman equation gives a recursive link to compute the value function or Q-value. The Bellman equation for the value function is as follows for value-based RL techniques:

$$V(s) = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) | S_t = s]$$

For Q-learning (a value-based method), the Bellman equation for the Q-value is:

$$Q(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')|S_t = s, A_t = a]$$

Using iteration over possible states and actions, these equations provide the basis for acquiring the best policy. The objective is to identify the policy maximizing the Q-value for every state-action combination.

The state and action spaces are large in complicated portfolio management settings, and conventional Q-learning could become computationally impractical. Deep Q-Learning (DQN) solves this problem using a deep neural network approximating the Q-value function. The deep Q-network $Q(s, a; \theta)$ maps states to action values by learning the optimal set of weights θ.

Derived from the Bellman equation, the DQN update rule minimizes the loss between the expected Q-value and the target Q-value:

$$L(\theta) = \mathbb{E}[(r + \gamma \max_{a'} Q(S', a'; \theta^-) - Q(S, a; \theta))^2]$$

Where r is the immediate reward, γ is the discount factor, and θ^- are the weights of the target network (periodically updated to stabilize training).

Experience replay updates the target Q-value by having the agent retain past experiences and sample mini-batches to modify the Q-values. This approach increases learning stability and helps to break correlations in the data.

Another category of RL algorithms is represented by policy gradient techniques, in which the agent directly learns a policy function $\pi(a|s;\theta)$ that translates states to actions. The aim is to maximize the expected return, so optimizing the policy:

$$J(\theta) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t R_t]$$

The policy is updated using gradient ascent to increase the expected return:

Where α is the learning rate. Often used with deep learning models to approximate complicated policies, policy gradient techniques are especially beneficial in settings with a continuous action space.

Portfolio allocation defines the environment as the market and the agent, the portfolio manager, interacts with this environment by changing the weights of several assets in the portfolio. The state s_t at time t can include historical asset prices, returns, and other market indicators. The action a_t represents the allocation decisions, such as the percentage of capital to invest in each asset.

Usually, the reward is determined from the portfolio return, which can be modified for risk using measures such as the Sharpe ratio or total return.

$$R_t = \frac{r_t}{\sigma_t}$$

Where r_t is the portfolio return at time t and σ_t is the standard deviation (risk) of the portfolio returns.

Using RL algorithms such as DQN or policy gradients, the agent learns to dynamically change portfolio weights to maximize cumulative rewards over time, considering both returns and risks. Optimization seeks to maximize the risk-adjusted return, enhancing portfolio performance under uncertain and dynamic market conditions.

4 Results

4.1 Equally Weighted, Max Sharpe, Min Volatility

Our benchmarks' portfolios underlying assets were the so-called *Magnificent Seven* (MAG7) stocks: Amazon (AMZN), Google (GOOGL), Apple (AAPL), Microsoft (MSFT), Tesla (TSLA), Nvidia (NVDA), and Meta (META). From January 1, 2021, to December 31, 2023, historical adjusted closing prices were retrieved from Yahoo Finance. Daily returns were calculated from these prices and annualized to get expected returns. The covariance matrix of returns was calculated to capture the correlation and variance across the assets.

Although it showed the highest volatility (44. 38%), the maximum sharpness ratio portfolio produced the best return of 47.72%. Despite this, the Sharpe ratio of 107.54% showed that the portfolio achieved the best risk-adjusted return. With a volatility of 26.11%, the Minimum Volatility Portfolio, while providing a lower return (19.67%), effectively reduced risk, making it more appropriate for conservative investors. Although basic, the Equal Weights Portfolio offered a fair return (23.49%) and a Sharpe ratio (73.34%) lower than the other two approaches,

Portfolio	Return	Volatility	Sharpe Ratio
Equal Weights Portfolio	23.49%	32.03%	73.34%
Maximum Sharpe Ratio Portfolio	47.72%	44.38%	107.54%
Minimum Volatility Portfolio	19.67%	26.11%	75.34%

Table 1: Performance Metrics for Portfolio Weights (2021-2023)

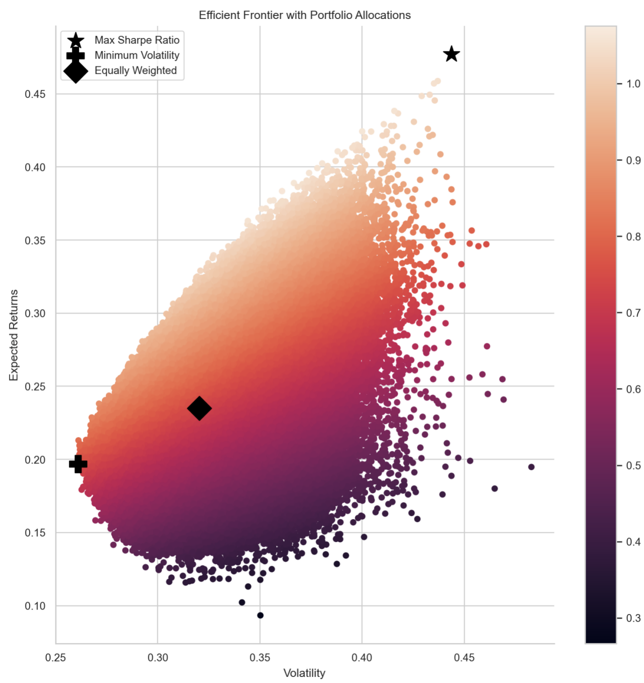

Figure 1: Efficient Frontier with Portfolio Allocations

suggesting suboptimal risk-adjusted performance. With the risk (volatility) on the x-axis and the expected return on the y-axis, the efficient frontier plot shows the trade-off between risk and return for the portfolios. Positioned at the ideal point, the Maximum Sharpe Ratio Portfolio (black star) obtains the maximum Sharpe ratio with the optimal risk-return tradeoff. The Minimum Volatility Portfolio (black diamond) is located toward the lower left side of the graph, reflecting its lower risk and lower returns. Situated in the center of the graph, the Equal Weights Portfolio (black plus sign) suggests a fair risk-return trade-off.

4.1.1 Equally Weighted

The Equal Weighted Portfolio aims for simplicity by assigning each asset an equal share of capital. The constant weights are spread evenly across all seven assets (no optimization is required), so the seven weights are simply $\frac{1}{7}$:

- AMZN: 14.29%
- GOOGL: 14.29%
- AAPL: 14.29%
- MSFT: 14.29%
- TSLA: 14.29%
- NVDA: 14.29%
- META: 14.29%

With no regard for individual asset returns or risk variations, this allocation approach assumes that all assets are equally important. Investors who want a balanced diversified portfolio without favoring one asset over another or requiring complex optimization techniques may find this approach to be most appropriate for their needs. However, this approach ignores any variations in the performance or risk profile of the assets.

4.1.2 Max Sharpe

The Maximum Sharpe Ratio Portfolio seeks to maximize the risk-adjusted return by investing more in assets that produce the highest return for a given level of risk. The correlation between the stocks and expected returns determines the asset allocation. As a result, the strategy considers risk diversification and prioritizes the asset with the highest Sharpe ratio. The weights for this portfolio are:

- **AMZN**: 0.89%
  Amazon's low allocation is likely a result of its volatility relative to other stocks. Although it may have decent returns, its risk (volatility) may have been too high to justify a larger weight in a portfolio aiming to optimize the Sharpe ratio.
- **GOOGL**: 2.09%
  Google's allocation is relatively small. Like Amazon, while its returns might be favorable, its volatility may not justify a larger allocation. In addition, the correlation of GOOGL with other assets in the portfolio can affect its weighting in the optimized strategy.
- **AAPL**: 12.77%
  Apple received a moderate allocation, reflecting its strong return potential. However, the strategy also considers its volatility, which is likely moderate compared to other assets, allowing for a larger allocation.
- **MSFT**: 4.27%
  Microsoft's allocation is relatively modest, likely because its volatility is higher than that of Apple or lower-risk assets. Despite its growth potential, the strategy balances returns against volatility, keeping the allocation in check.
- **TSLA**: 6.37%
  Tesla's allocation is higher than other tech stocks such as Amazon and Microsoft. Tesla has shown high returns in the past, but has also had high volatility. The higher weight suggests that the model is willing to accept risk for higher return.
- **NVDA**: 72.04%
  Nvidia received the largest allocation in this strategy by far. Nvidia's strong historical performance and high expected returns outweigh its risk in this portfolio. The plan allocates a significant portion to Nvidia as it contributes significantly to the portfolio's overall return and achieves a favorable risk-return tradeoff.
- **META**: 1.56%
  Meta's weight is low, due to its lower return potential and volatility compared to other portfolio stocks. The strategy allocates only a small percentage to Meta, probably because of its lesser impact on the Sharpe ratio.

Nvidia (NVDA) is heavily favored in the allocation due to its high potential for return and the portfolio's aim to optimize return while controlling volatility. In addition to taking into account the other assets to preserve diversification and control risk, the portfolio seeks to maximize its expected returns by allocating a larger portion to NVDA.

4.1.3 Minimum Volatility

The goal of the Minimum Volatility Portfolio is to reduce the overall risk of the Portfolio. The approach frequently sacrifices returns to allocate capital to assets with less volatility. The best candidates for this strategy are risk-averse investors willing to forgo a certain amount of return to lower portfolio risk. The weights for this portfolio are:

- **AMZN**: 41.50%
  Amazon's relatively high allocation in the Minimum Volatility Portfolio suggests that it is one of the less volatile assets in the set. Although its returns are not the highest, its volatility is likely more in line with minimize overall portfolio risk.

- **GOOGL**: 5.45%
  Google, like Amazon, has relatively low volatility compared to other assets in the set, which is justified by its allocation to the portfolio. However, it still receives a smaller allocation than Amazon, reflecting its relatively lower contribution to the portfolio's return.
- **AAPL**: 16.72%
  With moderate volatility, Apple has also allocated more capital to this strategy. Balances return and risk, making it a valuable asset in a portfolio with minimum volatility.
- **MSFT**: 3.04%
  Microsoft, like Google, has a moderate level of risk, but due to its relatively higher volatility compared to Apple or Amazon, it receives a smaller allocation.
- **TSLA**: 32.18%
  Despite its high returns, Tesla's high volatility leads to the allocation of a significant portion of capital. In the Minimum Volatility strategy, the weight of TSLA is substantial because, relative to its volatility, it contributes positively to the portfolio's risk-adjusted return.
- **NVDA**: 0.53%
  Nvidia's allocation is minimal in this strategy due to its high volatility. Although it has excellent return potential, the strategy minimizes risk by significantly reducing Nvidia's allocation.
- **META**: 0.58%
  Meta also receives a minimal allocation. Its low expected returns and higher volatility compared to assets such as Apple or Amazon make it less favorable for a minimum volatility approach.

