MATH 441: Linear Programming Project

Group 15 - Fatima, Husain, Stallon, Shadi

February 2025

1 Introduction and Context

In financial markets, investors aim to maximize their returns while managing risk within a given budget. The challenge of asset allocation is particularly relevant for short-term traders who need to optimize their portfolio daily. We seek to determine an optimal allocation strategy given a fixed daily budget and a selection of assets (stocks, bonds, and cryptocurrencies) over a five-year historical period. Using linear programming techniques taught in class, we aim to construct a portfolio that maximizes daily returns while adhering to various constraints, such as investment caps per asset and risk tolerances. We used optimization methods, statistical analysis, and computation to evaluate the performance of assets and build an optimal portfolio.

Problem Statement:

Given a fixed daily budget B and a set of financial assets (stocks, bonds, and cryptocurrencies) with their respective daily opening and closing prices over a 5-year period, our objective is to optimally allocate the budget across these assets at the start of each trading day to maximize returns by the end of the day. The model must ensure that the allocation adheres to constraints on risk, individual asset exposure, and regulatory investment limits.

2 Variables & Parameters

Description	**Symbol**	**Dimensions**
Amount invested in stock i	s_i	CAD$
Amount invested in bond i	b_i	CAD$
Amount invested in crypto i	c_i	CAD$
Opening price of stock i	o_{s_i}	CAD$
Opening price of bond i	o_{b_i}	CAD$
Opening price of crypto i	o_{c_i}	CAD$
Mean Absolute Deviation for stock i	α_{s_i}	CAD$
Mean Absolute Deviation for bond i	α_{b_i}	CAD$
Mean Absolute Deviation for crypto i	α_{c_i}	CAD$
Risk parameter	γ	1
Asset number	i	1
Budget	B	CAD$

Asset Data

- **Stocks:** Microsoft (MSFT), Apple (AAPL), Amazon (AMZN), Alphabet (GOOGL), Alibaba (BABA).
- **Bonds:** Vanguard Total Bond Market ETF (BND), iShares Core U.S. Aggregate Bond ETF (AGG), Vanguard Total International Bond ETF (BNDX), iShares 20+ Year Treasury Bond ETF (TLT), Vanguard Intermediate-Term Corporate Bond ETF (VCIT).
- **Cryptocurrencies:** Bitcoin (BTC), Ethereum (ETH), Ripple (XRP), Solana (SOL).

3 Assumptions & Constraints

- The total money invested cannot exceed the budget

$$\sum_{i=1}^{5} s_i + b_i + c_i \leq B, \quad i \in \{1,2,3,4,5\}$$

- No more than 20% in one asset to ensure diversification in the portfolio and mitigation of risk

$$\frac{s_i}{B} \leq 0.2, \quad \frac{b_i}{B} \leq 0.2, \quad \frac{c_i}{B} \leq 0.2, \quad i \in \{1,2,3,4,5\}$$

- Investments cannot be negative

$$s_i, b_i, c_i \geq 0, \quad i \in \{1,2,3,4,5\}$$

- The sum of risk terms cannot exceed the gamma value

$$\sum_{i=1}^{5} \frac{s_i \alpha_{s_i}}{B o_{s_i}} + \frac{b_i \alpha_{b_i}}{B o_{b_i}} + \frac{c_i \alpha_{c_i}}{B o_{c_i}} \leq \gamma$$

- Only the top 5 stocks, bonds, and cryptocurrencies will be taken into consideration
- For this paper, we assume the daily budget used at the start of the day to be $150,000
- We focus our results on the daily returns over a period of 5 days using actual data associated with each asset
- We only consider the daily opening and closing price of a stock for the past 5 years to plot our distribution
- The amount of profit we made from the assets is the end return we receive at the end of the day
- The data used is from the past five years
- Mean Absolute Deviation of daily return is used for the evaluate risk
- We used a fixed value of $\gamma = 0.4$ to compute and report our results
- The model can predict closing prices only for the range of opening prices observed in the data

