Project 3

Group 15 - Fatima, Husain, Stallon, Shadi

April 2025

1 Introduction and Context

In Project 1, we used linear programming to determine the optimal allocation of a daily investment budget across different asset classes based on expected return and risk. Project 2 extended this by introducing a scoring system to decide which assets to hold or sell based on momentum, volatility, and past returns. Both models approached investment through the lens of allocation and decision-making.

This project takes a different approach. Rather than optimizing how to allocate money or making binary hold/sell decisions, our goal is to identify which stocks are predictable, and what features can help identify their trend. In other words, we aim to filter stocks not just by expected returns, but by the reliability of their underlying patterns.

To do this, we use ElasticNet regression with L1 and L2 regularization, which helps select a small number of useful features while eliminating noise. This model eliminates noisy or weak signals while retaining features that meaningfully contribute to return predictability. Features such as momentum (MACD), volatility (MAD), liquidity (Turnover Ratio), size (Market Capitalization), seasonality (Month), and macroeconomic influences (Interest Rate) are extracted and standardized. ElasticNet balances feature sparsity and stability, improving both the interpretability and robustness of the model. Beyond identifying features, we introduce a trend-scoring mechanism. For each stock, we examine the recent behavior of its top predictive features over the last three months. Favorable movements such as increasing momentum signals or decreasing volatility measures contribute positively to a stock's trend score.

Finally, we integrate these trend scores into a Markowitz portfolio optimization model. The original Markowitz optimization focuses purely on minimizing portfolio variance for a given expected return and leverages historical return and risk data. In contrast, our extended model adjusts the optimization objective to also reward allocations toward stocks exhibiting favorable trends for the feature recognized as being predictive for each individual stock.

By embedding predictive feature selection and trend scoring into portfolio allocation, we move beyond historical returns alone. Our model builds portfolios that are not only based on well-known optimization model, but also dynamically informed by the trend of deduced predictive features. This feature-driven enhancement improves portfolio robustness, enables more forward-looking investment decisions, and creates a framework that adapts to changing market conditions rather than simply extrapolating the past.

2 Variables and Parameters

This section outlines the variables and parameters used in the model. The setup is based on the same type of daily stock data that we used in Projects 1 and 2, but instead of focusing on allocation or decision rules, we now build a predictive model using these variables.

Description	**Symbol**	**Units**
Number of stocks	n	-
Number of time points	T	Days
Feature matrix (combined across all stocks)	$X \in \mathbb{R}^{T \times p}$	-
Target return vector	$y \in \mathbb{R}^{T}$	-
Model coefficients	$\beta \in \mathbb{R}^{p}$	-
L1-L2 ratio parameter	λ	-
ElasticNet regularization parameter	α	-
Stock index	$i \in \{1, \ldots, n\}$	-
Feature group for stock i	$R_i \subset R$	-
Weightage of stock allocation	$x \in \mathbb{R}^{n}$	-
Covariance matrix	Σ	-
Trend-score regularization parameter	λ_{trend}	-

Each row of the matrix X represents a time step, and each column is a different feature from a specific stock. The corresponding entry in y is the return we aim to predict. The coefficient vector β is learned using convex optimization. A non-zero coefficient indicates that the associated feature contributes to return prediction.

Feature Definitions:

- Return: returns associated with the stock
- MACD: moving average convergence divergence
- MAD: mean absolute deviation (volatility)
- BB distance: metric derived using the Bollinger bands
- Turnover ratio: trading activity relative to shares outstanding
- Market cap: market capitalization
- Month: encoded as a numeric variable from 1 to 12
- Interest rate: macroeconomic indicator

We made the following assumptions while building and using the model:

1. The model assumes that future return can be reasonably estimated using a linear combination of the selected features.
2. All input features are based on information that would have been available before the return we are trying to predict.
3. Features from each stock are handled separately, but we combine everything into one matrix when training the model.