Due to their comparatively lower volatility when compared to high-risk stocks like Nvidia (NVDA) and Tesla (TSLA), Amazon (AMZN), and Apple (AAPL) are strongly preferred in the Minimum Volatility Portfolio's weight distribution. Although the portfolio reduces exposure to high-risk assets, minimizing portfolio risk leads to sizable allocations to AMZN and AAPL.

4.1.4 Forward Testing Results

The portfolios were forward-tested using actual market data for 2024, applying the weights obtained from historical data. The performance of each portfolio during the testing period was analyzed as follows. Figure 2 shows the

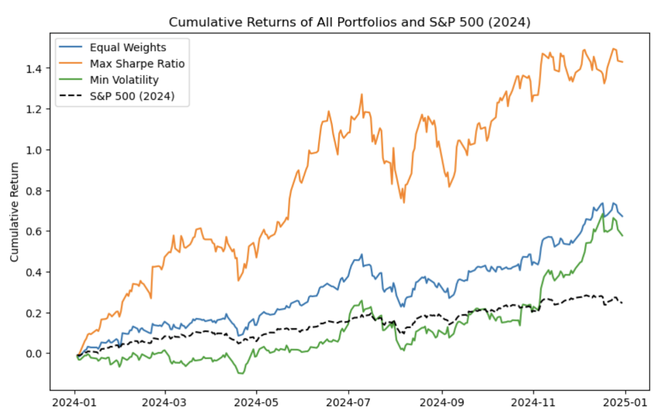

Figure 2: Portfolios' Cumulative Returns in 2024 (Naive, Max Sharpe, Min Volatility

cumulative returns of three different portfolio strategies, Equal Weights (blue), Maximum Sharpe Ratio (orange), and Minimum Volatility (green), alongside the S&P 500 index benchmark (black dashed line). The test offers insights into the robustness of each optimization technique by assessing how portfolios built from historical data perform under live conditions, thereby evaluating their performance under live conditions.

By the end of the year, the Maximum Sharpe Ratio Portfolio outperforms all others, producing a very significant total return. Its better performance shows its optimization toward maximizing risk-adjusted returns, enabling it to exploit bullish trends and favorable market volatility structures efficiently.

Though simple, the Equal Weights Portfolio shows a consistent and moderate upward trend. Though it has no optimization or risk management, its varied structure offers a strong baseline, finishing the year with decent gains.

By contrast, the Minimum Volatility Portfolio offers the most consistent performance with small downward trends, which follows its design to prioritize low variance over high return. Although it falls short of the other strategies in total return, it might attract more risk-averse investors looking for capital preservation and more consistent performance.

Based on an initial investment of $\$10,000$, the final portfolio values were:

- **Equal Weights Portfolio Value**: $16,717.46
- **Portfolio Value with Maximum Sharpe Ratio**: $24,028.55
- **Value of Minimum Volatility Portfolio**: $13,349.75

These results highlight the trade-off between risk minimization and return maximizing across several approaches. Though maybe subjecting investors to more volatility, the Maximum Sharpe Portfolio produced the greatest return. By contrast, the Minimum Volatility Portfolio preferred stability over return; the Equal Weights approach revealed the drawbacks of naive diversification without optimization.

The real-world asset purchases were then simulated using the Discrete Allocation approach; uninvested cash remained because of share indivisibility. The naive portfolio is equally split across the seven assets, with small amounts of uninvested cash ($14.96).

The Maximum Sharpe Ratio Portfolio allocated 52 shares bought, Nvidia (NVDA) gets a notable amount of capital, suggesting a great expected return from this asset. The remaining cash was $26.12. The Minimum Volatility Portfolio allocated 17 shares to AMZN and eight shares to TSLA, minimizing volatility but creating a large cash reserve of $26.21. These allocations, which reflect the particular objectives of each strategy—maximizing risk-adjusted returns, minimizing volatility, and guaranteeing diversification, were based on the computed portfolio weights.

These separate allocations reflect the fundamental goals of every strategy, whether maximizing the Sharpe ratio, minimizing volatility, or guaranteeing equal diversification.

This forward-testing stage confirms that past-derived portfolio weights may be helpful in real-time market environments. On the other hand, it also emphasizes the need for more improvements, especially with more empirical results, which we will see later.

4.2 Multi-Factor Models

4.2.1 Capital Asset Pricing Model

Portfolio	**Beta**	**R-squared**	**P-value**	**Alpha**
Maximum Sharpe Ratio	1.7494	0.391	0.000	0.0012
Minimum Volatility	1.4749	0.482	0.000	-0.0024
Equally Weighted	1.4238	0.614	0.000	0.0072

Table 2: CAPM Results for the Three Portfolios

To evaluate market sensitivity and performance attribution of the three portfolio strategies: maximum sharpe ratio, minimum volatility, and equally weighted, we performed CAPM regressions using historical return data from January 1, 2021, to December 31, 2023. Table 2 summarizes the beta, R-squared , p-value, and alpha for each portfolio.

The Maximum Sharpe Ratio Portfolio demonstrates the highest beta (1.7494), indicating it is the most volatile and sensitive to market movements. Its R-squared value of 0.391 shows that approximately 39.1% of its return variability is explained by market returns, implying that other non-market factors may influence performance. The portfolio's alpha of 0.0012 is positive but statistically insignificant, suggesting no evidence of abnormal returns beyond what the market explains.

Despite its risk-averse objective, the Minimum Volatility Portfolio has a beta of 1.4749, indicating greater volatility than the market. With an R-squared of 0.482, it has a stronger relationship with market returns than the Sharpe portfolio. It alpha of -0.0024 is negative and statistically insignificant, reinforcing that the portfolio's performance is mainly attributable to market movements rather than skill-based returns.

Among all three portfolios, the Equally Weighted strategy shows the highest R-squared at 0.614, suggesting that the market explains 61.4% of its returns. Its beta of 1.4238 reflects moderate market sensitivity, closely mirroring market behavior. It also produces the highest alpha at 0.0072, but this value is statistically insignificant again, indicating no meaningful outperformance after accounting for market exposure.

All three portfolios exhibit statistically significant relationships with the market (p-values of 0.000), confirming that market movements are a primary driver of returns. However, none of the portfolios produced statistically significant alpha, meaning they did not generate returns that exceeded what would be expected given their market exposure. From a risk-return perspective, the Maximum Sharpe Ratio Portfolio is the most aggressive, offering the potential for higher returns but with increased volatility. The Minimum Volatility Portfolio maintains a conservative risk profile but offers lower market-relative returns. While unoptimized and straightforward, the Equally Weighted Portfolio closely tracks the market and achieves a relatively high explanatory power.

These results imply that although conventional optimization techniques fit portfolios with preferred risk levels, they do not naturally generate alpha. Investors wanting to improve outperformance might have to include extra elements outside the single-factor CAPM framework, including macroeconomic indicators, sector momentum, or nonlinear machine learning models.

4.2.2 Fama-French 3-Factor Model

Portfolio	**Intercept**	**MKT**	**SMB**	**HML**	**R-squared**
Maximum Sharpe Ratio	0.0204	1.3216	-0.5790	-0.8752	0.527
Minimum Volatility	0.0111	1.1196	-0.5368	-0.6194	0.741
Equally Weighted	0.0177	1.3427	-0.1244	-0.8263	0.820

Table 3: Fama-French 3-Factor Model Regression Results

We apply the Fama-French 3-Factor Model to the Maximum Sharpe Ratio, Minimum Volatility, and Equally Weighted portfolios to understand better the analysis of return drivers beyond market exposure alone. This model extends CAPM by incorporating two additional explanatory variables: SMB, the size factor (small minus big), capturing the return difference between small- and large-cap stocks, and HML, the value factor (high book-to-market minus low), capturing the return difference between value and growth stocks.

Table 3 summarizes the regression coefficients for each factor, the intercept (alpha), and R2R2 values, indicating the model's explanatory power for each portfolio's excess returns.

For the Maximum Sharpe Ratio Portfolio, the R-squared is 0.527, indicating that approximately 52.7% of the portfolio's excess returns variation can be explained by the three Fama-French factors (market, SMB, and HML). The F-statistic: 11.49, with a p-value of 3.14e-05, indicating that the model as a whole is statistically significant. With a p-value of 0.175, the intercept of 0.0204 is not statistically significant. This implies that rather than any unusual return, the market and the two factors—SMB and HML—drive the portfolio's returns mostly. A p-value of 0.000 and MKT coefficient of 1.3216 show a statistically significant positive correlation between the portfolio's excess returns and the market. A positive beta of 1.3216 indicates that the portfolio is more responsive to changes in the market. The coefficient for SMB is -0.5790, but the p-value of 0.275 suggests the size factor is not statistically significant in accounting for the portfolio's excess returns. HML's coefficient is -0.8752; with a p-value of 0.005, it is statistically relevant, suggesting the value element damages the portfolio's returns.

For the Minimum Volatility Portfolio, the R-squared is 0.741, indicating that 74.1% of the portfolio's excess returns variation is explained by the three factors in the model, which is a relatively high value. With a p-value of 3.24e-09, the F-statistic of 29.52 shows that the model is quite statistically relevant. With a p-value of 0.141, the intercept (Alpha) is 0.0111, implying that the alpha is not statistically relevant, so the three factors mainly account for the returns. A MKT coefficient of 1.1196 with a p-value of 0.000 indicates a statistically significant positive correlation between the portfolio's returns and the market, suggesting that the portfolio is vulnerable to market fluctuations. Statistically significant at the 5% level, the SMB coefficient is -0.5368 with a p-value of 0.049. This implies that the performance of smaller stocks (size factor) damages the portfolio's returns. With a p-value of 0.000, the HML coefficient is -0.6194, suggesting that the value factor notably reduces the portfolio's performance.

