Project 2: Optimizing Stock Portfolio to Match Index

Evan Hadd (eh28834)
Vishwa Patel (vp8792)
Sankeerth Viswanadhuni (vps386)
Oliver Gault (osg249)
Nawen Deng (nd9352)

Introduction:

The objective of this research is to determine the optimal number of component stocks needed to construct a portfolio that closely replicates the NASDAQ-100 Index. Selected for its historical record of outperformance compared to broader market indices, the NASDAQ-100 offers a balanced mix of high-growth companies that collectively deliver strong returns. This index represents a benchmark for investors seeking exposure to leading companies, especially in technology and other innovative sectors, making it a desirable foundation for mutual funds focused on long-term growth.

For our clients, building a portfolio that closely tracks the NASDAQ-100 allows access to a diversified investment strategy with a proven record of stability and growth. The challenge lies in creating a portfolio that mirrors the index's performance without requiring investments in all 100 components. Doing so can lead to operational complexities, increased transaction costs, and potential over-diversification, which may dilute returns. By finding the minimal number of stocks that can effectively track the index's performance, we can enhance portfolio efficiency and allocate resources more effectively, maximizing returns for our clients.

This research is relevant not only for constructing a high-performing mutual fund but also for streamlining fund management practices. Identifying an optimal subset of stocks allows us to simplify rebalancing processes, minimize costs, and potentially reduce the volatility associated with over-diversified portfolios. Furthermore, our approach aligns with a growing demand from investors who seek both transparency and strategic selectivity in their investments. Through careful analysis and modeling, this report aims to outline a methodology for achieving an efficient portfolio composition that remains in close alignment with the NASDAQ-100 Index, delivering stable returns that support our clients' financial goals and uphold our firm's commitment to excellence in portfolio management.

Methodology:

1. Data Preparation

We began by loading daily price data for the NASDAQ-100 and its component stocks for the years 2023 (in-sample) and 2024 (out-of-sample). The following steps were taken for data processing:

- **Daily Returns Calculation**: Calculated the daily returns for each stock and the NASDAQ-100 index.
- **Correlation Matrix**: Constructed a correlation matrix using 2023 returns data to quantify the similarity between stocks. This matrix serves as the basis for stock selection.

2. Stock Selection Using Integer Programming

The first step in constructing the index-tracking portfolio is selecting a subset of stocks that best represents the NASDAQ-100. Given the constraints of managing a smaller portfolio, our goal was to select a subset of m stocks from the total n stocks in the NASDAQ-100, where m is substantially smaller than n. This selection was based on maximizing the overall similarity between the selected subset and the index.

Formulation:

- Decision Variables:
  - ○ y_j: Binary variable indicating whether stock j is included in the fund (1 if selected, 0 otherwise).
  - ○ x_{ij}: Binary variable indicating whether stock j represents stock i in terms of similarity.
- Objective Function: Maximize the total similarity, calculated as the sum of correlations between the stocks in the fund and the stocks they represent.
  - ○ $Maximize \sum_{i=1}^{n}\sum_{j=1}^{n} \rho_{ij} \cdot x_{ij}, \ where\ \rho_{ij}\ represents\ the\ correlation\ between\ stock\ i\ and\ j$
- Constraints:
  - ○ Ensure exactly m stocks are selected for the fund: $\sum_{j=1}^{n} y_j = m$
  - ○ Each stock i in the index must be represented by exactly one selected stock j: $\sum_{j=1}^{n} x_{ij} = 1$ for all i
  - ○ Stock i can only be represented by stock j if stock j is included in the fund: $x_{ij} \le y_j\ for\ all\ i\ and\ j$

3. Weight Optimization Using Linear Programming

With the selected subset of stocks, the next step was to determine their weights in the portfolio to minimize tracking error relative to the NASDAQ-100 index. We formulated a linear programming model to achieve this goal.

Formulation:

- Decision Variables:
  - ○ w_i: Continuous variable representing the weight of stock i in the portfolio, constrained to be non-negative.
  - ○ z_t: Auxiliary variables representing the absolute deviation between portfolio returns and index returns at each time t.