4 Building Solutions

Decision Variables:

- s_i : Amount invested in stock i
- b_i : Amount invested in bond i
- c_i : Amount invested in crypto currency i

Probabilistic and data-driven model using Kernel Density Estimation:

The goal is to optimize portfolio allocation using a data-driven, probabilistic approach while maintaining risk management constraints. Given the inherent unpredictability of financial markets, relying on long-term patterns and data distributions can provide deeper insights than simply averaging historical data. Based on this, the model uses Gaussian Kernel Density Estimation (KDE) to approximate the joint distribution of opening and closing prices based on historical price data, rather than assuming a fixed rate of return.

The model constructs a joint KDE for the opening and closing price of each asset, thus capturing the full range of price behaviors and interconnected dynamics between the opening and closing prices for the asset. The KDE is then used to calculate the distribution across a grid of values, covering the minimum and maximum opening prices observed for each asset. Subsequently, the marginal probability density function of the closing price is calculated for each opening price, enabling the calculation of the expected closing price given any specific opening price for an asset.

Objective Function:

$$\max \sum_i \frac{s_i}{o_i} E[\bar{s}_i] + \frac{b_i}{o_i} E[\bar{b}_i] + \frac{c_i}{o_i} E[\bar{c}_i]$$

The objective of the optimization is to maximize our returns by choosing an optimal way to invest our budget across the specified assets. The portfolio allocation for each asset is denoted by s_i, b_i and c_i, corresponding to the amounts invested in stocks, bonds, and cryptocurrencies, respectively. Consequently, the opening price for each asset is represented as o_{s_i}, o_{b_i}, and o_{c_i}, and $E[\bar{s}_i], E[\bar{b}_i]$, and $E[\bar{c}_i]$ denote the expected closing price for stocks, bonds, and cryptocurrencies, respectively. The values s_i, b_i, and c_i represent the amount allocated to each asset class, and the only distinction among these variables is the asset category they represent (stocks, bonds, and cryptocurrencies). Each term in our objective function represents the expected return from an asset, weighted by the number of units of that asset bought. By dividing the investment amount by the asset's opening price, we calculate the number of units of the asset purchased. Multiplying this by the expected closing price provides the anticipated total return from the asset at the end of the day.

Computing the Risk:

The risk associated with each asset is calculated using the Mean Absolute Deviation (MAD) of the daily return, which is computed using the opening and closing price of the asset over the past five years.

$$\text{Daily return} = \text{Closing price - Opening price (for a particular day)}$$
$$\text{Mean Absolute Deviation } (\alpha) = \tfrac{1}{n}\sum_j |\ r_j \text{ - M(r)}|$$

where n = Number of daily returns considered, r_j = Return on day j, and $M(r)$ = Mean of the n daily returns. However, the Mean Absolute Deviation is a subjective measurement which depends on opening and closing prices for each asset. In order to generalize the risk, we divide the MAD for each stock by its opening price. Additionally, we introduce a weighting factor based on proportion of the total portfolio budget allocated to each asset. The measure of risk for each stock, bond, and asset can then be defined as

$$\frac{s_i \alpha_{si}}{B o_{s_i}}, \frac{b_i \alpha_{bi}}{B o_{b_i}}, \text{ and } \frac{c_i \alpha_{ci}}{B o_{c_i}}$$

where s_i, b_i, and c_i represent the investments in stocks, bonds, and cryptocurrencies, respectively, $\alpha_{s_i}, \alpha_{b_i}$, and α_{c_i} are the Mean Absolute Deviations, B is the total portfolio budget, and o_{s_i}, o_{b_i}, and o_{c_i} are the opening prices for each asset class. The sum of these terms are constrained by the parameter γ, with the higher values of γ allowing for a higher permissible risk. The MAD for each asset is pre-computed before solving the linear program.