4. Each row in the feature matrix R corresponds to a different time step and is treated as independent.

5. All features are standardized so that they have mean zero and standard deviation one before fitting the model.

6. The value of λ and λ_{trend} is treated as a tuning parameter and is adjusted through trial and error.

7. The results are based on analysis of each individual stock's data since 2020 January till 2025 March.

3 Building the model

3.1 ElasticNet Regression For Feature Extraction

The model uses ElasticNet regression to assign coefficients to the features based on their predictive contribution. The ElasticNet regression combines features of both L1 and L2 regularization. Intuitively, we can base our model on just L1 regularization, since that would help us zero out the features with minimal contribution. However, since the number of factors affecting the returns of stocks are innumerable, it is possible that the return is manipulated by a combination of these features; in this case, using just L1-regularization would inhibit this possibility by forcing and singling out features. In order to avoid this, we add L2-regularization to strike a balance between sparsity and stability.

Our model consumed data from the Yahoo Finance API and computed all the necessary values to build a feature matrix, where each row corresponded to data derived from one month of observations, and each column represented a feature used to predict the stock's return in the following month. Using monthly-level features helps avoid overfitting to noisy daily data and emphasizes broader trends rather than short-term fluctuations.

The features included:

- **MACD (Moving Average Convergence Divergence)**: MACD is a momentum indicator which is calculated as the difference between the 12-day and 26-day exponential moving averages (EMAs) of the stock's daily closing price. The main idea behind using exponential average of the asset price is to smooth out any rapid fluctuations and capture the essential trend in the price. Furthermore, exponential average puts a higher weightage on the more recent prices compared to the prices recorded further in the past; this helps to put a higher importance on the most recent price changes. The value of MACD on the last day of the month was used, computed using the *ta* library in Python.

- **Bollinger Band Distance**: Bollinger bands are a financial metric that provide a dynamics range of an asset's moving average and captures the variability in its price. The bollinger bands consist of three lines namely, the middle, lower and upper band. The middle band tracks the 20-day simple moving average of the asset's closing price.
  - Middle band: A 20-day simple moving average (SMA) of the stock's closing prices.
  - Upper band (U_t): The middle band plus two times the standard deviation of the last 20 days of prices.

  - Lower band (L_t): The middle band minus two times the standard deviation of the last 20 days of prices.

  When market volatility increases, the standard deviation of prices also increases, as a result of which the upper and lower Bollinger Bands widen and create a broader channel around the stock's moving average. This widening indicates that prices are experiencing greater variability, often corresponding to periods of uncertainty, heightened trading activity, or major market events. Similarly, when the volatility decreases, the standard deviation decreases, and the Bollinger Bands contract and form a narrower channel around the moving average. Tight bands suggest a period of low volatility and market consolidation, often preceding a significant breakout in either direction. To quantify this metric, we use the Bollinger band distance which is defined as the relative distance between the stock's closing price and the lower Bollinger Band scaled by the width of the Bollinger Bands:

$$\text{BB\_Distance} = \frac{C_t - L_t}{U_t - L_t}$$

  where C_t is the closing price, L_t is the lower band, and U_t is the upper band.

- **MAD (Mean Absolute Deviation)**: MAD is a statistical measure of the fluctuations in price regardless of the direction of the trend. It is computed over daily intraday returns within each month. The intraday return for each day is:

$$r_t = \frac{C_t - O_t}{O_t}$$

  and the monthly MAD is calculated as:

$$\text{MAD} = \frac{1}{n}\sum_{t=1}^{n} |r_t - \bar{r}|$$

  where $\bar{r}$ is the average intraday return for the month, r_t is the return on day t, C_t is the closing price on day t, and O_t is the opening price on day t.

- **Turnover Ratio**: The turnover ratio captures the trading activity of a particular asset relative to the number of its shares available in the market, i.e., it tells us how frequently the company's shares are traded relative to the total available supply. It is calculated as the ratio of daily traded volume to the number of shares outstanding, averaged across the month:

$$\text{Turnover} = \frac{1}{n}\sum_{t=1}^{n} \frac{V_t}{\text{Outstanding shares}}$$

  where, V_t is the number of shares traded on day t, n is the number of days in the trading month, and "Shares Outstanding" refers to the total number of shares available to the public for buying.

  The turnover ratio indirectly measures the liquidity and serves as a tracker for investor sentiment. A high turnover ratio usually indicates strong market interest, while a stable and lower turnover ratio indicates that the shares are being traded at a consistent rate without any large buying or selling swings.