For the Equally Weighted Portfolio, the R-squared is 0.820, indicating that 82.0% of the variation in the portfolio's excess returns can be explained by the three factors, which is very high. The F-statistic: 47.08, with a p-value of 1.18e-11, suggesting that the model is quite statistically relevant. Statistically significant at the 5% level, the intercept (Alpha) is 0.0177 with a p-value of 0.022. This implies that the Equally Weighted Portfolio has a slight but notable positive abnormal return, suggesting that the portfolio slightly outperformed the market during the period. With a p-value of 0.000, the MKT coefficient of 1.3427 indicates a statistically significant positive correlation between the portfolio's returns and the market, implying that the portfolio moves closely with the market. With a p-value of 0.637, the SMB coefficient is -0.1244, which is not statistically relevant. This suggests that the size element has little effect on the portfolio's returns. With a p-value of 0.000, the HML coefficient of -0.8263 suggests a statistically significant negative influence of the value factor on the portfolio returns.

Across all three portfolios, the market factor is a statistically significant and dominant driver of returns. However, each portfolio demonstrates a consistent and statistically significant negative loading on the value factor (HML), indicating that value stocks underperformed across the observed period. Only the Minimum Volatility Portfolio exhibited a significant relationship with the size factor (SMB), showing a negative influence from small-cap exposure.

Although the Equally Weighted Portfolio displayed a statistically significant alpha, it was modest. The Maximum Sharpe Ratio nor the Minimum Volatility Portfolio exhibited meaningful abnormal returns after adjusting for the three factors. These findings suggest that while factor models can effectively explain portfolio return dynamics, none of the strategies consistently delivered excess returns beyond exposure to known risk factors. This opens the door for more advanced models or alternative alpha sources to enhance portfolio performance in future research or applications.

4.3 PCA Portfolio Results

Focusing on lowering the dimensionality of the data while capturing the most critical components of variability in the stock returns, the PCA Portfolio offers a different portfolio optimization method. The method seeks to find a smaller set of principal components that account for most of the variation in asset returns, simplifying the portfolio allocation process. The PCA portfolio aims to focus on the most influential factors, offering a simpler but still effective portfolio optimization strategy compared to the traditional methods.

At 2.79%, the portfolio volatility was relatively low compared to other portfolios, including the Maximum Sharpe Ratio Portfolio (44.38% volatility) and the Minimum Volatility Portfolio (26.11% volatility). At 0.04, the Sharpe ratio for the PCA portfolio is lower than the Maximum Sharpe Ratio Portfolio's (1.08). However, it still suggests the portfolio produced returns about its risk, albeit with less efficiency than other approaches.

The portfolio weights were optimized by maximizing the explained variance with a reduced number of factors using PCA. The PCA model decomposed the covariance matrix of asset returns, extracting the leading eigenvectors (principal components) responsible for the most significant portion of the variability in the data.

The best PCA portfolio weights for the MAG7 stocks were as follows:

- **AAPL**: 6.69% Reflecting its status as a stable, high-cap stock with less volatility than TSLA and NVDA, Apple (AAPL) was given a modest allocation of 6.69%. Though AAPL is a significant market player, especially in the context of the main components, its returns are more stable and less volatile than those of other MAG7 stocks. PCA gives assets with more variance more weight, thus AAPL's more consistent returns and smaller swings led to a lower weight in the portfolio relative to more volatile stocks like TSLA and NVDA, which have a larger impact on market-wide movements.

- **AMZN**: 5.99% Reflecting its importance in the consumer goods and e-commerce industries, Amazon (AMZN) got a fair allocation of 5.99%. Though AMZN shows significant volatility, its return behavior is less extreme than that of TSLA and NVDA. Therefore, its contribution to the first principal component is less than that of more volatile shares. Though a significant and very liquid stock with strong growth potential, AMZN's less dramatic return swings caused a more moderate weight in the PCA-optimized portfolio, reducing its influence in the first principal component.

- **GOOGL**: 2.44% Google (GOO) got a 2.44% allocation, reflecting its consistent standing in the tech sector, much like META. But, like META, GOOGL's lesser volatility and risk profile led to a smaller portfolio weight. Though it greatly affected market movements, its lower variance in returns relative to more volatile stocks like TSLA and NVDA made clarifying the dominant principal components obtained via PCA less important.

- **META**: 2.68% Meta (META) got a 2.68% allocation, indicating its lower volatility than some higher-weighted equities. Although META is a powerful stock, it has struggled recently, which has caused its returns during the examined period to be less variable. META's relatively lower volatility and risk caused a smaller portfolio weight since its less dramatic price changes make explaining the dominant principal components produced by PCA less critical.

- **MSFT**: 3.67% Microsoft (MSFT), like AAPL, is a major tech firm with less volatility than stocks like TSLA and NVDA. Reflecting its somewhat lower variability in returns, it got a lesser allocation of 3.67%. Although MSFT is still a powerful stock, its more consistent growth and lower return variance caused a more equal and less forceful inclusion in the portfolio. PCA's weighting system prioritizes assets that significantly influence market volatility; thus, MSFT's moderate behavior causes a smaller allocation in the optimized portfolio.

- **NVDA**: 26.06% Nvidia (NVDA) got a notable 26.06% allocation, indicating its central importance in propelling expansion in the technology sector, especially in AI and semiconductors. Like TSLA, NVDA showed significant volatility and strong co-movement with more general market trends, particularly in technology and growth equities, which made it a key element in the first principal component. PCA gives more weight to assets with high variance and strong correlation with other market-driving stocks; NVDA's considerable weight shows its significant impact in sectors like AI, where it has generated strong returns during times of expansion.

- **TSLA**: 52.45% With more than 50% of the whole capital assigned to this stock, Tesla (TSLA) got the most allocation in the PCA portfolio. TSLA's significant volatility and co-movement with other high-performing equities in the market explain most of this. The MAG7 and TSLA showed the most considerable return variation since it is one of the most erratic stocks; thus, it significantly influenced the first principal component. Stocks with greater variance tend to get more weight from PCA since they are more potent in clarifying the general market dynamics. Reflecting its position as a market leader in the electric vehicle sector and a high-growth stock, with significant potential to drive returns during strong market performance, TSLA's considerable weight in the portfolio is evident.

Strongly favoring TSLA and NVDA, the PCA model assigned 52.45% of the portfolio to TSLA and 26.06% to NVDA. This result shows the significant variance and co-movement of these two stocks with the first principal component, which accounts for most of the market variability. Both stocks showed a strong correlation with the general market, which helped them to dominate the PCA optimization approach.

4.3.1 Forward Testing Results

The PCA-optimized portfolio was forward-tested using 2024 market data, applying portfolio weights derived from principal component analysis of historical asset returns. The purpose was to evaluate how well PCA, as a dimensionality reduction method, could identify and translate dominant return drivers into adequate real-world portfolio allocations.

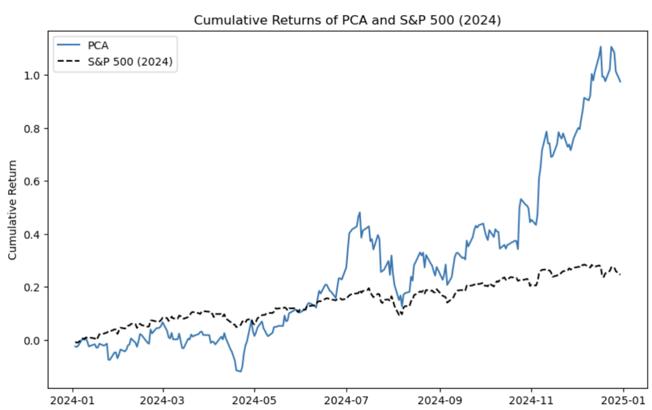

Figure 3: Portfolios' Cumulative Returns in 2024 (PCA)

The PCA portfolio outperformed the S&P 500 during the testing period, as indicated in Figure 3. Although early performance was slow, the portfolio rose significantly in the year's second half, finally returning 97.5This growth aligns with large allocations to TSLA and NVDA, two top-performing stocks in 2024. PCA's capacity to highlight the main directions of variance in the data lets it focus on these high-momentum assets. This performance was primarily accomplished with a reasonably consistent risk profile, therefore stressing PCA's capacity for low-volatility yet high-return portfolio building. The final portfolio value reached $19,750.25, nearly doubling the initial

$10,000 investment over the year. This value implies that PCA was able to significantly compound returns and find positive market dynamics in the second half of the year.

Realistic investment conditions were simulated using a discrete allocation method. Reflecting practical limits such as indivisible shares and transaction concerns, this changes continuous portfolio weights into integer share numbers. The resulting allocation is as follows:

- **AAPL**: 3 shares
- **AMZN**: 3 shares
- **GOOGL**: 1 share
- **MSFT**: 1 share
- **NVDA**: 18 shares
- **TSLA**: 13 shares

The portfolio ended with $64.41 in leftover cash, indicating that most of the capital was effectively deployed. The concentration in NVDA and TSLA is consistent with PCA's identification of assets contributing most to variance in the returns data.

The PCA portfolio shows the practical value of dimensionality reduction in portfolio optimization. PCA builds a strategically weighted portfolio by focusing on principal components that explain the most important variance in asset returns. Though the Maximum Sharpe Ratio Portfolio had greater total returns, the PCA portfolio offered a good alternative by producing more consistent, lower-volatility performance while seizing notable profits.

Furthermore, the effective use of discrete allocation verifies PCA's viability in actual portfolio control. Its capacity to produce notable returns while maintaining a modest risk attitude makes it appealing for investors looking for a compromise between development and stability—especially in unstable or uncertain market conditions.

PCA, therefore, provides a strong optimization technique that converts complicated asset relationships into practical portfolio weights. Especially appropriate for long-term, risk-aware investors, it is a hopeful compromise between purely risk-minimizing and aggressive return-seeking tactics.