- Objective Function: minimize the sum of absolute deviations between the portfolio's daily returns and the NASDAQ-100 index returns.
    - $Minimize\ \sum_{t=1}^{T} z_t$
- Constraints:
    - Ensure the sum of portfolio weights is equal to 1: $\sum_{i \in selected\ stocks} w_i = 1$
    - Ensure that z_t captures the absolute difference between portfolio and index returns at each time t: $z_t \geq q_t - \sum_{i \in selected\ stocks} w_i r_{it}$, $for\ all\ t$

4. Mixed-Integer Programming for Alternative Weight Selection

To explore an alternative approach, we reformulated the weight selection problem as a Mixed-Integer Program (MIP) that constrains the number of non-zero weights in the portfolio, providing a sparse weighting scheme. This model involved using binary variables to control the sparsity of the weights.

Formulation:

- Binary Decision Variables:
    - y_j: Binary variable indicating if stock i is assigned a non-zero weight (1 if selected, 0 otherwise).
- Continuous Decision Variables:
    - w_i: Continuous variable representing the weight of stock i, constrained to be zero if y_j = 0 using a "big M" constraint.
- Objective Function: Similar to the LP model, minimize the total absolute deviation between portfolio and index returns.
- Constraints:
    - Limit the number of non-zero weights to m: $\sum_{i=1}^{n} y_i = m$
    - Ensure w_i = 0 if y_j = 0: $w_i \leq M \cdot y_j$, for all i where M is a sufficiently large constant (1.0 in our case).

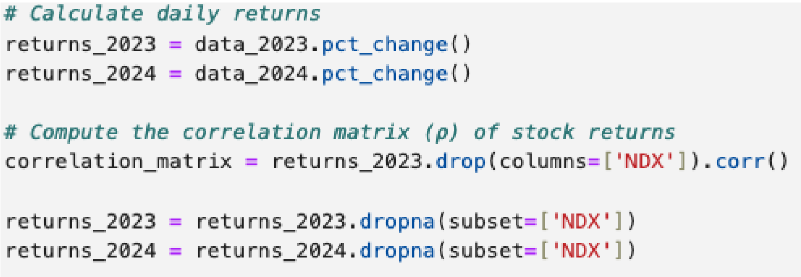
This MIP model allows for weight selection with a limited number of stocks having non-zero weights, resulting in a sparser portfolio. Due to the computational intensity of this approach, we set a time limit for the Gurobi solver to ensure efficient execution.

5. Performance Evaluation

To assess the effectiveness of the portfolio, we calculated tracking error and standard deviation for both in-sample (2023) and out-of-sample (2024) periods. The tracking error is measured as the mean absolute deviation between portfolio and index returns. The standard deviation compared the volatility of the portfolio against the NASDAQ-100 index. And these metrics allowed us to evaluate the tracking accuracy and stability of the constructed portfolio under different values of m.

Data and Process:

When we did some exploratory data analysis, we discovered that our csv files were not flawless. There were missing values for the index, and stocks within the index. This makes sense, because stocks can enter and leave the index, based on trading volume, market cap, etc.
We first decided it would be best to compare the percentage change between the prices between stocks. This lets us easily assess similarity by computing a correlation matrix.



```
# Calculate daily returns
returns_2023 = data_2023.pct_change()
returns_2024 = data_2024.pct_change()

# Compute the correlation matrix (ρ) of stock returns
correlation_matrix = returns_2023.drop(columns=['NDX']).corr()

returns_2023 = returns_2023.dropna(subset=['NDX'])
returns_2024 = returns_2024.dropna(subset=['NDX'])
```

We decided not to drop any missing values. pct_change() returns ‘NaN’ if there is a missing value, and corr() handles NaN by doing a “pairwise complete observations” approach. This means that if there is a NaN value in any column, it will only use the rows where each pair are non-missing values. This allows us to get a more accurate correlation matrix than if we dropped all NaN rows, or filled them with another value.

Since quicksum ignores NaN values, we left them in, except for where they were in the NDX return. We need all of the NDX values in order to minimize the tracking error since the goal is to get close to the NDX or y of the function. We removed these rows after we calculated the correlation matrix. The reason for removing them was to not attempt to skew the data or use 0 which might bias the optimization model accurately to future data.