- **Market Capitalization**: It is the total value of the company's outstanding shares in the market. It is computed daily as the product of the closing price and the number of shares outstanding, with the monthly feature being the average across all days:

  $$\text{MarketCap} = \frac{1}{n}\sum_{t=1}^{n} C_t \times \text{Outstanding shares}$$

  where C_t is the closing price on day t.

  For our model we take the average Market Cap over each month in order to smoothen the rapid daily fluctuations and represent the stock's overall size during that period. Generally, large-cap stocks, i.e. stocks with a high market cap are considered to be relatively stable and less susceptible to sudden price changes. Conversely, small-cap stocks usually observe higher volatility and are comparatively unpredictable.

- **Interest Rate**: The end-of-month value of the U.S. 10-Year Treasury Yield, divided by 100 to convert from percentage to decimal form. The interest rate is not associated with each individual stocks, rather it is a common metric across all assets, and generally guides the investor's confidence in future economic growth.

- **Month**: An integer between 1 and 12 representing the calendar month, allowing the model to capture seasonal trends.

The target variable was the next month's return, computed as:

$$\text{Monthly Return} = \frac{\text{Closing Price (last day)} - \text{Opening Price (first day)}}{\text{Opening Price (first day)}}$$

This value was shifted backwards so that the features from month t predict the return in month $t+1$, making the task forward-looking.

After computing all features and removing rows with missing data, we split the dataset into training and test sets in chronological order, with 70% used for training and 30% for testing. We standardized all features and the target to have mean zero and standard deviation one before fitting the model. The model was implemented using the *scikit-learn* library in Python.

We trained the ElasticNet regression model, solving the following optimization problem:

$$\min_{\beta} \quad \|X\beta - y\|_2^2 + \alpha\left(\lambda\|\beta\|_1 + (1-\lambda)\|\beta\|_2^2\right)$$

where:

- X is the standardized feature matrix,
- y is the standardized target (next month's return),
- β is the vector of model coefficients,
- α controls the overall regularization strength ($\alpha = 0.01$),
- λ balances the relative weight between L1 and L2 penalties ($\lambda = 0.7$).

After training, we evaluated model performance by comparing the predicted and actual directions of month-to-month return changes. We measured how often the predicted return direction matched the true direction, referring to this metric as *sync*.

For stocks with high sync counts (above 70%), we further analyzed the learned model coefficients. We identified the top three most influential features for each stock by sorting coefficients by absolute magnitude. These features revealed which aspects of the stock's behavior were most predictive of future returns. For example, a strong negative coefficient on market capitalization implied that smaller companies tended to be associated with higher predicted returns.

3.2 Application using Markowtiz Model

The Markowitz model is a well-known optimization model that solves the problem of portfolio allocation using convex optimization. The model help construct an optimal portfolio by minimizing risk for a given level of expected return. Formally, it can be written as:

$$\min_{x} \quad x^T \Sigma x \quad \text{subject to} \quad \sum_i x_i = 1, \quad x^T \mu \geq \text{target return}$$

where:

- x is the portfolio weight vector,
- Σ is the covariance matrix of asset returns,
- μ is the vector of expected returns (monthly average return in our case).

We extended this Markowitz model to incorporate tracking the top 1, 2, and 3 features for each asset (as calculated using the technique mentioned above), into 3 separate models to compare their performances against the conventional Markowtiz model. To extend the analysis beyond feature identification, we introduced a trend-scoring mechanism. For each stock, we evaluated the recent trend of its top features over the last three months. Features showing favorable trends (upward for momentum-related indicators or downward for volatility indicators) contributed positively to a stock's trend score.

More precisely, for each top predictive feature associated with a stock, we extracted its last three monthly values and computed the relative change between the first and last value. A feature was considered to be trending favorably if:

- For momentum-related features (such as MACD and Turnover Ratio), the feature increased over the period, suggesting strengthening momentum or increasing trading activity.
- For volatility or risk-related features (such as MAD, Market Capitalization, BB Distance, and Interest Rate), the feature decreased over the period, suggesting declining risk, improving stability, or favorable macroeconomic conditions.

Each favorable trend contributed a score of 1 to the stock. Neutral or insufficiently strong trends contributed partially, and unfavorable trends contributed little to the score. The final trend score for each stock was computed as the average of its favorable trends across its top features, normalized between 0.5 (neutral) and 1 (strongly favorable). Thus, stocks with strong, consistent feature improvements received higher trend scores, while stocks with weak or inconsistent feature movements scored closer to neutral.