4.4 Hierarchical Risk Parity

A key financial management component, portfolio optimization seeks to distribute assets to maximize returns and minimize risk. Conventional techniques like mean-variance optimization (MVO) are often used to reach this equilibrium. These methods, however, have many difficulties, especially when handling large, complicated, and varied asset collections such as those found in major indices, including the Dow Jones Industrial Average (DJ30). Among these problems are excessive volatility, inadequate diversification, and over-reliance on past covariances, all of which can result in less-than-ideal asset allocations.

Unlike MVO, which may suffer from overfitting or excessive reliance on historical data, HRP optimizes the portfolio by focusing on diversification and risk management. The HRP portfolio provides a more complex and empirical approach to meet the difficulties of reaching an equilibrium between volatility and the Sharpe Ratio. HRP effectively reduces problems of concentration and volatility by using hierarchical clustering to group assets according to their risk profiles. HRP ensures better portfolio diversification by applying risk parity to distribute risk across assets through a hierarchical framework evenly. This method enhances the portfolio's risk-adjusted returns and boosts its resilience in changing market conditions.

Applied to a group of stocks from the Dow Jones Industrial Average (DJ30), the HRP portfolio optimization process is explored in this part. The study then thoroughly evaluates the findings, starting with a description of the HRP methodology. Using 2024 market data, we will also look at the forward-testing performance of the HRP model, therefore assessing the effectiveness of the weights obtained from historical data (2021–2023) about actual market results.

The optimal portfolio weights derived from the HRP model represent the allocation across the Dow Jones Industrial Average (DJ30) stocks that minimizes risk while achieving a reasonable expected return. Below are the calculated optimal weights for the DJ30 stocks:

- **AAPL**: 0.01799
- **AMGN**: 0.04726
- **AXP**: 0.01334
- **BA**: 0.01851

- **CAT**: 0.01604
- **CRM**: 0.01787
- **CSCO**: 0.03526
- **CVX**: 0.02496
- **DIS**: 0.02065
- **DOW**: 0.01631
- **GS**: 0.01645
- **HD**: 0.02889
- **HON**: 0.02910
- **IBM**: 0.04842
- **INTC**: 0.01464
- **JNJ**: 0.06650
- **JPM**: 0.01934
- **KO**: 0.05521
- **MCD**: 0.06407
- **MMM**: 0.02088
- **MRK**: 0.04697
- **MSFT**: 0.01794
- **NKE**: 0.01620
- **PG**: 0.04763
- **TRV**: 0.04963
- **UNH**: 0.04597
- **V**: 0.02334
- **VZ**: 0.07580
- **WBA**: 0.02603
- **WMT**: 0.05879

While assigning smaller weights to those with higher volatility and stronger correlations, the HRP portfolio allocates larger weights to stocks with lower volatility and weaker correlations. This strategy improves diversification, lowers portfolio volatility, and helps to distribute risk over assets.

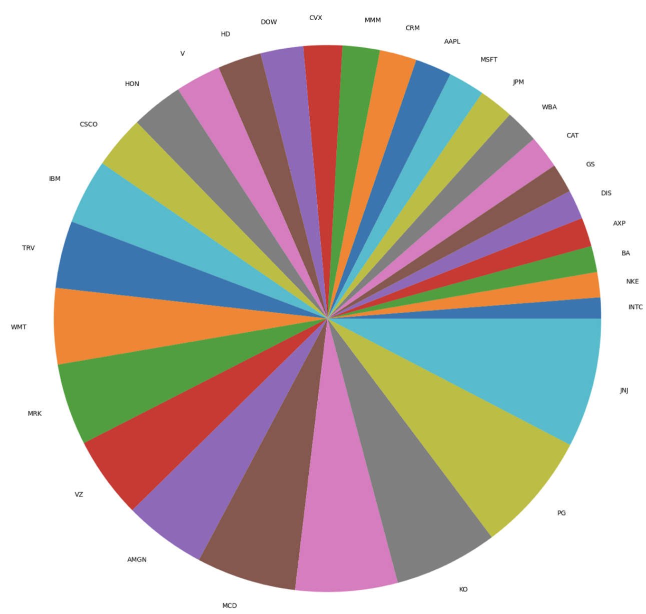

Figure 4: Pie Chart of Weight Distributions for HRP

The pie chart shows the risk-managed allocation of the HRP portfolio; the color coding helps to highlight which stocks get bigger or smaller portfolio portions. While limiting exposure to those with higher risk, the allocation follows the HRP strategy's emphasis on diversification by assigning larger weights to more stable, less volatile stocks.

4.4.1 High-Weighed Stocks (Weight > 0.05)

Stocks with historically low volatility, stability, and weak correlations with other assets are assigned higher weights in the HRP portfolio. These include **Johnson Johnson** (JNJ), **Verizon Communications** (VZ), **McDonald's**

(MCD), **Walmart** (WMT), and **Coca-Cola** (KO). These companies are known for providing consistent returns with relatively lower risk profiles. Their stability and low correlation with other market assets make them ideal candidates for larger portfolio weights, helping to balance risk and ensuring a more stable performance.

4.4.2 Mid-Weighed Stocks ($0.03 < \text{Weight} \leq 0.05$)

Stocks like **TRV** (Travelers Companies), **PG** (Procter & Gamble), **IBM** (International Business Machines), **MRK** (Merck & Co.), **AMGN** (Amgen), **UNH** (UnitedHealth Group), and **CSCO** (Cisco Systems) receive moderate weights in the HRP portfolio. These stocks generally exhibit more volatility and stronger correlations with the broader market, but they also provide important diversification across different sectors. By assigning them moderate weights, HRP achieves a balance between risk and return, offering sectoral variety while mitigating the potential for high correlation risks.

4.4.3 Low-Weighed Stocks ($\text{Weight} \leq 0.03$)

Stocks such as **CVX** (Chevron), **HD** (Home Depot), **HON** (Honeywell International), **WBA** (Walgreens Boots Alliance), **AAPL** (Apple), **AXP** (American Express), **BA** (Boeing), **CAT** (Caterpillar), **CRM** (Salesforce), **DIS** (Walt Disney), **DOW** (Dow Chemical), **GS** (Goldman Sachs), **INTC** (Intel), **JPM** (JPMorgan Chase), **MMM** (3M), **NKE** (Nike), **MSFT** (Microsoft), and **V** receive lower weights in the HRP portfolio due to their higher volatility and stronger correlations with specific market or sector trends. While these stocks contribute to portfolio diversity, their higher risk levels require reduced allocations to minimize their impact on overall portfolio volatility. HRP ensures a more balanced risk-return tradeoff by limiting their influence in the portfolio structure.

4.4.4 Dendrogram

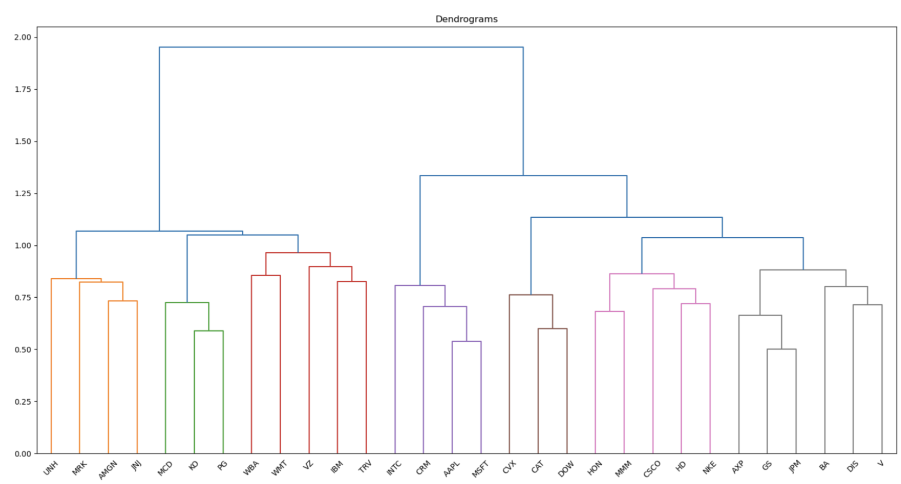

Figure 5: Dendrogram of DJ30 Stocks

The dendrogram visualizes how stocks are connected by grouping assets with strong correlations. Longer branches show weaker correlations, while shorter branches show stronger correlations. The HRP approach ensures no single group becomes overexposed by allocating capital depending on these clusters. Because assets inside the same cluster tend to move together, this approach lowers general portfolio volatility and reduces the over concentration risk. The color-coded grouping simplifies the interpretation of the diversification strategy applied in portfolio construction and highlights the strength of correlations within sectors.

Based on sector-specific and market-sensitive correlations among stocks, the dendrogram reveals several distinct clusters:

1. Cluster 1 (Healthcare and Pharmaceuticals): The blue cluster on the left comprises stocks including UNH, MRK, AMGN, and JNJ. Often moving together because of shared macroeconomic and sector-specific elements, such as market and regulatory pressures, these healthcare and pharmaceutical sector firms are closely related.

2. Cluster 2 (Consumer Staples): The green group comprises MCD, KO, and PG consumer staples firms. Due to their resilience during economic downturns, these stocks often exhibit low volatility and strong correlation with one another. Their consistent performance helps to build a stable portfolio, especially in uncertain economic conditions.

3. Cluster 3 (Diverse Sectors): The red cluster contains the shares WMT, VZ, IDM, INTC, and CRM, which are industries in retail, technology, and insurance. Though varied, the strong correlation within the group suggests a common market sensitivity reflecting more general economic elements affecting these industries equally.

4. Cluster 4 (Technology and Industrial Dectors): The purple cluster includes stocks such as AAPL, MSFT, CRM, CVX, CAT, and DOW. Although covering technology and industrial sectors, these equities reveal a strong correlation motivated by larger market trends, including technological innovations and worldwide supply chains, which makes them susceptible to comparable macroeconomic changes.

5. Cluster 5 (Financial Sector): The orange cluster contains V and GS which reflects financial stocks related because of their shared exposure to macroeconomic influences, including interest rates, regulatory changes, and market liquidity. Though more isolated, the same elements impacting the financial sector affect this financial sector.

This shows how HRP builds a portfolio that optimizes diversification and controls risk using sectoral and market-driven correlations.