Results:

We first created a tracking portfolio using integer programming with m=5 stocks, chosen and weighted using data from 2023, and subsequently assessed its performance in tracking the NASDAQ-100 index throughout 2024.

Selected Stocks and Portfolio Weights

Using integer programming (IP), we selected five stocks from the NASDAQ-100—*HON, INTU, NXPI, PEP, and SNPS* - to form a representative subset that **maximized similarity with the index**. Each selected stock was assigned an optimized weight to minimize daily tracking deviations:

- HON: 16.20%
- INTU: 22.01%
- NXPI: 17.48%
- PEP: 19.86%
- SNPS: 24.45%

These weights reflect the stocks' estimated contribution to closely mirror the index, derived from 2023 data.

Performance Analysis

The portfolio's tracking accuracy was evaluated for both in-sample (2023) and out-of-sample (2024) periods, with the following key performance metrics:

1. Tracking Error (Mean Absolute Error):

2023	0.004636
2024	0.0005914

   - These low tracking errors indicate that the portfolio closely followed the index, with only slight increases in deviation during 2024. The out-of-sample increase is expected due to natural shifts in market conditions, suggesting that the portfolio is effective yet sensitive to some market changes.

2. Portfolio Standard Deviation:

2023	0.011115
2024	0.011627

   - The portfolio’s standard deviation aligns closely with that of the index (2023: 0.011416; 2024: 0.011566), suggesting that its risk profile was well matched to the NASDAQ-100.

Observations and Justification

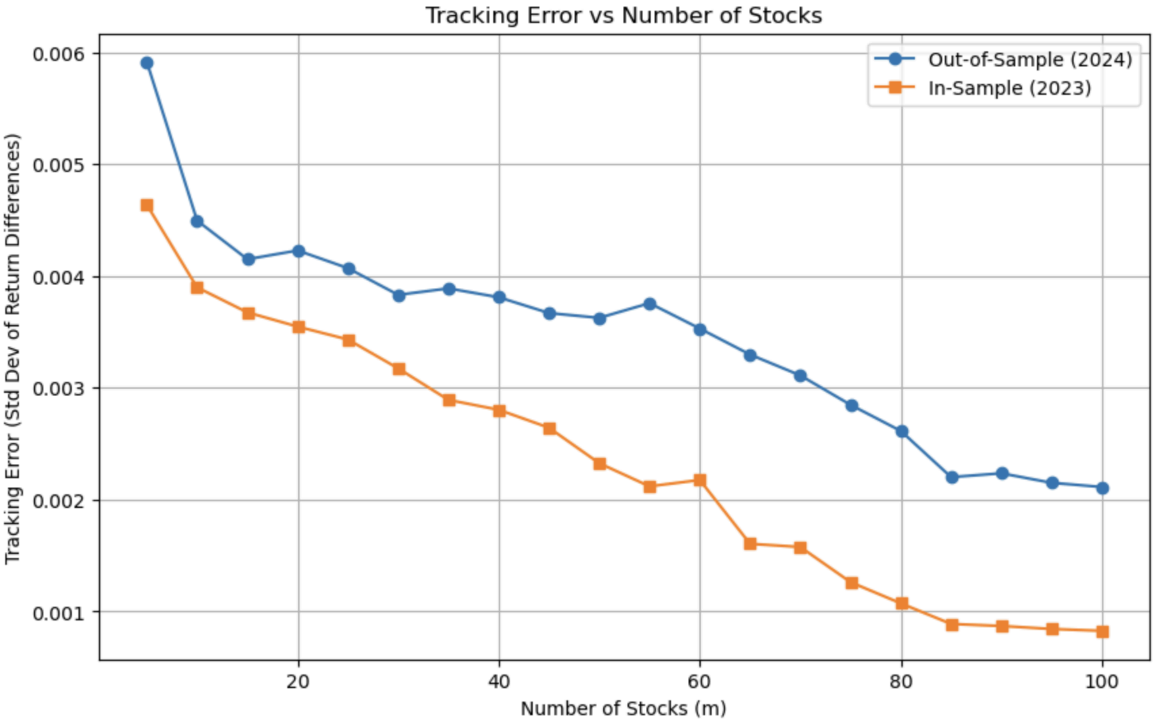
This close alignment in both tracking error and standard deviation for 2023 and 2024 validates our IP-based selection and weight optimization. The stocks chosen represent a well-diversified subset of the NASDAQ-100 that, together with the calculated weights, effectively tracks the broader index. The slight increase in tracking error for 2024 reflects natural variation due to out-of-sample testing but remains within acceptable bounds, reinforcing that this optimized selection is well-suited to track index performance with a minimal stock count.