Finally, we incorporated these trend scores into a convex optimization framework to construct optimized portfolios. We formulated a portfolio optimization problem that minimizes portfolio variance while rewarding allocations to stocks with strong trend scores, thereby combining classical risk management principles with dynamic feature-driven adjustments.

$$\min_x x^T \Sigma x - \lambda_{\text{trend}} (\text{trend scores}^T x)$$

subject to:

$$\sum x_i = 1, \quad x^T \mu \geq \text{target return}$$

where:

- x is the vector of portfolio weights,
- Σ is the covariance matrix of monthly returns,
- μ is the vector of average monthly returns,
- λ_{trend} controls the influence of the trend bonus ($\lambda_{trend} = 0.05$).

The target return is defined as the average of the individual expected monthly returns across all selected stocks. Specifically, for each stock, we compute the expected monthly return as the mean of its historical monthly returns over the training period. The target return is then calculated by taking the simple arithmetic average of these expected returns. This approach ensures that the optimized portfolio must perform at least as well as the mean return of the investment universe under consideration, based on historical data.

We solved this optimization problem for different scenarios: pure variance minimization, and models that embedded the trend scores based on the top-1, top-2, and top-3 features. This approach not only identifies stocks that are more predictable but also constructs systematic portfolios that exploit this predictability to achieve superior risk-adjusted performance.

3.3 Methodology

The model was implemented in Python using *scipy, scikit-learn, numpy, pandas, ta, and yfinance* libraries. A set of 50 arbitrarily selected stocks was analyzed for ElasticNet-based feature selection, as well as portfolio allocation. The tickers and the names of the stocks used have been appended at the end of the document.

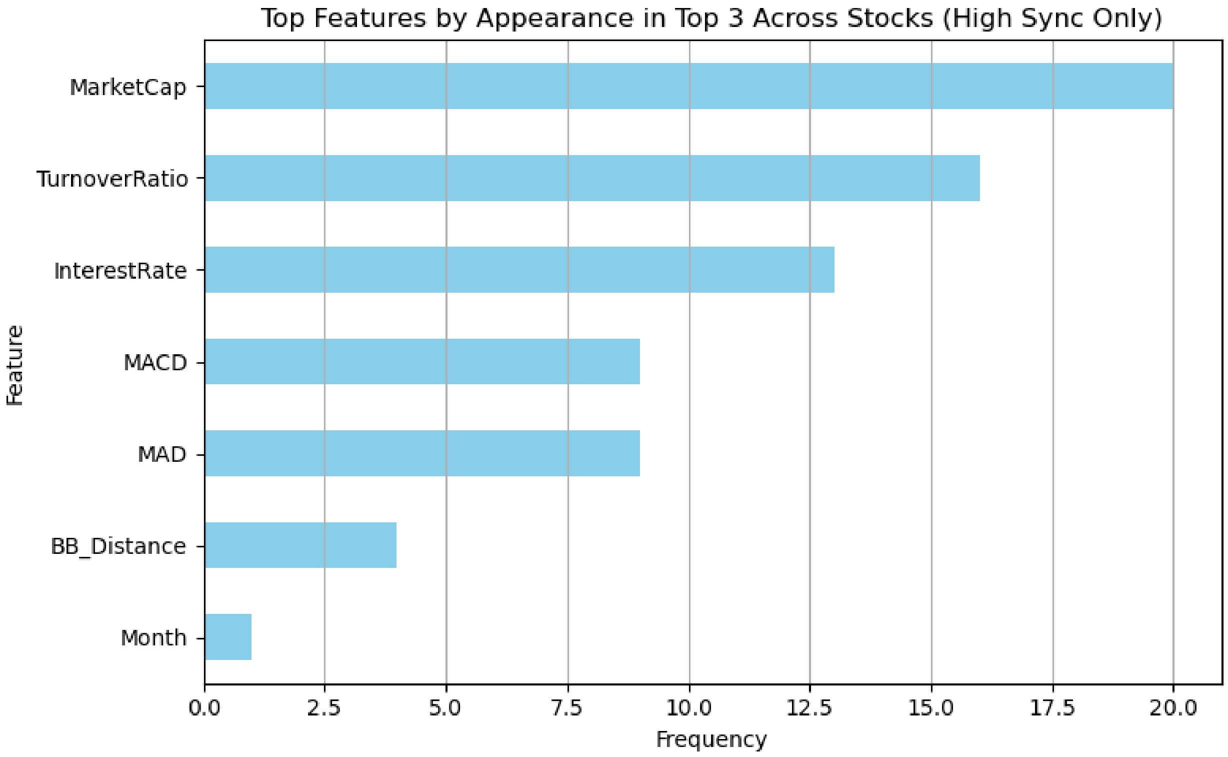
The model for Markowitz portfolio optimization was trained on stock data retrieved from January 2020 to March 2024 and tested on data from April 2024 to March 2025. The stock data was downloaded from Yahoo Finance API to ensure deterministic results, which otherwise would be unpredictable due to repeated API calls. Once the data is downloaded, it is used to reproduce the results.