4.4.5 Correlation Heatmap of DJ30 Stocks

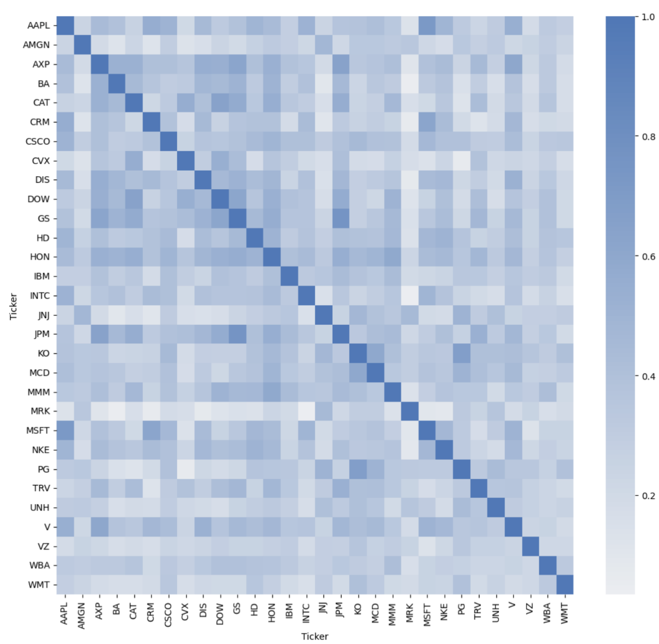

Figure 6: Pairwise Correlation Heatmap of DJ30 Stocks

Figure 6 presents the correlation matrix of the Dow Jones 30 stocks, where each cell represents the Pearson correlation coefficient between the daily returns of two assets. The values range from 0 to 1, with darker shades indicating stronger correlations. A value of 1 along the diagonal represents the perfect self-correlation of each stock with itself.

This matrix is the foundational input for HRP portfolio construction, where assets with higher correlations are clustered together. HRP leverages these relationships to build diversified portfolios that reduce exposure to redundant or co-moving assets.

Several strong pairwise correlations are visible in the heatmap. For example, Apple (AAPL) and Microsoft (MSFT) show a high correlation, as expected, due to their shared position in the technology sector and similar

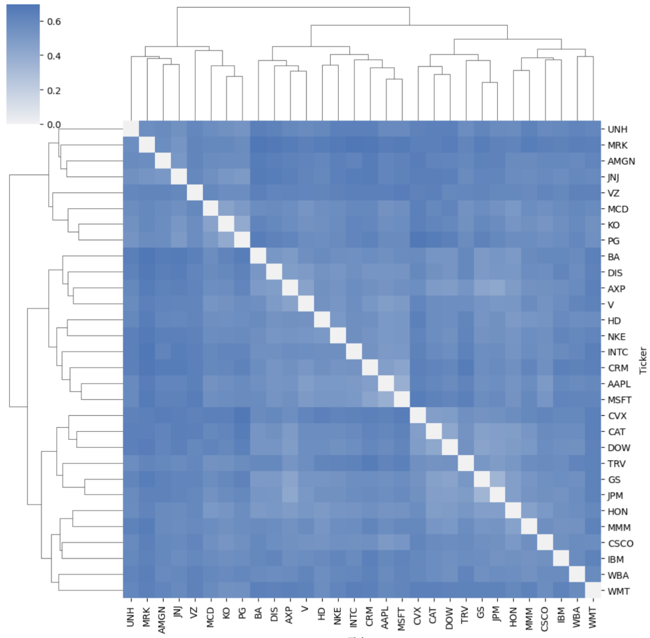

Figure 7: Hierarchical Clustering and Heatmap Dendrogram of DJ30

sensitivity to macroeconomic variables. Another example includes Visa (V) and American Express (AXP), both part of the financial services industry, demonstrating tightly associated movement patterns.

In contrast, weakly correlated asset pairs are also evident. For instance, Walmart (WMT), Apple (AAPL), Procter Gamble (PG), and Microsoft (MSFT) exhibit lower correlations due to differing sector exposures—retail and consumer goods versus technology. These lower-correlation relationships contribute significantly to diversification benefits when constructing optimized portfolios.

4.4.6 Hierarchical Clustering and Heatmap Dendrogram

Figure 7 extends the previous analysis by incorporating a dendrogram, hierarchically clusters stocks based on pairwise correlations. The dendrogram branches reflect the similarity of stock return patterns—shorter branches indicate a higher correlation.

At the top of the dendrogram, we observe clusters of highly correlated technology stocks, such as Apple (AAPL) and Microsoft (MSFT), grouped closely due to their shared exposure to the tech sector. Similarly, financial institutions like Goldman Sachs (GS) and JPMorgan Chase (JPM) form their cluster, influenced by comparable drivers such as interest rate policies, credit risk, and economic outlook. Moderate correlation clusters include companies like Visa (V), American Express (AXP), and Boeing (BA), which span the financial and industrial sectors. These firms respond to broad macroeconomic trends but with sector-specific idiosyncrasies.

At the lower end of the heatmap, we find low-correlation clusters, particularly within the healthcare sector. Stocks like UnitedHealth Group (UNH), Merck (MRK), and Amgen (AMGN) form a distinct group that behaves independently of broader market cycles, driven instead by healthcare-specific news, drug pipeline updates, and policy shifts. Their relatively low correlation to the rest of the index underscores their value for risk diversification.

By focusing on a mix of assets with lower correlations and reducing the weight of highly correlated stocks, we can create more diversified portfolios, optimizing both risk and return.

4.4.7 Bar Chart of Weights

After using HRP, the bar chart above shows the weights given to every stock in the portfolio. The top stocks are mainly from the healthcare and consumer staples sectors, and these weights are the proportion of total capital assigned to each stock.

JNJ is the highest allocation stock at about 7.8%, probably because of its stability as a large healthcare company, which usually has lower volatility than sectors including technology and industrials. Its steady returns also make it a safer wager for risk diversification. As major consumer staples corporations, KO and MCD also get significant

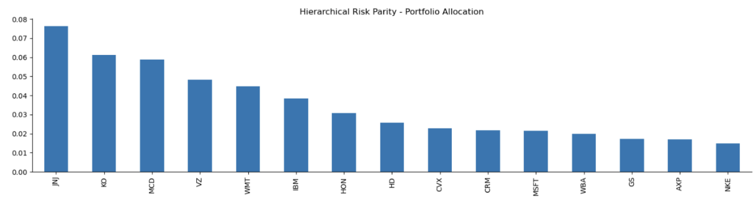

Figure 8: Bar Chart of Weights for DJ30

weight, stressing their stability and lower correlation with more volatile equities. Their greater weight in a portfolio designed for risk parity might be explained by their stability, size, and membership in defensive industries.

After these, moderate allocations go to stocks like HD (Home Depot), HON (Honeywell), and IBM. Spanning technology, industrials, and consumer products helps these businesses diversify by lowering concentration risk. At the bottom of the distribution, we see GS (Goldman Sachs), AXP (American Express), and NKE (Nike). These long-standing businesses have relatively lesser allocations, implying that HRP sees them as more volatile or risky than other portfolio equities.

By guaranteeing diversification, the HRP approach seeks to reduce concentration risk. Biased toward defensive stocks—healthcare, consumer staples, and telecommunications—the portfolio tends to be less volatile. Relatively high weights among the top 10 stocks—including JNJ, KO, MCD, and VZ—help to stabilize the portfolio. By contrast, financial stocks like GS and AXP have lesser allocations, suggesting that these sectors correlate more with the larger market, possibly raising portfolio volatility under market stress.

By giving bigger weights to firms with lower volatility and correlation with the broader market, HRP gives risk minimization and diversification top priority. This approach builds a portfolio with a more consistent risk profile, lowering sensitivity to market changes. HRP helps balance risk and return with varied allocations, reducing total volatility while seizing return potential from low correlation sectors.

4.4.8 Forward Testing Results

Figure **??** displays the cumulative returns of the HRP portfolio against the S&P 500 index during the 2024 testing period. Despite encountering short-term fluctuations, the HRP portfolio maintained a generally upward trajectory throughout the year.

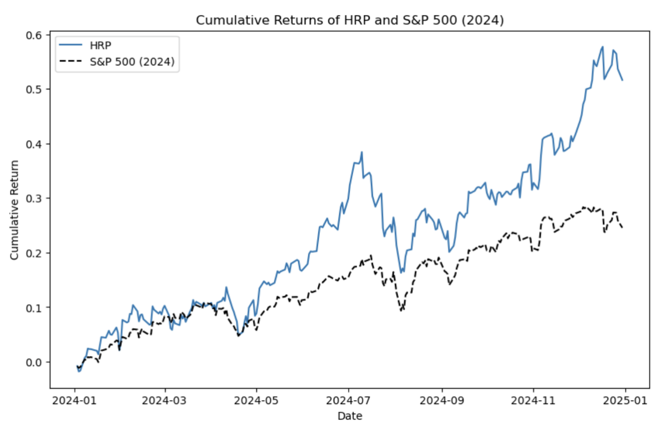

Figure 9: 2024 Cumulative Returns for HRP Portfolio

The steady upward movement of the HRP portfolio shows how well HRP's risk allocation ideas work. The HRP approach could withstand market volatility and capture growth by using the correlation structure among assets and avoiding concentrated exposures. Substantial recoveries followed short-term downward trends, implying that the HRP portfolio took advantage of rotating sector performance while lowering exposure to co-movements among volatile assets.

Starting with a \$10,000 initial investment, the HRP portfolio produced a total return of about 10% by the end of the testing period, reaching a final value of \$11,010.16.This result highlights HRP's capacity to produce consistent, risk-adjusted returns using diversified exposure and asset hierarchical clustering. Although the HRP portfolio was lower than the Maximum Sharpe Ratio Portfolio, which produced a significantly higher return over the same time frame, it showed a more stable and conservative growth path. Though more aggressive and return-oriented, the Maximum Sharpe portfolio also showed more volatility and sensitivity to market changes.

By contrast, the HRP approach prioritized risk parity and diversification, spreading capital portfolio correlation clusters and lessening the influence of sector-specific declines. Consequently, HRP provided a less risky option that saw good, compounding growth all year.