Estimating Expected Returns and Risks:

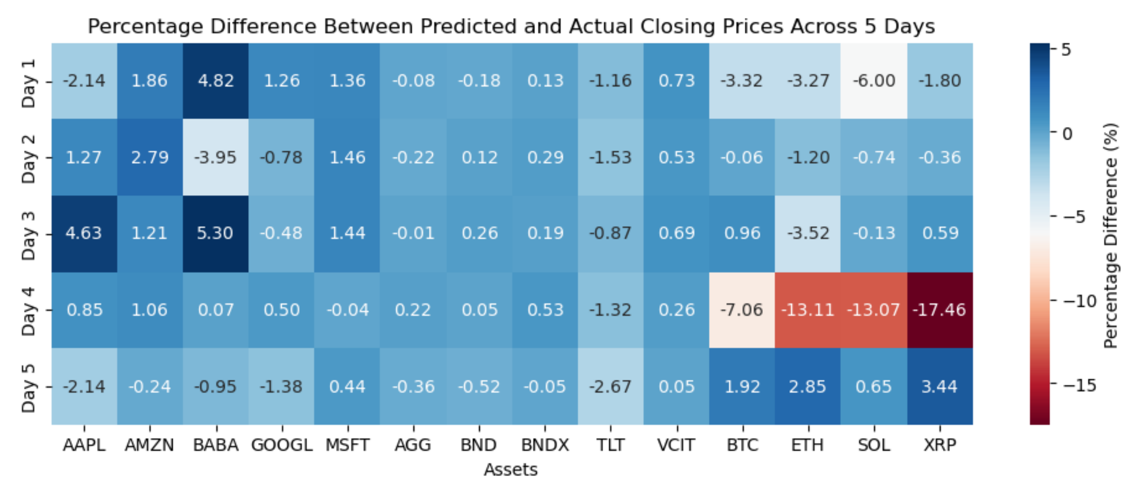
Since our model is probabilistic, there is always a chance that our predictions may not be entirely accurate. To mitigate this risk, we enforce the constraint that no more than 20% of the budget can be allocated to any single asset. This ensures that even if the model predictions are incorrect, we do not concentrate all our funds in one asset, thereby reducing risk. By spreading investments across multiple assets, we protect the portfolio from major losses due to unexpected price fluctuations in any one investment, giving a more well-balanced investment strategy.

Implementing the model and solving the Linear Program:

The model to calculate the expected closing price is implemented in Python using the scipy, numpy, and pandas library. The Mean Absolute Deviation for each asset is calculated using numpy and pandas. The Linear Program (LP), formulated based on the objective functions and constraints outlined above, is set up using numpy and pandas, and is solved using scipy.

5 Analyze and Assess

Performance of the model:



The heatmap above visualizes percentage differences between predicted and actual closing prices, highlighting the model's predictive accuracy relative to the closing price. Most stocks and bonds show relatively small deviations, with Amazon, Microsoft, and VCIT maintaining errors below ±2%

on most days. However, Alibaba exhibits a 5.30% deviation on Day 3, suggesting the model struggled with its price movement, while Google and Microsoft had milder discrepancies of -3.95% on Day 2 and $\pm$1.5% overall, indicating moderate accuracy for tech stocks. The largest errors occur in cryptocurrencies, with Ethereum (ETH) and Solana (SOL) exceeding -13% on Day 4 and XRP recording -17.46% on Day 5, suggesting the model underestimated their closing prices. These discrepancies likely stem from the high volatility of crypto assets, which the Kernel Density Estimation (KDE)-based model may not fully capture, as crypto markets exhibit rapid, unpredictable fluctuations compared to traditional stocks and bonds.