The features that are used for the ElasticNet Regression-based feature selection are all calculated using the data extracted from the Yahoo Finance API. The features as well as the target variable (returns) are scaled to ensure that all the features are treated equally and the model selects features based purely on their predictive power, not their raw size.

The inbuilt function *ElasticNet* defined in the *sklearn.linear_ model* library is used for the ElasticNet regression. The function *minimize* defined in *scipy.optimize* is used to solve the Markowitz models.

4 Reporting Results

4.1 ElasicNet Regression

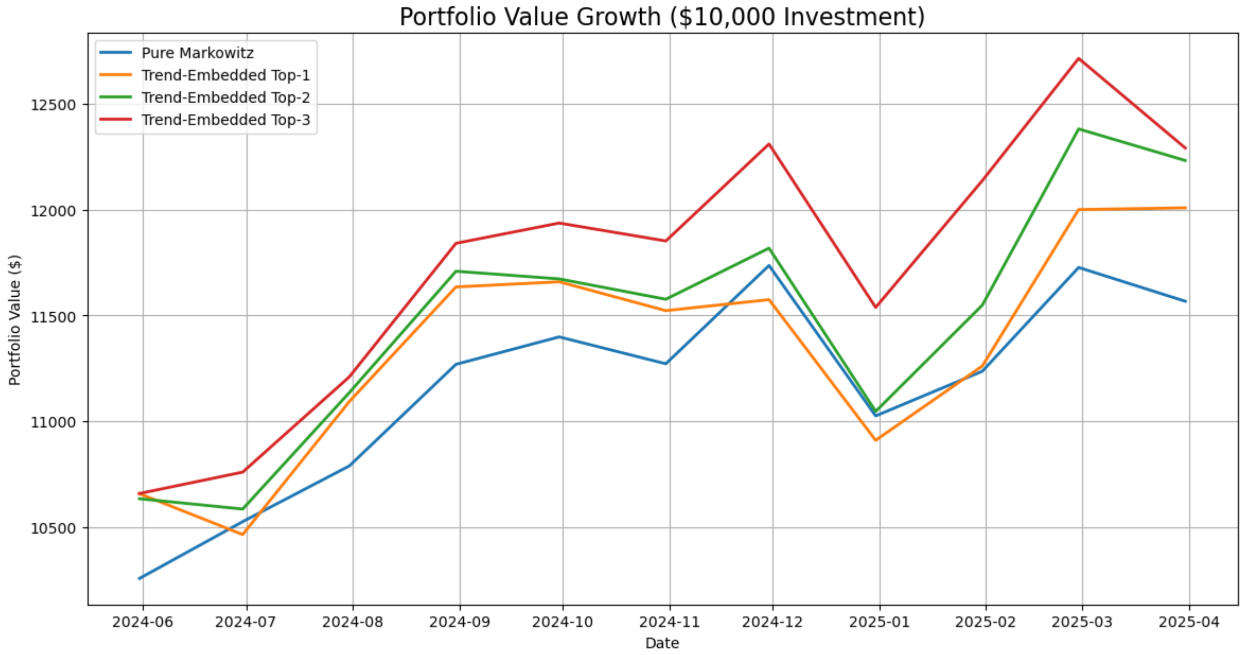


After utilizing Lasso Regression and ElasticNet regression to find out what the three best respective predictors for each stock's evolution are, we can broaden the scope by looking at the features that appear the most in all of the stocks to get a better sense of what features are the most desirable ones when trying to find the predictability of stocks.