To simulate real-world investability, the HRP portfolio was converted into discrete allocations using whole-share rounding. The allocation is as follows:

- **AAPL**: 1
- **AMGN**: 2
- **BA**: 1
- **CRM**: 1
- **CSCO**: 6
- **CVX**: 2
- **DIS**: 2
- **DOW**: 5
- **HD**: 1
- **HON**: 1
- **IBM**: 2
- **INTC**: 8
- **JNJ**: 5
- **JPM**: 1
- **KO**: 9
- **MCD**: 2
- **MMM**: 2
- **MRK**: 5
- **NKE**: 2
- **PG**: 3
- **TRV**: 2
- **UNH**: 1
- **V**: 1
- **VZ**: 20
- **WBA**: 31
- **WMT**: 7

The allocations show the inverse-variance concept that the HRP approach uses, which aims to balance risk and return over assets. The stocks with the highest weights, such as VZ, WBA, and JNJ , reflect the less volatile assets in the portfolio and therefore deserve a bigger share of the allocation to reduce portfolio risk. On the other hand, stocks with lower weights, such as AXP, DIS, and HD, signify more volatile or higher-risk equities.

The weighting approach stresses long-term stability and risk minimization by under weighting more volatile stocks and allocating more capital to assets with a consistent track record of performance.

After the discrete allocation process, the portfolio had a remaining cash balance of $12.22, a tiny fraction of the original capital not assigned to any assets. The discrete allocation process, in which the available capital is split according to the number of shares required to reach the ideal portfolio, naturally produces this cash reserve.

4.5 Reinforcement Learning

This section discusses the portfolio optimization strategy using Reinforcement Learning (RL), particularly applying the Deep Q-Learning approach to the MAG 7 stocks. The objective of the RL model is to learn optimal portfolio allocations by interacting with the stock market environment and maximizing cumulative returns over time. Below, we explain the structure and the key components of the implemented RL model, followed by a detailed analysis of the performance results.

4.5.1 Code

The portfolio optimization procedure for the MAG 7 equities is implemented using a deep RL model, constructed within the Keras neural network framework. The primary objective of this model is to learn optimal portfolio weights that maximize risk-adjusted returns across the assets.

The analysis begins by extracting and preprocessing the adjusted closing prices of the MAG 7 stocks. These prices are converted into daily returns, which are then used both for the state representation fed into the RL agent and as the basis for the reward calculation that guides learning. Accurate and consistent return data is essential for effectively training the model and for constructing a reliable reward function that reflects real-world portfolio performance.

The architecture of the RL model is based on the actor-critic architecture, a widely adopted approach in modern deep RL. In this framework, two neural networks are trained simultaneously. The first, known as the actor network, is responsible for selecting actions by proposing specific portfolio weights in response to the observed market state. The second, the critic network, evaluates the chosen actions by estimating their associated state-action values, effectively providing feedback on the quality of the portfolio allocations proposed by the actor.

At each decision point, the actor network generates a vector of portfolio weights based on the current state of the environment, which is characterized by recent asset returns or other relevant features. The critic network then assesses these weights by predicting their long-term value, allowing the system to compute a reward signal that drives policy improvement. Through iterative interaction with the environment and continuous feedback from the critic, the actor network refines its strategy, converging toward portfolio allocations that maximize cumulative, risk-adjusted returns over time.

This actor-critic approach enables the RL agent to develop an adaptive and robust investment policy, leveraging the power of deep learning to identify complex, nonlinear patterns in asset price dynamics and to optimize portfolio weights accordingly.

The model takes as input the current portfolio state, which consists of historical asset returns over a chosen lookback window (which we will discuss later). The input shape is determined by the number of assets in the portfolio and the number of time steps used to represent the state. For this study, the input comprises the historical returns of the MAG 7 equities for the period from 2021 to 2023.

To capture complex relationships among assets and their returns, the network utilizes several fully connected (dense) layers, each regularized with L2 penalty and dropout to mitigate overfitting. The dense layers employ the Exponential Linear Unit (ELU) activation function, which, compared to the conventional ReLU, better handles the vanishing gradient problem and accelerates convergence. The input matrix is initially passed through a flattening layer, converting the two-dimensional state representation into a one-dimensional feature vector suitable for subsequent dense layers.

The model architecture includes both actor and critic components. The actor network is responsible for generating the portfolio weights for each asset. To ensure that the output weights lie within the allowable range for long and short positions, the actor uses the hyperbolic tangent (tanh) activation function, which constrains the portfolio weights to the interval $[-1, 1]$. This allows the agent to dynamically allocate long (positive) or short (negative) exposures as dictated by the learning process.

The critic network produces a single scalar value, representing the estimated value of the current state-action pair. This value serves as a measure of the expected future reward associated with the selected portfolio weights, and it guides the learning process by providing feedback on the effectiveness of the actor's allocations. The critic's output layer uses a linear activation to allow for real-valued state-value estimates.

The learning process is driven by the mean squared error (MSE) loss function, applied independently to both the actor and critic networks. This loss quantifies the difference between predicted and target values, ensuring that the actor network gradually improves its portfolio allocation policy, while the critic network refines its value estimates. The Adam optimizer is employed for efficient and stable convergence during training.

The following code illustrates the core structure of the model, including input processing, dense layers, and output layers for action scoring:

```
def _model(self):
    inputs = Input(shape=self.input_shape)
    x = Flatten()(inputs)
    x = Dense(100, activation='elu')(x)
    x = Dropout(0.5)(x)
    x = Dense(50, activation='elu')(x)
    x = Dropout(0.5)(x)
    predictions = []
    for i in range(self.portfolio_size):
        asset_dense = Dense(self.action_size, activation='linear')(x)
        predictions.append(asset_dense)
    model = Model(inputs=inputs, outputs=predictions)
    model.compile(optimizer='adam', loss='mse')
    return model
```

The next step in the workflow is the conversion of network predictions into actionable portfolio weights. This process is implemented in the `nn_pred_to_weights` method, which maps the neural network's raw output into a valid portfolio allocation vector.

After the model generates a set of action scores for each asset (corresponding to possible actions such as *sit*, *buy*, or *sell*), the action with the highest score is selected for each asset. This selection is accomplished by taking the index of the maximum value along the action dimension for each asset. The resulting actions are then mapped to portfolio weights as follows: if the chosen action is *sit*, the corresponding asset receives a weight of zero; if the action is *buy*, the asset is assigned a positive weight proportional to the model's predicted score; and if the action is *sell*, the asset receives a negative weight, again scaled by the model's output.

To accommodate both long-only and long-short strategies, the method includes logic for adjusting weights. If short selling is not permitted, all negative weights are shifted upward so that the minimum weight becomes zero, ensuring non-negativity across the allocation vector. In either case, the weights are then normalized so that their sum equals one, maintaining a fully invested portfolio.

The code for this process is presented below:

```
def nn_pred_to_weights(self, pred, allow_short=False):
    weights = np.zeros(len(pred))
    raw_weights = np.argmax(pred, axis=-1)
    saved_min = None
    for e, r in enumerate(raw_weights):
        if r == 0:  # sit
            weights[e] = 0
        elif r == 1:  # buy
            weights[e] = np.abs(pred[e][0][r])
        else:
            weights[e] = -np.abs(pred[e][0][r])
    if not allow_short:
        weights += np.abs(np.min(weights))
        saved_min = np.abs(np.min(weights))
        saved_sum = np.sum(weights)
    else:
        saved_sum = np.sum(np.abs(weights))
    weights /= saved_sum
    return weights, saved_min, saved_sum
```

This conversion ensures that the model's discrete action space is faithfully translated into a continuous set of portfolio weights, respecting investment constraints and facilitating robust portfolio rebalancing in each decision period.

A key component of the RL agent is the action selection process, implemented in the `act` method. This method governs how portfolio weights are determined at each decision point by balancing the trade-off between exploration and exploitation using an epsilon-greedy strategy.

During training, the agent must occasionally explore new actions to avoid premature convergence to suboptimal policies. With probability ϵ, the agent engages in exploration by generating a random portfolio allocation. These exploratory weights are drawn from a normal distribution and adjusted to satisfy investment constraints. If short selling is not permitted, the weights are shifted to ensure all positions are non-negative. In all cases, the weights are normalized so that the total allocation sums to one, thereby respecting the budget constraint of the portfolio.

Conversely, with probability $1-\epsilon$, the agent exploits its learned policy by selecting actions based on the current state. The model predicts a set of action scores for each asset, which are subsequently converted to portfolio weights using the previously described `nn_pred_to_weights` method. This conversion ensures that the predicted actions are mapped into valid portfolio allocations.

The implementation of this process is as follows:

```
def act(self, state):
    if not self.is_eval and random.random() <= self.epsilon:
        w = np.random.normal(0, 1, size=(self.portfolio_size, ))
        saved_min = None
```

```python
        if not self.allow_short:
            w += np.abs(np.min(w))
            saved_min = np.abs(np.min(w))
        saved_sum = np.sum(w)
        w /= saved_sum
        return w, saved_min, saved_sum
    pred = self.model.predict(np.expand_dims(state.values, 0))
    return self.nn_pred_to_weights(pred, self.allow_short)
```

As training progresses, the exploration rate ϵ gradually decays, encouraging the agent to increasingly rely on exploitation of the learned policy. This mechanism promotes efficient learning by enabling broad exploration in the early training stages and greater focus on optimized allocations as the policy improves.

Following the prediction of portfolio weights by the actor network, an additional normalization step is performed to ensure the output weights remain within the prescribed range and collectively sum to one. This post-processing step is critical for maintaining valid portfolio allocations at all times.

```python
actor, _ = self.model.predict(np.expand_dims(state.values, 0))
actor = np.clip(actor, -1, 1)
actor /= np.sum(np.abs(actor))
return actor
```

Ultimately, the output of the action selection process is a vector of portfolio weights that are executed in the trading environment, enabling the agent to dynamically adjust its investment strategy in response to evolving market conditions.

Experience replay is a critical mechanism for stabilizing and improving the learning process in deep RL. The `expReplay` method implements this approach by leveraging a memory buffer of past experiences to perform off-policy Q-learning updates in batches. Each stored experience consists of the current state (`s`), the subsequent state (`s_`), the action taken, the reward received, and a terminal flag indicating whether the episode has ended (`done`).