Portfolio Allocation:

Asset	Day 1	Day 2	Day 3	Day 4	Day 5
Apple	X	30000.0	X	X	24684.93
Amazon	30000.0	30000.0	30000.0	30000.0	30000.0
Alibaba	30000.0	30000.0	30000.0	30000.0	30000.0
Google	30000.0	28564.71	30000.0	30000.0	X
Microsoft	27042.94	30000.0	23092.74	X	30000.0
AGG	X	X	X	X	X
BND	X	X	X	X	X
BNDX	2957.06	1435.29	6907.26	30000.0	X
TLT	X	X	X	X	X
VCIT	30000.0	30000.0	30000.0	X	30000.0
BTC	X	X	X	30000.0	X
ETH	X	X	X	X	5315.07
SOL	X	X	X	X	X
XRP	X	X	X	X	X

The investment table provides a structured view of asset allocation over five days as calculated using our formulated LP. Amazon, Alibaba, and VCIT consistently received 30,000 CAD, suggesting they were considered stable and profitable under the given constraints. In contrast, Google and Microsoft saw fluctuations, with Google receiving 28,564.71 CAD on Day 2 and Microsoft dropping to 23,092.74 CAD on Day 3, indicating the model's dynamic response to changing expected returns. BNDX initially received minimal allocation (1,435.29 CAD on Day 2) before increasing to 30,000 CAD on Day 4, suggesting an improved return projection. Meanwhile, Apple, bonds like AGG, and cryptocurrencies (BTC, ETH, XRP, SOL) saw little to no investment, likely due to their higher risk or lower expected returns. The model's near-complete avoidance of cryptocurrencies suggests that the imposed risk constraints limited allocations to these volatile assets, reflecting a strong preference for more predictable investments.

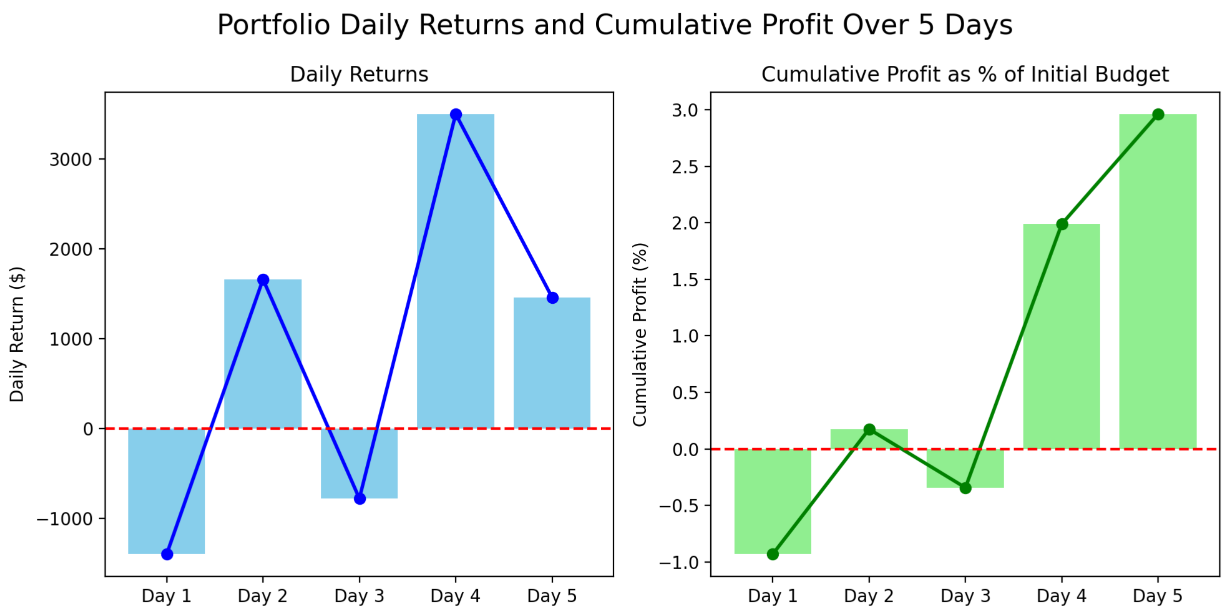
The model exhibited a strong preference for allocating capital to a limited subset of assets, consistently selecting no more than 5-6 investments per day. Despite a broader range of options, the solver repeatedly concentrated funds at the boundary of the imposed 20% allocation constraint, suggesting that it systematically prioritized maximizing expected returns over diversification. This rigid allocation pattern raises concerns about whether the model is overly constrained or failing to account for the benefits of spreading risk across a wider range of assets. The lack of variation across different risk levels and KDE bandwidths further suggests that the optimization framework may not be sufficiently sensitive to parameter adjustments, warranting further investigation into the effectiveness of the constraints and the potential need for additional diversification mechanisms.