When looking at the graph above, we notice that from the results obtained from our code, the three most frequent features among all stocks are Market Capitalization (appears 20 times), Turnover Ratio (appears 16 times), and Interest Rate (appears 13 times), and that the three least frequent features among all of said stocks are MACD (appears 9 times), Bollinger Band Distance (appears 4 times) and Month (appears 1 time). This tells us that indicators like Market Capitalization and Turnover Ratio, which are features whose primary focus is capturing the "how" part of price behavior with a multifaceted view of a company's market presence, liquidity, and the economic climate, are better predictors of a stock's growth/decay than those like Bollinger Band Distance and MACD, which are mainly focused on "why" part of said behavior from a fundamental and economic perspective, as they measure aspects like market behavior, momentum and volatility, which gives us a more technical perspective of a stock's performance rather than a theoretical one.

From said graph, we can infer that indicators that give us a better prediction of a stock's evolution revolve around a company's financial robustness, whereas those that are less effective at doing so are more technical and seasonal than fundamental and economic.

4.2 Original vs. Extended Markowitz model



Portfolio	Total Return	Monthly Volatility
Pure Portfolio	15.67%	3.16%
Top-1 Portfolio	20.09%	4.01%
Top-2 Portfolio	22.34%	4.19%
Top-3 Portfolio	22.93%	4.09%

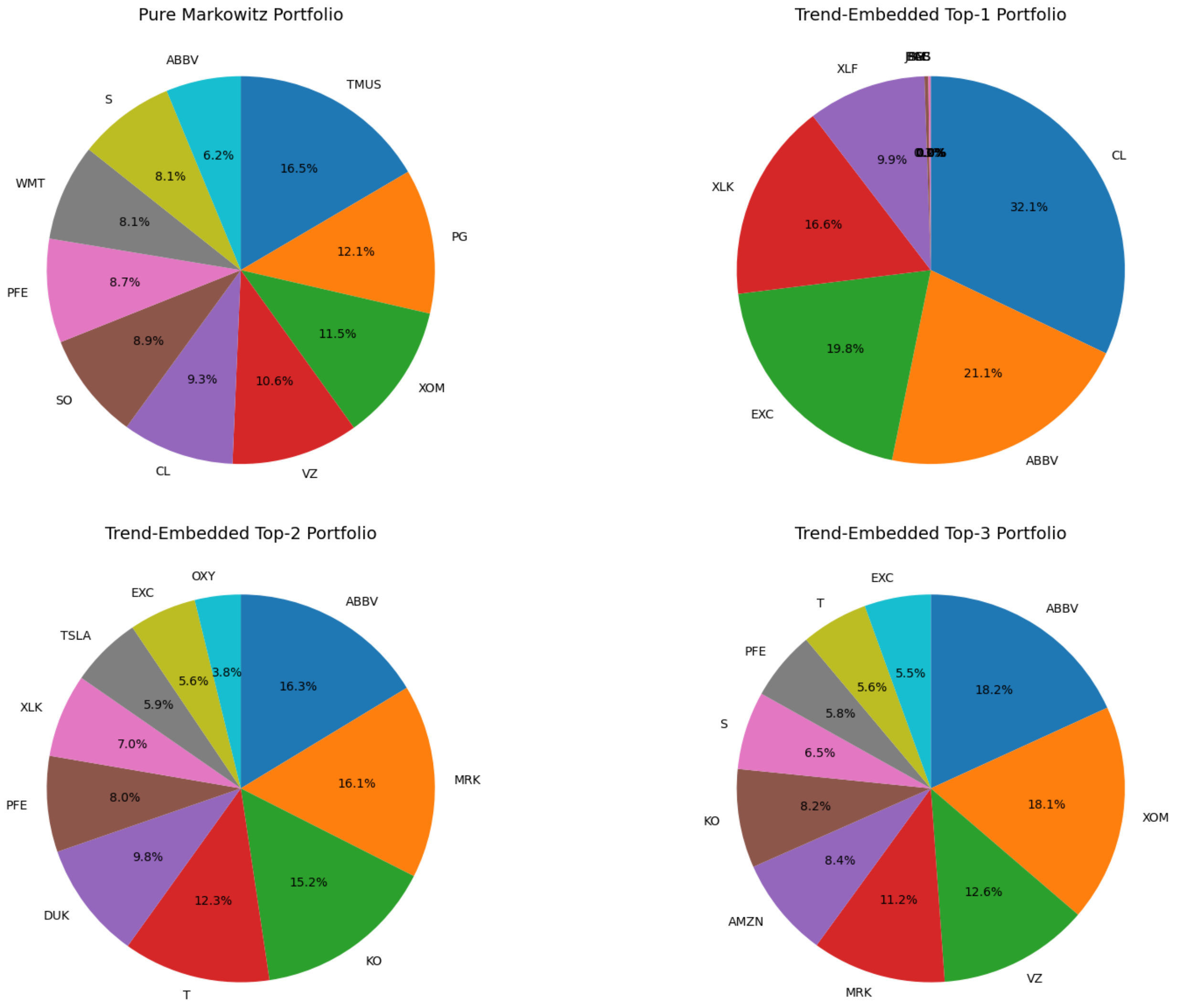
Using the original Markowitz model, and three other variations of the Markowitz model, each embedded with top 1, 2 and 3 features associated with each stock, we individually allocated an investment of $10,000 from a set of arbitrarily selected stocks. The above plot tracks the value of the portfolios allocated by each model, every month from June 2024 to April 2025. While the overall trend in the portfolio value of all the allocations follows a similar trend, the embedded models outperform the pure Markowitz model. While the Top-1 and Top-2 models observe close competition against the pure Markowitz model, the Top-3 model consistently outperforms the pure Markowitz model. The total return from each portfolio over the 12 months further highlights a stark disparity in the overall return generated by the pure vs. embedded models. The best return is given by the Top-3 model with a total return of 22.93% over 12 months. We also observe a rise in monthly volatility of the Top-3 model compared to the pure portfolio; however, a 1% increase in volatility isn't drastic and an increase of 7% at the expense of 1% higher volatility seems reasonable.