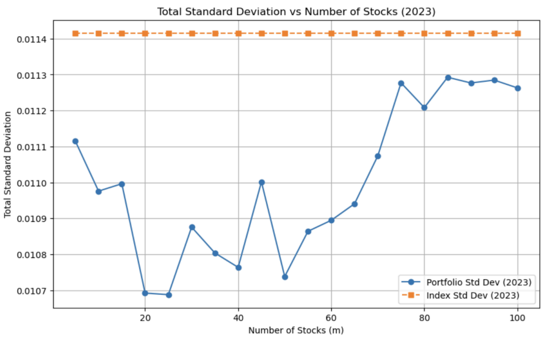
Following the same methodology, we experimented with different values of *m* to track the NASDAQ index. The results shown in the graphs offer additional insights into how portfolios with varying numbers of stocks (*m*) perform in tracking the NASDAQ-100 index.

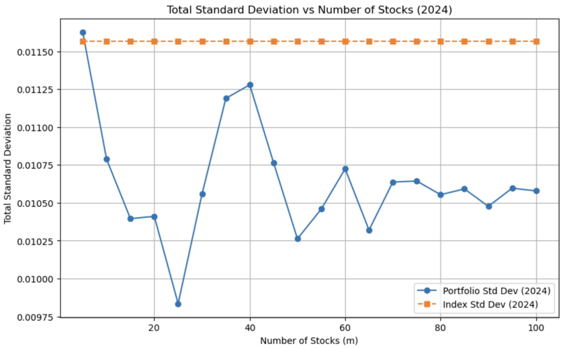
1. **Tracking Error**: The tracking error graph reveals that as m increases, the tracking error steadily decreases for both in-sample (2023) and out-of-sample (2024) scenarios. This reduction indicates that a larger number of stocks allows the portfolio to better replicate the index's returns. Notably, in the in-sample data, the tracking error is minimized at m=100, suggesting close alignment with the NASDAQ-100. However, tracking error in the out-of-sample scenario is consistently higher than in-sample across all values of m. This discrepancy may point to potential overfitting to the 2023 data or increased volatility in the 2024 data, underscoring the importance of examining both in-sample and out-of-sample tracking error to gauge robustness. *(See graph below*)
2. **Portfolio Volatility**: The standard deviation of portfolio returns (a measure of volatility) also follows a similar pattern, showing high volatility for portfolios with smaller m values, which then stabilizes as m grows. This initial high volatility is likely due to limited diversification, as portfolios with fewer stocks are more susceptible to individual stock risks. As m increases, the portfolio’s volatility aligns more closely with the NASDAQ-100 index, indicating reduced idiosyncratic risk and a risk profile more in line with the benchmark.

3. **Assessment**: The consistent decline in tracking error and convergence of portfolio volatility towards the index's level with increasing supports the conclusion that larger portfolios (higher m) offer better index-tracking performance. However, the ongoing gap in tracking error between in-sample and out-of-sample scenarios highlights the potential need for a balanced approach—while larger portfolios improve accuracy, they may also encounter diminishing returns in terms of added tracking precision.

In summary, increasing m reduces tracking error and volatility, enhancing the portfolio's alignment with the NASDAQ-100 index. However, the diminishing returns and out-of-sample variability suggest that an optimal m should balance tracking accuracy with cost-efficiency and robustness, ensuring the portfolio can perform reliably across different time periods.







