Math CS 120FO Final Project Report: Finance Model

Daniel Naylor

Contents

1 Introduction

The goal of this project is to attempt to create an investment strategy based on volatility constraints that can maximize annual return. My intention is to attempt to create a model entirely on my own with my current knowledge, and then compare it to other models and advice made by experts.

2 Terminology

This report assumes familiarity with linear programming, but not finance. Hence, here are some useful finance-related terms that I will use extensively.

2.1 Dividend

Money paid regularly, usually on a quarterly basis, to shareholders of a company (i.e. stock-owners). Some stocks entitle you to dividends, which would be included in your profit.

2.2 Return

The percent change of the value (price + dividends) of an asset in a time period. For example, annual return refers to the return computed between two consecutive years; monthly return refers to the return between two consecutive months

Mathematically, let P_i be the original price, P_f be the new price, and D be the sum of all dividends paid during the time period. Then return is given as:

$$r = \frac{D + P_f}{P_i} - 1$$

Raw return is simply $r + 1$, i.e. getting rid of the -1 term. Both are useful, although raw return is the one most used mathematically.

2.2.1 Annualized Return

The annual return computed from a set of monthly returns. If $r_1, r_2, \ldots, r_{12}$ represents the monthly returns of January, February, ..., December, then the Annualized Return is given by

$$\text{Ann. Monthly Return} = (1 + r_1) \cdot \ldots \cdot (1 + r_{12}) - 1 = \left(\prod_{i=1}^{12} 1 + r_i \right) - 1$$

Of course, one may consider a different set of months, for example May 2020, ..., May 2021.

2.3 Volatility

The fluctuation of the return of an asset over a given time period. In this problem, I will be using Annual Volatility (i.e. how much the return fluctuates on a year-by-year basis). Mathematically, this is the standard deviation of a set of raw returns.

2.4 Basket

A collection of securities (financial assets) that are related together. In this model, I define the constraints in terms of the baskets I formed. The primary motivation for this is baskets tend to perform roughly equally.

2.5 Ticker

A short-hand symbol used to identify specific securities. For example, Google's ticker is GOOG.

It is usually more convenient to identify securities by their tickers since they are unique, while names aren't.

3 Data

3.1 Raw Data

All data was gathered from Yahoo! Finance. Given a set of tickers, I collected the following raw data for each ticker:

1. Price: The price of a security at the beginning & ending of the day. This was collected on a monthly basis, with only the first of the month considered. In order to aggregate this into one number, I considered the average of these prices.
2. Dividends: These were denominated per unit of stock (e.g. 5 means $5 paid per unit of stock owned). I had complete information for this (i.e. all dividends paid, if any, since the security's listing on the market, with exact dates).

Everything was labeled with dates, and all data collected is between January 2018 and now (i.e. 5 years). Using this information, I computed the following data

3.2 Average Annualized Return

Originally, I computed annual returns by only considering data on a yearly basis; however, this gave me an incredibly inaccurate picture, so I modified the computation like so:

First, I sectioned off a year's worth of data (e.g. June 2020–June 2021) and computed the monthly return between each month. This gave me 12 separate raw monthly returns that I could annualize very straightforwardly. I ran this computation over the course of 5 years, giving me 5 separate annualized returns, from which I took the average.

As with all datasets, Yahoo! Finance had some imperfections, so some months were missing. I addressed this by skipping over the missing month and computing the monthly return between the two adjacent months. I.e. If I had the price for February 2020, April 2020, then March is clearly missing, so I treated February and April as if they were adjacent and computed their monthly return.

Imperfections are also addressed in the **Confidence** metric.

3.3 Volatility

Volatility was very straightforward to compute: it was simply the standard deviation of the computed annualized returns. Interestingly, volatility was always less than 1.

3.4 Confidence

A metric I created to measure how complete and accurate the above two metrics are. I mention the possibility of missing months in **Average Annualized Returns**, which is quantified here. Mathematically, this is given by

$$\text{Confidence} = \frac{\#\text{ of months present in data}}{\#\text{ of months measuring}}$$

It is easy to see that the maximum confidence is 1, while the minimum is 0 (although it doesn't make sense to have 0 confidence).

