# Capital Allocation & Portfolio Optimization

## Using Calculus, Mean-Variance Theory and Sharpe's Ratio

## Cluster Innovation Centre

### University of Delhi

February 2023

Month Long Project submitted for

***Single and Multivariable Calculus***

# Acknowledgment

I would like to express my special thanks of gratitude to my mentor **Dr Harendra Pal Singh** for his guidance and supervision as well as providing necessary information regarding the project. It would be a great pleasure for me to present my month long project on **Multivariable Calculus**

# Certificate of Originality

The work embodied in this report entitled “Capital Allocation & Portfolio Optimization” has been carried out by Saswat Susmoy for the paper “Single and Multivariable Calculus”. I declare that the work and language included in this project report are free from any kind of plagiarism.

**Dr Harendra Pal Singh**
Assistant Professor
Department of Mathematics
Cluster Innovation Centre
University of Delhi

# Certificate of Completion

This is to certify that the following person: **Saswat Susmoy Sahoo** has completed this Month Long Project for "Multivariable Calculus" under my guidance and supervision as per the contentment of the requirements of the first semester in the course B.Tech (Information Technology and Mathematical Innovations) at the Cluster Innovation Centre, University of Delhi.


**Dr Harendra Pal Singh**

Assistant Professor

Department of Mathematics

Cluster Innovation Centre

University of Delhi

# Index

# Abstract

# Capital Allocation & Portfolio Optimization

## Using Calculus, Mean-Variance Theory and Sharpe's Ratio

by

|| **Saswat Susmoy Sahoo** ||

Cluster Innovation Centre, 2023

In this paper, I'm interested in the contribution of Calculus for the optimization of portfolio funds to minimize risk while profiting optimally. Risks and Returns are two sides of the same coin and needs to be balanced to provide a risk-free portfolio. How to choose the asset, combination of assets, or security investments that present the best balance between risk and return on investment and maximize their expected utility is my major concern in this project.

# Introduction

## I.1 Background and Context

My motive was to build a mathematical model which would maximize the returns for a given number of investments. I have tried to use calculus to obtain the desired results.

## I.2 Scope and Objectives

My objective was to provide accurate results for our users to optimize their portfolios. I aimed to create a user-friendly interface for users which would allow them to get an idea about how to diversify their investments just by using historical data and figures.

## I.3 Achievements

I have successfully created a basic mathematical model and implemented it in python by using various modules and libraries. A user can achieve his desired output just by entering historical data which is widely available and generally used by seasoned investors.

## II.1 Problem Statement

Investing in various stocks and bonds is a trade for many individuals. Many people also have trading and investing as their passive income. The market, which is a volatile entity, always surprises investors. More often, people lose out on opportunities and money due to poor market study and a lack of free/valuable models to help with it. The existing sites that proclaim investing "gurus" more often turn out to be frauds and often base their predictions on instincts. Some valuable models in the market are available on a subscription-based model which may not be trusted by the users.

## Pre-Requisites

Theoretical

- Differentiation
- Determinants and Matrices
- Basic Knowledge of Modules & Libraries in Python

Hardware/Software

- IDE to run Python
- Python 3 or above

# ***II.2 Methodology***

# ***Return and Risk of an Asset***

## ***Return of an Asset***

Let $R_i$ represent the return generated by asset $A$ in state $i$ and $p_i$ represent the likelihood of state $i$ happening.

The expected return on the asset, $E(A)$ can be computed as below:

$$E(A) = \sum p_i R_i$$

## ***Expected Return of a Portfolio***

If we denote the return of the portfolios by $Rp$, then the return on the portfolio is given by:

$$Rp = \sum wjRj$$

where,

$wj$ = The weights of fund to be invested
$Rp$ = The return on a portfolio
$Rj$ = The return on an asset $j$

## ***Risk of an Asset***

The standard deviation of the asset can be computed below: $p_i$

$$\sqrt{\sigma_1^2 + \sigma_2^2....\sigma_n^2}$$

where, $\sigma$ is the standard deviation of a single stock
n is the nth stock in the portfolio

## ***Sharpe's Ratio***

The Sharpe ratio compares the return of an investment with its risk. The Sharpe ratio's numerator is the difference over time between realized or expected, returns and a benchmark such as the risk-free rate of return or the performance of a particular investment category. Its denominator is the standard deviation of returns over the same period of time, a measure of volatility and risk.

$$S_p = \frac{R_p - R_f}{\sigma_p}$$

where, $S_p$ is Sharpe's Ratio
$R_p$ is Returns of the portfolio
$R_f$ is Risk-Free Rate
$\sigma_p$ is Standard deviation

**Note:** Risk-Free Rate is a constant value calculated by finding the difference between the country's treasury bond yields and inflation rate.