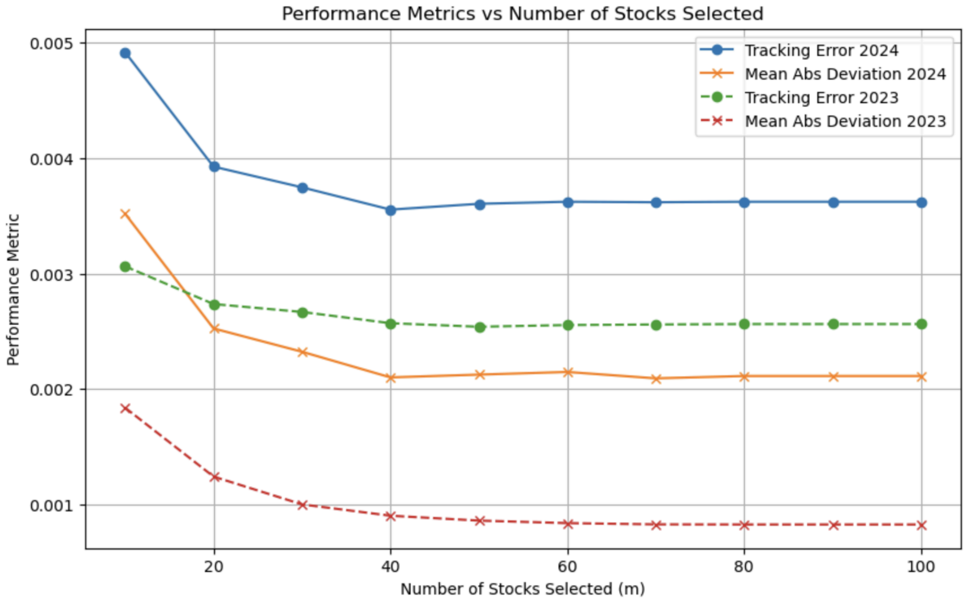
Mixed-Integer Program (MIP)

Continuing further, we explored an alternative approach by using a **Mixed-Integer Program (MIP)** to directly optimize weights across all NASDAQ-100 stocks without an explicit stock selection step. Instead of choosing specific stocks through integer programming, this method introduced binary variables to limit the number of non-zero weights to exactly m, allowing us to evaluate different portfolio sizes and their tracking performance.

Results and Observations

The results showed that the **tracking error stabilized around m=40**. This stabilization suggests that including more than 40 stocks provides minimal additional benefit in tracking accuracy, indicating that 40 stocks are sufficient to capture the main movements of the NASDAQ-100 index while minimizing unnecessary complexity.

1. **Tracking Error Behavior**: As m increased from smaller values, the tracking error initially decreased, showing that a larger number of stocks allows the portfolio to better approximate the index. However, beyond m=40, the improvement in tracking error became negligible, highlighting a point of diminishing returns. This stabilization at m=40 suggests an efficient balance where the portfolio is complex enough to track the index accurately without incurring excessive costs.



2. **Cost and Complexity Considerations**: Larger portfolios (higher m) naturally reduce idiosyncratic risk, but they also involve increased transaction costs and greater management complexity. The fact that tracking error no longer significantly improves beyond 40 stocks implies that adding more stocks would increase these costs without corresponding improvements in performance. Thus, m=40 represents an optimal portfolio size that achieves close tracking while controlling costs.
3. **Practical Implications**: The MIP-based approach allowed for a flexible and efficient portfolio design, where only 40 stocks need to be actively managed, reducing operational burdens. Additionally, this method demonstrated strong tracking performance comparable to larger portfolios, making it a viable option for cost-sensitive fund management.

**Note*

The stocks selected with MIP varied differently at m=5 then with the similarity matrix. This is in line with the findings that the of the 100 stocks a smaller amount (m=40) are the main drivers of the index.

AMZN	*15.51%*
APPL	*27.18%*
MSFT	*24.75%*
NXPI	*18.66%*

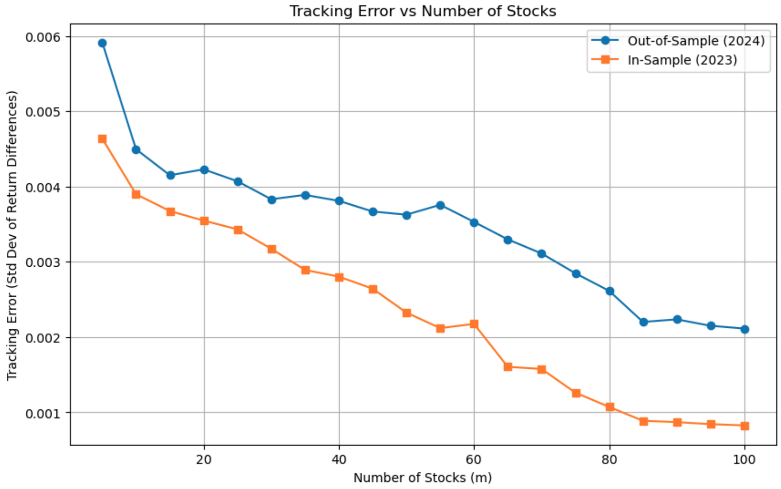
PEP	*13.89%*

Recommendations:

We recommend using the mixed integer programming approach to identify the optimal stocks and weights for the final portfolio. This version of the code directly minimizes the total absolute deviation between the portfolio and the index, which is ideal when we want to achieve a portfolio that closely matches and tracks the original index. In addition, the MIP approach optimizes the stocks themselves as well as the weights simultaneously, which helps avoid the mismatch which could occur in the original approach where stocks and weights are selected separately in two different optimization algorithms. With that said, the MIP program is more computationally demanding than the two-stage integer programming approach, so this should be considered if we attempt to scale this approach to other indexes.

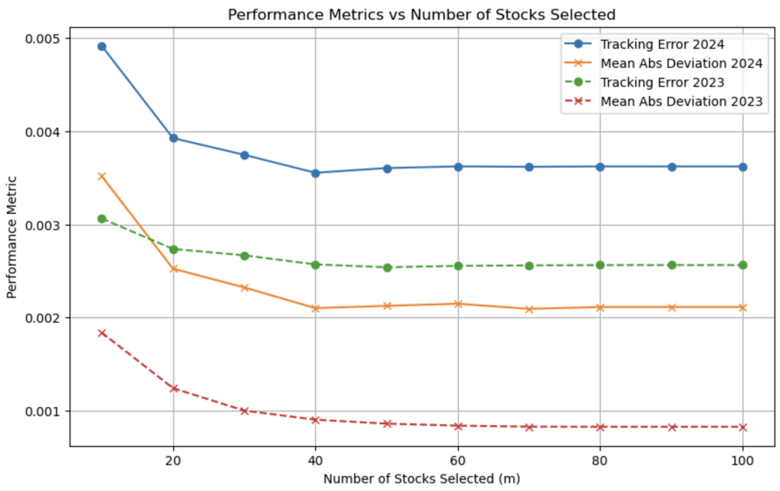
Below are graphs showing the performance metrics for the integer and MIP methods.

Tracking Error for Integer Programming Approach



As we can see in the MIP programming metrics, the tracking error of the fund seems to stabilize at roughly m = 40, suggesting that this is the ideal number of stocks which we should have in our portfolio. At m = 40, we have an optimized trade off between getting as close as possible to the real index (i.e., minimizing the tracking error) and minimizing the costs of creating such a large fund - transaction costs, computational complexity, challenges managing such a large portfolio, etc.

Performance Metrics for MIP Programming Approach



Our code can be used across the company for customized stock portfolios of any size. This will help reduce our time spent on optimizing portfolios, and allow us to spend more time obtaining new clients, tracking fund performance, and doing other value-add activities. The code can also be modified to include additional constraints such as incorporating sector weights and ESG factors. This will help us attract a diverse array of clients and customize to their needs, thereby improving customer satisfaction and improving our company's bottom line.

This tool has applicability to other indexes as well, such as international stocks, or groups of stocks from particular industries. For example, we could create funds that track indices while tilting towards certain factors such as value and momentum without increasing (or significantly increasing) the tracking error. In addition, we can dynamically rebalance our portfolio when there are changes in the stock market, by simply rerunning this code on a recurring basis and feeding it new training data. Lastly, there is also an opportunity to license this as a software product and offer it to other asset managers - this would serve as a low cost, high profit source of value to our company.

In conclusion, this code offers many benefits and opportunities for the company, including more efficient allocation or resources and saved time, cost reduction from having fewer stocks to manage in each portfolio, and enhanced decision making. It's also clearly reportable and well documented, which makes it easy to work with regulatory agencies. Implementing the MIP approach allows us to offer superior service to meet our clients' evolving needs and positions our company at the forefront of quantitative portfolio management.