5 Appendix

List of stocks used:

- AAPL — Apple Inc.
- MSFT — Microsoft Corporation
- NVDA — NVIDIA Corporation
- AMZN — Amazon.com, Inc.
- GOOG — Alphabet Inc. (Class C)
- JNJ — Johnson & Johnson
- PFE — Pfizer Inc.
- MRK — Merck & Co., Inc.
- GSK — GSK plc
- ABBV — AbbVie Inc.
- JPM — JPMorgan Chase & Co.
- C — Citigroup Inc.
- BAC — Bank of America Corporation
- GS — The Goldman Sachs Group, Inc.
- AXP — American Express Company
- KO — The Coca-Cola Company
- PG — The Procter & Gamble Company
- WMT — Walmart Inc.
- COST — Costco Wholesale Corporation
- CL — Colgate-Palmolive Company
- XOM — Exxon Mobil Corporation
- CVX — Chevron Corporation
- OXY — Occidental Petroleum Corporation
- SLB — Schlumberger Limited
- NEE — NextEra Energy, Inc.
- BA — The Boeing Company
- CAT — Caterpillar Inc.
- DE — Deere & Company
- GE — General Electric Company
- UPS — United Parcel Service, Inc.

- TSLA — Tesla, Inc.
- MCD — McDonald's Corporation
- HD — The Home Depot, Inc.
- LVMUY — LVMH Moët Hennessy Louis Vuitton SE (ADR)
- NKE — NIKE, Inc.
- DUK — Duke Energy Corporation
- SO — The Southern Company
- AEP — American Electric Power Company, Inc.
- EXC — Exelon Corporation
- FCX — Freeport-McMoRan Inc.
- BHP — BHP Group Limited
- LIN — Linde plc
- ALB — Albemarle Corporation
- NEM — Newmont Corporation
- T — AT&T Inc.
- VZ — Verizon Communications Inc.
- TMUS — T-Mobile US, Inc.
- S — SentinelOne, Inc.
- XLF — Financial Select Sector SPDR Fund (ETF)
- XLK — Technology Select Sector SPDR Fund (ETF)



Pure Markowitz Portfolio

Trend-Embedded Top-1 Portfolio

Trend-Embedded Top-2 Portfolio

Trend-Embedded Top-3 Portfolio

The figure above shows the allocation of our initial investment into the stocks using the four different models used for our analysis.