## ***Optimizing Sharpe's Ratio***

The better Sharpe's Ratio the better is the return from the Portfolio. To maximize Sharpe's Ratio, we have to differentiate Sp with respect to $w$ which represents the weighted investment.

$$\frac{\partial}{\partial w_i} S_p = \frac{\partial}{\partial w_i}[\frac{R_p}{\sigma_p}] = 0$$

$$\Rightarrow \frac{\partial}{\partial w}[\frac{R_p}{\sigma_p}] = \frac{\sigma_p \frac{\partial R}{\partial w_1} - R_p \frac{\partial \sigma_p}{\partial w}}{\sigma_p^2} = 0$$

$$\Rightarrow \frac{[\sqrt{(w_1\sigma_1)^2 + (w_2\sigma_2)^2}\frac{\partial}{\partial w_1}[w_1 R_1 + w_2 R_2]] - [\frac{\partial}{\partial w_1}\sqrt{(w_1\sigma_1)^2 + (w_2\sigma_2)^2}[w_1 R_1 + w_2 R_2]]}{(w_1\sigma_1)^2 + (w_2\sigma_2)^2}$$

•

•

•

•

$$w_1 = \frac{R_1\sigma_2^2}{R_1\sigma_2^2 + R_2\sigma_1^2} \quad w_2 = \frac{R_2\sigma_2^2}{R_1\sigma_2^2 + R_2\sigma_1^2}$$

## ***Case-Study***

To better understand this optimization let's take a real stock from NASDAQ USA which is the second-largest stock exchange in the world.

For our Study, we will choose AAPL (Apple Inc.) & MSFT (Microsoft Corp). We will observe our model by using data from FY 2013-2018 and use FY 2018-2019 as our current year.

Returns

| | 13-14 | 14-15 | 15-16 | 16-17 | 17-18 | AVG |
|---|---|---|---|---|---|---|
| AAPL | 12.66 | 67.27 | -15.51 | 27.44 | 40.20 | 26.412 |
| MSFT | 42.01 | 9.67 | 40.05 | 20.56 | 50.23 | 32.50 |

For FY 2018-19:

- Expected Returns for AAPL = 26.412, MSFT = 32.504
- Standard Deviation for AAPL = 27.59, MSFT = 15.003
- Risk-Free Rate in the USA for the period = -2.945

According to our Model,
W1 should be 0.44 and W2 should be 0.55 so as to obtain an optimized Sharpe's Ratio of 46.449 and Returns of 29.504

Is it true? Let's check with the actual data of FY 2018-23

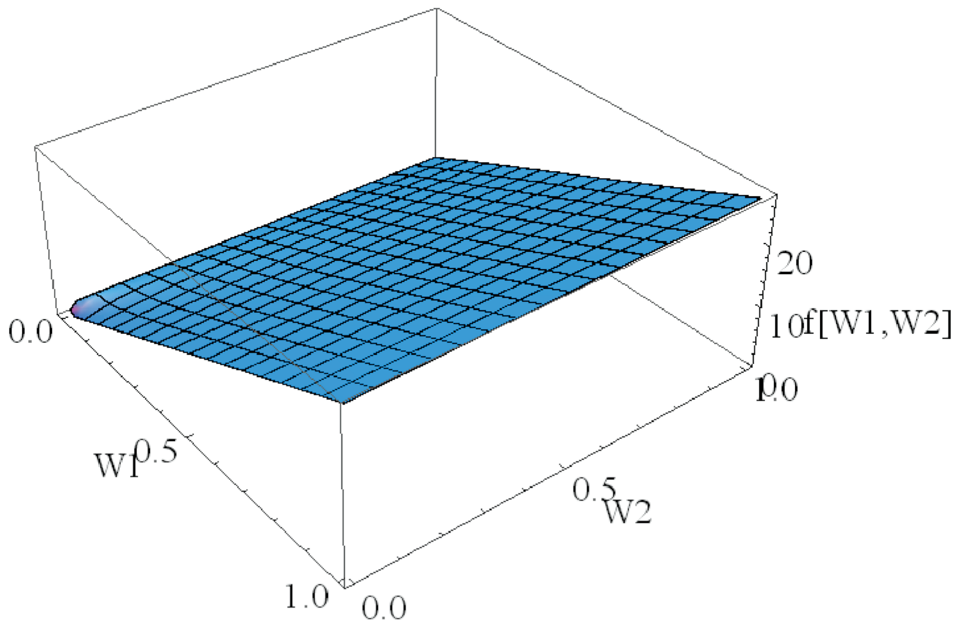
Our investment will be of $1000 in our portfolio with around 45% going to AAPL and 55% to MSFT.