3.5 Basket

This wasn't something I directly computed, nor is it used in any of the previous formulas (though it is used later on). This is a hyperparameter specifying the baskets a security belongs to; since some securities are multifaceted (meaning they influence/are influenced by many different sectors of the economy), they belong to multiple baskets. A good example of a multifaceted security is the S&P 500 Index.

4 LP Model

4.1 Is LP Appropriate?

4.1.1 Additivity

This assumption holds pretty clearly; the total money gained is equal to the sum of the profit of each individual security. Similarly for constraints.

4.1.2 Proportionality

Proportionality also holds for similar reasons. The new balance of money from an investment is $(1 + r)P$, where P is the initial balance and r is the return. The constraints are also designed to be proportional.

4.1.3 Divisibility

This one is also pretty obvious, since most markets allow you to invest in fractions of securities, instead of requiring you to buy a full security.

4.1.4 Certainty

This assumption is most definitely at risk of failure, and is largely why volatility is baked into the constraints. Since the return data is annualized from monthly returns over the period of 5 years, this gives a fairly accurate picture in the period of a year. Sensitivity analysis is very important in this project, especially on the annual returns.

Ideally, I would want to compute annual returns from more granular data (e.g. annualized daily returns). However, working with monthly returns was very straightforward, and it still gives a fairly accurate picture.

4.2 Models

For each of my models, the results will come from the following data. Of course, my dataset changed continuously as I improved my model, so this is not to suggest that only my model experienced change; I use one dataset to showcase differing results and performances.

	Annual Return	Volatility	Confidence	Name	Basket
^GSPC	0.102223	0.169448	1.0	S&P 500	INDEX, FINANCE, TECH
^IXIC	0.138486	0.238355	1.0	NASDAQ Composite	INDEX, TECH
V	0.130797	0.147103	1.0	Visa Inc.	FINANCE
JPM	0.137138	0.36892	1.0	JP Morgan Chase & Co.	FINANCE
BAC	0.081274	0.415928	1.0	Bank of America Corporation	FINANCE
AAPL	0.357177	0.343292	1.0	Apple Inc.	TECH
AMZN	0.111889	0.279667	1.0	Amazon.com, Inc.	TECH
MSFT	0.301471	0.179769	1.0	Microsoft Corporation	TECH
GOOG	0.209567	0.335871	1.0	Alphabet Inc.	TECH
NVDA	0.656287	0.830054	1.0	NVIDIA Corporation	TECH
AMD	0.564303	0.4009	1.0	Advanced Micro Devices, Inc.	TECH
LMT	0.118941	0.06552	1.0	Lockheed Martin Corporation	DEFENSE
RTX	0.085984	0.226441	1.0	Raytheon Technologies Corporati	DEFENSE
BA	0.024819	0.507837	1.0	Boeing Company (The)	DEFENSE
GD	0.058105	0.200998	1.0	General Dynamics Corporation	DEFENSE
NOC	0.09796	0.146441	1.0	Northrop Grumman Corporation	DEFENSE
GBPUSD=X	-0.0089	0.094251	1.0	GBP/USD	CURRENCY
EURUSD=X	-0.015408	0.073329	1.0	EUR/USD	CURRENCY
JPYUSD=X	-0.044671	0.074298	1.0	JPY/USD	CURRENCY
GC=F	0.095141	0.120014	0.866667	Gold Aug 23	COMMODITIES
CL=F	0.109666	0.579386	0.866667	Crude Oil Jul 23	COMMODITIES

There are **21** securities in total among **7** baskets. I will also use the following variables:

- x_i = Amount to invest in the ith security (e.g. $x_1 =$ ˆGSPC, $x_2 =$ ˆIXIC, . . .)
- r_i = Return of ith security
- σ_i = Volatility of ith security
- c_i = Confidence of ith security
- N = # of Securities (21), M = # of Baskets (7)

4.2.1 First Model

First, I defined the objective function

$$\sum_{i=1}^{N}(1+r_i)\cdot x_i \tag{1}$$

Which gives the new balance after a year. Of course, $x_i \geq 0$ arises naturally from this. Then I defined the constraint

$$\sum_{i=1}^{N} x_i \leq 5000 \tag{2}$$

Which says that my total money invested cannot exceed \$5000. While seemingly arbitrary, this reflects the amount of cash liquidity I have on hand (i.e. this models my circumstances).