During each training iteration, a batch of experiences is sampled from the memory buffer. For each experience in the batch, the method first reconstructs the original portfolio weights associated with the recorded action. It then computes the target Q-values for each possible action using both the immediate reward and the expected future reward, as determined by the critic network's estimate of the next state. The target for each action is computed as the sum of the observed reward and the discounted maximum predicted future value, in line with the standard Q-learning update rule.

If the episode has not terminated, the method utilizes the predicted Q-values from the next state to incorporate expected future rewards. The target Q-values for the batch are then blended with the model's current predictions using a learning rate parameter (`alpha`), ensuring stable and incremental updates. The model is trained on these updated Q-value targets by fitting the neural network for one epoch per experience.

To facilitate effective exploration early in training and convergence toward exploitation as learning progresses, the exploration rate (ϵ) is decayed after each training iteration, subject to a predefined minimum threshold. This adaptive schedule allows the agent to initially explore a diverse set of strategies before committing to the most promising policies discovered during training.

The implementation of experience replay and Q-learning updates is summarized in the following code:

```python
def expReplay(self, batch_size):
    def weights_to_nn_preds_with_reward(action_weights, reward, Q_star=np.zeros
        ↪ ((self.portfolio_size, self.action_size))):
        Q = np.zeros((self.portfolio_size, self.action_size))
        for i in range(self.portfolio_size):
            if action_weights[i] == 0:
                Q[i][0] = reward[i] + self.gamma * np.max(Q_star[i][0])
            elif action_weights[i] > 0:
                Q[i][1] = reward[i] + self.gamma * np.max(Q_star[i][1])
            else:
                Q[i][2] = reward[i] + self.gamma * np.max(Q_star[i][2])
        return Q
```

```
    def restore_Q_from_weights_and_stats(action):
        action_weights, action_min, action_sum = action[0], action[1], action[2]
        action_weights = action_weights * action_sum
        if action_min is not None:
            action_weights = action_weights - action_min
        return action_weights
    for (s, s_, action, reward, done) in self.memory4replay:
        action_weights = restore_Q_from_weights_and_stats(action)
        Q_learned_value = weights_to_nn_preds_with_reward(action_weights, reward)
        s, s_ = s.values, s_.values
        if not done:
            Q_star = self.model.predict(np.expand_dims(s_, 0))
            Q_learned_value = weights_to_nn_preds_with_reward(action_weights, reward, np.squeeze(Q_star))
        Q_learned_value = [xi.reshape(1, -1) for xi in Q_learned_value]
        Q_current_value = self.model.predict(np.expand_dims(s, 0))
        Q = [np.add(a * (1-self.alpha), q * self.alpha) for a, q in zip(Q_current_value, Q_learned_value)]
        self.model.fit(np.expand_dims(s, 0), Q, epochs=1, verbose=0)
    if self.epsilon > self.epsilon_min:
        self.epsilon *= self.epsilon_decay
```

Through this experience replay mechanism, the agent is able to efficiently reuse past experiences, mitigate correlations between consecutive samples, and achieve improved convergence and policy performance. The Q-learning targets calculated from each experience are used to update both the actor and critic networks, ensuring that both components of the model learn from observed rewards as well as anticipated future returns.

The following code demonstrates the practical instantiation and application of the RL agent for portfolio optimization. In this example, the number of assets is determined from the data, and the agent is initialized accordingly. The portfolio environment is created using historical adjusted closing prices.

A key aspect of this framework is the manner in which the dataset is utilized. Rather than dividing the data into distinct training and testing sets, the agent is trained and evaluated using a rolling window approach. At each time step, the model is trained sequentially on past data—specifically, using a lookback window of recent asset returns to construct the current state—while its performance is evaluated on subsequent, unseen data points at regular intervals. This continuous, episode-based methodology allows the agent to both explore and exploit information as new market data becomes available, closely mirroring a real-world trading scenario.

```
N_ASSETS = len(ASSETS)  # Number of assets in the portfolio
agent = Agent(N_ASSETS)
env = PortfolioEnvironment(prices=adj_close_prices)

lookback = 50  # Number of previous days comprising each state
t = 100        # Current time step for decision-making
state = env.get_state(t, lookback)
action_weights, saved_min, saved_sum = agent.act(state)
print("Action Weights:", action_weights)

reward, sharpe = env.get_reward(action_weights, action_t=t, reward_t=t+1)
print("Reward:", reward)
window_size = 180
episode_count = 100
batch_size = 32
rebalance_period = 30 # Portfolio weights updated every 30 days
```

The `lookback` parameter specifies the number of prior time steps (e.g., days) used to construct the state for the agent, allowing the model to capture temporal dependencies in the asset returns. For instance, a lookback of 50 means that each state input contains return data from the preceding 50 days. The variable `t` denotes the current time step at which the agent makes a portfolio decision. The `window_size` defines the typical length of the rolling window used for state construction during extended simulations. The `episode_count` represents the number of complete passes through the historical data during training, with each episode covering a distinct set of trading days. The `batch_size` determines how many past experiences are sampled from the replay buffer at each Q-learning update, which stabilizes and accelerates learning. The `rebalance_period` specifies how often the agent updates and rebalances its portfolio allocations, commonly set to every 30 trading days (approximately one month).

The training loop involves running the agent through 100 episodes using historical market data. At each decision point, the agent selects portfolio weights (actions) based on the current state, receives feedback in the form of a reward, and updates its internal model using the experience replay mechanism described earlier. This iterative process enables the agent to learn effective portfolio allocation strategies over time, balancing the trade-off between maximizing returns and controlling risk.

Upon completion of training, the agent's performance is evaluated over a separate testing period. During this stage, the agent applies the learned policy to new market data, making sequential trading decisions without further learning. The resulting portfolio returns are then compared against a benchmark strategy—typically the Equal Weights Portfolio—to assess the effectiveness of the RL approach.

The following code snippet illustrates how cumulative returns for the benchmark and the RL-optimized portfolio can be visualized:

```python
plt.figure(figsize=(12, 2))
plt.plot(np.array(result_equal_vis).cumsum(), label='Benchmark', color='grey', ls='--')
plt.plot(np.array(result_rl_vis).cumsum(), label='Deep RL portfolio', color='black', ls='-')
plt.legend()
plt.show()
```

The proposed framework employs an actor-critic architecture in conjunction with deep Q-learning. The model's input layer receives historical asset returns, which are processed through several dense layers for feature extraction. Two output heads are utilized: the actor, which determines the portfolio weights, and the critic, which evaluates the value of the current state-action pair. Action selection is managed by an epsilon-greedy policy, while the experience replay mechanism ensures that both recent and diverse historical experiences are leveraged during training.

The RL framework enables the agent to dynamically adjust portfolio weights in response to evolving market conditions, continuously improving its allocation strategy to maximize returns while managing risk. The joint optimization of portfolio allocation (via the actor) and value estimation (via the critic) promotes efficient learning and robust portfolio performance over time. Ultimately, this methodology demonstrates the potential of deep RL techniques for sophisticated, adaptive portfolio optimization in modern financial markets.

The experimental results highlight the effectiveness of this RL approach in optimizing allocations for the MAG 7 stocks over the chosen evaluation period. By leveraging historical return data and a deep actor-critic model, the agent was able to discover and exploit patterns in asset dynamics, yielding portfolio weights that maximize risk-adjusted performance.

4.5.2 Weights

Based on a policy acquired from previous interactions with the market environment, the RL agent produced portfolio weights for the MAG 7 stocks. With the option of giving the assets both long (positive) and short (negative) positions, the agent produces portfolio weights that total 1. The action weights generated by the agent for the MAG 7 stocks are as follows:

$$\text{Action Weights} = [0.8204, -0.0308, -0.3362, 0.1531, 0.0114, 0.0173, 0.3647]$$

- **AMZN (0.8204)** A significant positive weight suggests that the RL agent prefers Amazon, likely due to its projected high return with low risk. The agent views Amazon as one of the main contributors to returns, benefiting from its growth potential.

- **GOOGL (-0.0308)** A small negative weight indicates that the RL agent is betting against Google, suggesting lower expected returns or more volatility compared to other assets.
- **AAPL (-0.3362)** The agent has a relatively significant negative weight on Apple, possibly due to its more stable return profile or lower risk-adjusted returns compared to other assets.
- **MSFT (0.1531)** A small positive weight on Microsoft suggests that the RL agent views Microsoft as a medium-risk asset with reasonable return potential.
- **TSLA (0.0114)** A small positive weight on Tesla reflects the agent's conservative view of this volatile stock. The weight implies that Tesla has less impact on the portfolio, given the inherent risk linked to its erratic price movements.
- **NVDA (0.0173)** Although its weight is relatively low compared to Amazon, a small positive weight suggests that Nvidia remains an important asset in the portfolio.
- **META (0.3647)** A significant positive weight on Meta indicates that the agent expects strong returns from Meta, likely due to its favorable risk-return profile or potential upside relative to other equities.

With a clear bias towards high-volatility stocks like Amazon and a more conservative approach to large-cap, lower-volatility stocks like Apple and Google, these weights show the decision-making process of the RL agent.

4.5.3 Performance Evaluation

Using various important financial metrics, returns, volatility, Sharpe ratio, alpha, and beta, we assess the performance of the RL Agent versus the Equal Weights Portfolio in this part. These measures give a thorough knowledge of how every portfolio fared throughout the testing time.

Ignoring risk-adjusted asset returns, the Equal Weights portfolio's volatility of 0.0211 indicates more notable changes in asset values caused by the equal capital distribution. At 0.0109, the RL agent's volatility is lower than that of the equal weights portfolio. The RL agent's ability to adjust allocations based on asset performance leads to better risk management and reduced portfolio volatility.

The Equal Weights Sharpe ratio of 0.5841 for suggests positive returns, though it carries more risk. By comparison, the RL portfolio, which has a Sharpe ratio of 0.7148, beats the Equal Weights portfolio in risk-adjusted returns. A higher Sharpe ratio means that the RL agent outperforms the equal weights portfolio in terms of return relative to the volatility taken on by the portfolio.

The Equal Weights portfolio shows it produces no excess return above the market with an alpha of -0.0. Conversely, the RL agent has a small positive alpha of 0.0002, suggesting that its dynamic rebalancing approach is generating a little excess return.