For FY 2018-23

- Returns for AAPL = 262.99% and for MSFT = 177.03%

According to our model, an investment of 1000$ in this portfolio of AAPL and MSFT would yield a return of around 1179$ for AAPL and around 973.665$ for MSFT and the total value of the Portfolio would be 3152.94$ which is the highly optimized valuation minimizing the risk. Risk plays an important factor while investing in a stock.

| | 0/100 | 45/55 | 50/50 | 60/40 | 90/10 | 100/0 |
|---|---|---|---|---|---|---|
| Returns | 1770 | 2153 | 2185 | 2280 | 2544 | 2629 |




Graph supporting the Sharpe's Ratio

## ***Conclusion***

From the above study, we can conclude that Calculus and Markets go hand in hand when it comes down to Analysis and Predictions. We can somewhat predict a portfolio's success by using a Mathematical model based on Calculus.

But using this model solely while investing can prove very risky and one should adhere to market trends and choose portfolios and allocate funds accordingly.

Using previous trends, analysing graphs, watching out for a company's performance are some of the good practices while investing in stocks.

# *Appendix*

## (A) Python Implementation Source Code

```python
import numpy as np
from scipy.optimize import minimize

def portfolio_return(weights, returns):
    """
    Calculates portfolio return given a set of weights and asset returns
    """
    return np.dot(weights, returns)

def sharpe_ratio(weights, returns, risk_free_rate):
    """
    Calculates Sharpe ratio given a set of weights, asset returns and risk-free rate
    """
    return (portfolio_return(weights, returns) - risk_free_rate) / np.sqrt(np.dot(weights, weights))

def optimize_portfolio(returns, risk_free_rate):
    """
    Optimizes portfolio weights using Sharpe ratio as the objective function
    """
    num_assets = returns.shape[0]
    args = (returns, risk_free_rate)
    constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
    bound = (0.0, 1.0)
    bounds = tuple(bound for asset in range(num_assets))
    initial_guess = np.array([1/num_assets] * num_assets)
    result = minimize(fun=negative_sharpe_ratio, x0=initial_guess, args=args, method='SLSQP', bounds=bounds, constraints=constraints)
    return result

def negative_sharpe_ratio(weights, returns, risk_free_rate):
    """
    Returns the negative of the Sharpe ratio
    """
    return -sharpe_ratio(weights, returns, risk_free_rate)

if __name__ == '__main__':
    # Input asset expected returns and risk-free rate
    returns = np.array([float(x) for x in input("Enter expected returns of assets separated by space: ").split()])
    risk_free_rate = float("-2.945")

    # Optimize portfolio
    result = optimize_portfolio(returns, risk_free_rate)
    optimized_weights = result.x

    # Output results
    print("Optimized Weights:", optimized_weights)
    print("Portfolio Return:", portfolio_return(optimized_weights, returns))
    print("Portfolio Sharpe Ratio:", sharpe_ratio(optimized_weights, returns, risk_free_rate))
```

## (A) Graph Mathematica Code

```
f[AAPL_, MSFT_] := W1*26.412 + W2*32.504 /
Sqrt[(W1*27.59)^2 + (W2*15.002)^2]
Plot3D[f[W1, W2], {W1, 0, 1}, {W2, 0, 1}, PlotRange -> All,
 AxesLabel -> {"AAPL", "MSFT", "f[AAPL,MSFT]"}]
```

# ***Future Work***

We have developed some concepts that we intend to put into action in the future based on the progress we have made so far in this work:

- **Portfolio Manager**: I would like to create an Android/Web based application to apply a better and more accurate version of this model for public use.

- **Portfolio Optimization using Machine Learning**: I would like to explore the domains of Machine Learning and AI which would allow me to provide even better and more accurate analysis and data using more sophisticated algorithms and prediction models then conventional Calculus ones.

# References


- https://www.buyupside.com
- https://www.businessinsider.com
- https://www.investopedia.com/terms/s/sharperatio.asp
- https://www.macrotrends.net/countries/USA/united-states/inflation-rate-cpi#:~:text=U.S.%20inflation%20rate%20for%202018,a%200.31%25%20increase%20from%202017.
- https://analystprep.com/study-notes/actuarial-exams/soa/ifm-investment-and-financial-markets/mean-variance-portfolio-theory/