Finally, to define the remaining constraints, let $B_j \subseteq \{1,\ldots,M\}$ be a basket of securities, $|B_j|$ be the amount of securities in this basket, and $\beta = \sum_{j=1}^{M} |B_j|$ be the total size of all baskets (since some securities belong to multiple baskets, β is necessarily greater than N). Then we impose the constraint

$$\sum_{i\in B_j}(1+\sigma_i)x_i \leq \frac{5000\cdot|B_j|}{\beta} = b_j^u \tag{3}$$

I'll refer to the right-hand side of (3) as a **uniform partition**, since it evenly divides the amount to be invested among all the baskets. On the left hand-side, the coefficients for x_i in the constraint matrix is always either 0 or $1+\sigma_i$; $1+\sigma_i$ is used because I did not want the total investment of a basket to exceed the right-hand side of (3).

In total, there are $M+1$ constraints. Notice that the 'confidence' metric is not included in this model; this will be included in later models. Here is the report of this model from the data

- Total to Invest: 3848.1120156597462
  1. Visa Inc.: 726.4680527420202
  2. Advanced Micro Devices Inc.: 1189.7112379725904
  3. Lockheed Martin Corporation: 977.6132347243317
  4. EUR: 582.3004068854884
  5. Gold Aug 23: 372.0190833353209
- New Balance: 4757.190545347245

While seemingly interesting, this is very disappointing. For starters, the model does not invest near the full amount. EUR also has an average negative return, so the model should tell me to **not** invest in it. In fact, for each basket B_j, the model is simply selecting the security i that maximizes the following

$$(1+r_i)\cdot\frac{b_j^u}{1+\sigma_i}$$

And sets $x_i = \frac{b_j^u}{1+\sigma_i}$. For securities that belong to multiple baskets, it chooses the largest uniform partition b_j. This is very unrewarding because all of this could have been directly computed without applying linear programming.

(Note: I only verified the above statement for this data. I did not prove it in general.)

4.2.2 Second Model

To improve my first model, I overhauled each of the 3 above equations.

I changed (1) by introducing a new variable h, representing the **money to hold** (i.e. not invest). (1) became

$$\sum_{i=1}^{N}(1+r_i)x_i + h \tag{4}$$

Which intuitively says that not investing is equivalent to a 0% annual return. Similarly, (2) became

$$\sum_{i=1}^{N} x_i + h = 5000 \tag{5}$$

Note that this is equivalent to making the slack variable of (2) present in the objective function

Now to improve (3), define

$$\Sigma_j = \text{avg}\{\sigma_i\}_{i \in B_j}$$

In other words, Σ_j is the **average volatility** among all securities in a basket B_j. Now modify (3) to

$$\sum_{i \in B_j} \frac{(1+\sigma_i)x_i}{c_i} \leq \frac{5000}{1+\Sigma_j} = b_j^v \tag{6}$$

I will refer to the right-hand side of (6) as a **volatile partition**, since it depends on the volatility of each security in the basket.

The primary motivation for introducing c_i is to increase the cost of investing in securities whose data I have less confidence in.

The motivation for changing to the volatile partition (6) is largely to allow the model to invest more money in individual securities. With the uniform partition (3), adding new baskets imposes stricter bounds on all the other baskets. The point of this model isn't to force myself to invest in every basket; it's to identify worthwhile securities from a set of securities that may have influences on each other (hence the purpose of grouping by baskets). Investing evenly is not ideal since some baskets can be redundant (e.g. Index) or not worthwhile (e.g. Currency).

Here is the report of this model ran on the sample data.

- Total to Invest: 5000.0
    1. JP Morgan Chase & Co.: 2350.642721278203
    2. Advanced Micro Devices Inc.: 2649.357278721797
- New Balance: 6817.404747244822

The objective is clearly improved by a substantial amount, and moreover bad securities are identified and removed. However the investment portfolio is far less diversified. There's much debate in the finance community over the importance of diversification (namely, to what extent should one diversify their portfolio), and is largely subjective from investor to investor. I digress though, and leave this as it stands.