6 Reporting Results

After allocating our budget among the specified assets using the expected closing price produced by the model, we calculate the real-life performance of our portfolio using the actual closing price of the asset on the corresponding day to calculate the total return from our portfolio. The optimal investments gave us the following results:

Day	End Value of Portfolio (CAD)	Return (CAD)
Day 1	148,604	-1,396
Day 2	151,659	1,659
Day 3	149,223	-777
Day 4	153,497	3,497
Day 5	151,459	1,459

The total return over the course of 5 days is **4442 CAD** which around 3% of the budget. Hence, we yielded a small but positive return over the initial investment. Across multiple test days, the model demonstrated consistent results, with daily returns ranging from **148,604 CAD to 153,497 CAD**.



The results show a few important takeaways. Firstly, diversification played an important role in optimizing risk-adjusted returns by spreading investments across stocks, bonds, and cryptocurrencies. Due to the risk constraints high-yield crypto investments were limited. We could not allocate a large amount of money to assets like Bitcoin (BTC) and Solana (SOL), which had higher return potential but also came with more volatility. On top of that, the actual returns cited above were only measured by buying and selling stocks, cryptocurrencies, and bonds on a day-to-day basis: we did not hold any of them at all during the 5 day stretch, which in turn led to the graphs that you see above. Bonds added stability to the portfolio, and investments in VCIT and other bond ETFs helped offset risk while still contributing to overall returns. Lastly, linear programming proved to be an effective tool for balancing return maximization and risk minimization, ensuring that the portfolio remained within the given constraints while still aiming for the best possible outcome.

Furthermore, the model is limited in its ability to evaluate and assess the performance of the model over a certain period of time after buying it, since the model only accounts for buying and selling within the same day. The ability to hold them not only gives more depth to our model, but also allows the model to potentially produce increased actual returns, and expanding the possibilities of effectively achieving high returns by giving us the liberty of optimizing the portfolio allocation in more ways than one.

To further enhance the model, we can incorporate the beta value of each stock, which measures its volatility relative to the overall market. Currently, the model accounts for risk using Mean Absolute Deviation divided by the opening price, which provides a measure of absolute price variability. However, beta would allow us to adjust for systematic risk, indicating how much an asset moves relative to the market itself. By integrating beta into our risk constraints, we can ensure that the portfolio is not overly exposed to high-volatility assets that could significantly impact returns during market downturns, and this allows for a more balanced and stable portfolio that still optimizes for maximum returns.

Our project shows how linear programming can be applied to real-world financial decisions by optimizing portfolio allocation under specific constraints. By analyzing historical price data and using statistical estimation methods, we were able to build a model that efficiently distributes investments across different asset classes.

References

[1] List of public corporations by market capitalization. (2020). Wikipedia.org. Retrieved from https://en.wikipedia.org/wiki/List_of_public_corporations_by_market_capitalization

[2] Cryptocurrency Historical Prices. (2025). Nasdaq.com. Retrieved from https://www.nasdaq.com/market-activity/cryptocurrency/

[3] Stocks Historical Prices. (2025). Nasdaq.com. Retrieved from https://www.nasdaq.com/market-activity/bonds/

[4] Bonds Historical Prices. (2025). Nasdaq.com. Retrieved from https://www.nasdaq.com/market-activity/stocks/