With a beta of 1.0, the Equal Weights portfolio reflects average market risk, moving similar with the market. In comparison, the RL portfolio has a beta of 0.3937, indicating it is less sensitive to market fluctuations and carries lower systematic risk. This demonstrates the agent's ability to reduce exposure to broad market movements and manage risk through strategic asset allocation.

The RL Agent beats the Equal Weights Portfolio in risk-adjusted returns (Sharpe ratio), volatility, and beta (systematic risk). The RL agent's dynamic portfolio optimization approach helps it to control volatility more effectively and attain better risk-adjusted returns. Although the Equal Weights Portfolio offers a steadier and predictable return, it does so at the expense of greater volatility and no excess return above the market (alpha). The RL Agent's capacity to reduce volatility while generating reasonable returns and control risk through calculated weight changes indicates it provides a more sophisticated and efficient method to portfolio optimization than the simple equal-weight allocation approach.

4.5.4 Cumulative Return

The cumulative returns of the RL portfolio and the Benchmark (Equal Weights Portfolio) are compared in the Figure 10. The benchmark (grey dashed line) and Deep RL portfolio (black solid line) show how both portfolios performed over time.

At first, both portfolios follow a similar path, with returns varying around a slight upward trend. With notable oscillations between profits and losses across the early and mid-periods, the benchmark portfolio shows significant volatility. These fluctuations highlight the limitations of the Equal Weights approach, which ignores adaptive rebalancing and does not consider evolving market conditions.

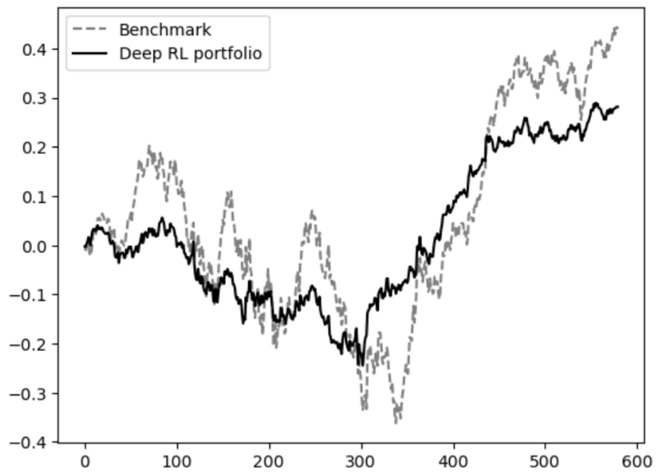

Figure 10: Cumulative Returns RL Agent vs. Equal Weights

By contrast, the RL portfolio significantly diverges after the halfway point. The Deep RL agent's dynamic reallocation of asset weights takes advantage of changing market trends, producing more stable and sustained cumulative growth. This steady upward trend in the second half of the period suggests the agent effectively found and exploited good market dynamics. Though not always better than the benchmark at every moment, the RL portfolio shows more resilience during falls in the graph and a more strong recovery pattern. This implies that the RL agent can still engage in upward movements while better able to reduce negative risk.

Overall, the graph emphasizes the advantages of RL in portfolio management. More efficient risk management and better long-term returns than the static benchmark result from the RL portfolio's capacity to respond dynamically to market signals. These findings support the possibility of RL-based strategies negotiating turbulent market conditions and producing better cumulative performance over time.

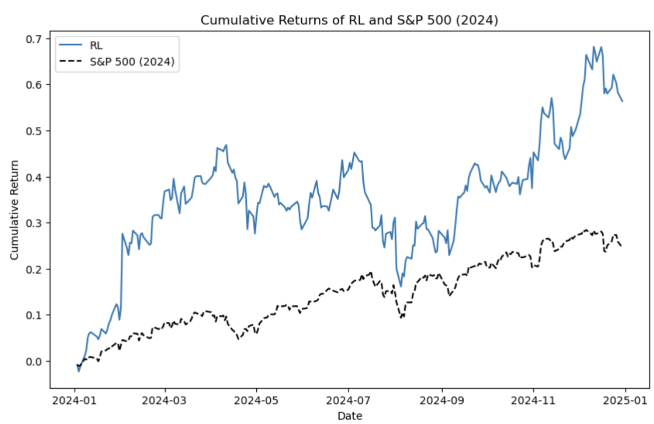

Figure 11: Cumulative Returns of RL Portfolio

5 Conclusion

This study explored the evolution and application of portfolio optimization techniques, beginning with the foundational concepts in financial theory and advancing to modern applications of artificial intelligence (AI) and maching learning (ML) methods. A key component of economic decision-making, portfolio optimization seeks to balance portfolio risk and return according to individual investors' objectives. Beginning with a discussion on conventional techniques like Harry Markowitz's Modern Portfolio Theory (MPT), we set the stage for developing the critical link between risk and return. However, MPT's assumptions fell short of fully accounting the true systemic dynamic reality of financial markets, so more sophisticated models were investigated.

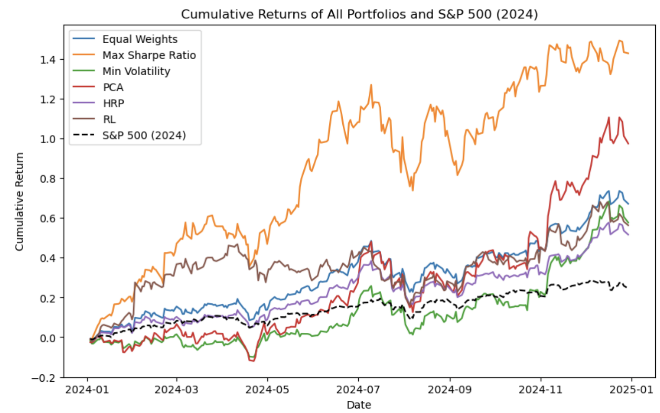

Figure 12: Cumulative Returns of all Portfolios Using MAG 7 Stocks and S&P 500 (2024)

We evaluated three conventional portfolio strategies to build on this foundation: Equal Weights, Maximum Sharpe Ratio, and Minimum Volatility. Each model offers distinct perspectives on the risk-return relationship. While the Minimum Volatility Portfolio emphasizes downside protection, the Maximum Sharpe Ratio Portfolio aims best risk-adjusted returns. Though unoptimized, the Equal Weights Portfolio is a reasonable benchmark for comparison analysis because of its simplicity and built-in diversity.

We then looked at multi-factor models, including the Capital Asset Pricing Model (CAPM) and the Fama-French Three-Factor Model, to evaluate how systematic risk factors propel returns. These models offered insightful analysis showing that market exposure mainly influences all portfolios. Although CAPM stresses market risk, the Fama-French model adds extra elements—size and value—that more accurately account for asset performance. Though the models also pointed out their shortcomings, such as the failure to consistently capture abnormal returns, we clearly demonstrated these models validated the significant influence of market exposure in driving portfolio returns.

The evolution of financial theory prompted the research of more complex methods such as Principal Component Analysis (PCA) and Hierarchical Risk Parity (HRP). PCA simplifies portfolio optimization utilizing dimensionality reduction, finding necessary components responsible for the most notable variation in asset returns. This approach makes a more efficient asset selection possible by emphasizing the most powerful market drivers. Conversely, HRP offers a robust and diversified portfolio using hierarchical clustering and risk parity, therefore lowering the concentration of risk/volatility. These modern techniques, especially for complex, multi-asset portfolios, indicated the potential to improve portfolio construction.

This work then covered portfolio optimization using Reinforcement Learning (RL) as a creative tool. Unlike traditional methods, RL dynamically adjusts portfolio allocations based on real-time market data, enabling the model to learn and optimize asset distribution over a defined time frame. This adaptability makes RL relevant in unstable markets, where traditional models struggle to forecast sudden changes in asset values.

Empirical results from forward-testing simulations using the market returns from the whole 2024 year verified the effectiveness of these optimization strategies. Figure **??** shows the cumulative returns of all portfolios. Though with significant volatility, the Maximum Sharpe Ratio Portfolio stands out by attaining the greatest total return over the year. This emphasizes the natural trade-off between risk and reward—periods of strong outperformance are followed by more notable drawdowns—and therefore corresponds with its aim of maximizing risk-adjusted returns. PCA and RL portfolios show competitive performance, surpassing the SP 500 for long periods and offering proof that advanced, adaptive, and risk-aware strategies can provide better returns under specific market circumstances. The Minimum Volatility and HRP portfolios had a more consistent pattern. Though their overall returns are lower than those of the Max Sharpe and PCA portfolios, their low volatility makes them attractive for risk-averse investors. A good starting point for comparison, the Equal Weights portfolio closely tracks the middle of the pack, providing reasonable returns and volatility.

Finally, this study emphasized the possibility of advanced computational tools, including PCA, HRP, and RL, even as it stressed the need for classical portfolio optimization techniques. Though still applicable, traditional models such as the Maximum Sharpe Ratio and Minimum Volatility portfolios cannot match the dynamic and flexible approach to portfolio management offered by the combination of AI/ML techniques. As more technologically sophisticated financial market analysis techniques are developed, combining economic theory and modern AI/ML technology provides a robust framework for building more resilient and efficient portfolios.

References

[1] Wikipedia contributors. Dow jones industrial average — wikipedia, the free encyclopedia, 2024. Accessed: 2024-04-27.

[2] Marcos López de Prado. Building diversified portfolios that outperform out-of-sample. *SSRN Electronic Journal*, 2016. SSRN preprint, https://ssrn.com/abstract=2708678.

[3] Fidelity Viewpoints. What are the magnificent 7 stocks? https://www.fidelity.com/learning-center/smart-money/magnificent-7-stocks, 2024. Accessed: 2024-04-27.

[4] Stefan Jansen. *Machine Learning for Algorithmic Trading*. Packt Publishing, Birmingham, UK, 2nd edition, 2020. Chapter 22: Deep Reinforcement Learning – Building a Trading Agent.

[5] Steven Johnson. *Python for Quant Finance: Building Financial Models and Trading Algorithms*. Independently Published, 2021. Check actual publication year and publisher if available.

[6] Eryk Lewinson. *Python for Finance Cookbook*. Packt Publishing, Birmingham, UK, 2nd edition, 2022.

[7] Hariom Tatsat, Sahil Puri, and Brad Lookabaugh. *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly Media, Sebastopol, CA, 2021.