4.2.3 $1 + \sigma_i$ vs. σ_i

Note that using $1+\sigma_i$ as opposed to σ_i introduces bias into the model, and diminishes the effect of volatility (e.g. doubling σ_i will not double $1+\sigma_i$). Hence, I present the following modification to (6)

$$\sum_{i \in B_j} \frac{\sigma_i x_i}{c_i} \leq \frac{5000}{1+\Sigma_j}$$

This yields

- Total to Invest: 5000.0
    1. NVIDIA Corporation: 3977.55386362484

2. Advanced Micro Devices Inc.: 1022.4461363751599

- New Balance: 8187.388339175516

While the original bias is still present with $1 + \Sigma_j$, the alternative model where I divided by Σ_j simply tells me to put all my money into NVIDIA. My goal is to have a somewhat diverse portfolio (i.e. not all in one stock).

I don't have the expertise to comment on whether this modification or the original model is more accurate with real-world scenarios. Interestingly, this modification targets two securities in one basket, while the original version targets two separate securities. The 'Less Diversification' section addresses this.

4.3 Sensitivity Analysis

As mentioned in the beginning of this section, certainty is most clearly at risk of failure. Hence, sensitivity analysis is especially of interest.

Since each of the b_j coefficients are arbitrarily chosen by me (they effectively represent my desire to diversify), σ_i is the standard deviation of a sample, and c_i is computed by me, the coefficients that largely require sensitivity analysis are $1 + r_i$.

For the sake of brevity, I will not include a sensitivity analysis on any of the models here (since there are 21 different variables, i.e. 21 different coefficients to talk about). The analysis is very straightforward though. The original version of Model 2 is included in the GitHub repo in the appendix.

5 Further Research

Note that I am not, nor do I claim to be, an expert in modeling financial scenarios. Here I include comments on what experienced financial modelers look at.

5.1 Price & Return

In my model, I considered the average between closing and opening price. However, the convention is to use closing price, since this reflects the value of an asset after trading.

When considering closing price instead of average price, here is how Model 2's performance changed

- Total to Invest: 5000.0
  1. JP Morgan Chase & Co.: 2518.239662656423
  2. Advanced Micro Devices Inc.: 2481.760337343577
- New Balance: 6819.052954421253

The objective slightly increased, and the amounts changed by roughly $150.

Most financial models also look at the **natural logarithm** of raw returns, i.e. $\ln(1 + r_i)$. This is because return is exponential over the course of many periods. In the case of my model though, since I was restricting myself to optimizing return during a single cycle (annual), considering logarithmic return would be inappropriate.

5.2 Inflation

A point of interest is the coefficient of h in (4). Instead of leaving it as 1, modelers will account for the effects of inflation and consider it as a lossy investment. Measuring and including this is not as straightforward as changing the h coefficient though, so for simplicity's sake I did not include it in my model.

5.3 Back Testing

Another interesting point is something called "back testing". There are many metrics used to model financial markets, so a comprehensive and necessary list does not exist, but what many financial modelers do is create several models based on several variables and test it on relevant historical data. In my case, I made two (technically three) models; if I wanted to back test, I would apply each of these models on past data and see how it would have performed. Of course, I would refine the models with these results (as I did with my second model).

5.4 Less Diversification

While seemingly attractive, some investors argue against diversification (e.g. billionaire Warren Buffet calls it a "protection from ignorance"). Instead of looking at a few metrics among many securities, experienced investors will track a select few securities and weigh in as much information as possible (e.g. market capitalization, and external factors such as corporate interest rates for company stock).

5.5 Moving Average

Similar to my intention of only looking at the past years to model the present, moving average is a metric that also quantifies this. A simple definition uses a straightforward mean of prices within a time period; another definition gives weight to more recent prices through recursive definitions. A full discussion can be found here.

5.6 Volatility

Interestingly, most investors consider volatility as a "fear factor" that measures how other people will invest in securities. For long term investments, it is often not considered as a major factor. In other words, it isn't a constraining factor in the same way I used it for more sophisticated models.

5.7 Shorting

My model only accounts for a strategy that will maximize return. In particular, a security with negative annual return is always considered a bad investment in my model. This isn't true in general, as a profit can still be made from falling prices.

If one expects the price of a security to fall, the best course of action is to **short** the security. This consists of first selling the stock, let it decline, then buy the original shares back. While risky, if deployed correctly this can lead to very lucrative profits.

6 Appendix

All data was collected from Yahoo! Finace. I used the yfinance package in Python to scrape data. The linear optimization package was from scipy.optimize.

All code can be found here: `https://github.com/DanLeEpicMan/LP-Finance-Model`