



A project report on

**Flask based Web Application for Portfolio Optimization
for US based equity instruments using Monte Carlo
Simulation**

Submitted in partial fulfillment of the requirements for the Degree of

B. Tech in Computer Science & Engineering

by

Ayan Mukherjee (1805027)

Gargi Mondal (1805389)

Sohita Kumari (1805346)

under the guidance of

Dr. Pradeep Kumar Mallick

School of Computer Engineering
Kalinga Institute of Industrial Technology
Deemed to be University
Bhubaneswar

March 2021



CERTIFICATE

This is to certify that the project report entitled “ **Flask based web application for portfolio optimization for US based equity instruments using Monte Carlo Simulations**” submitted by

Ayan Mukherjee **1805027**

Gargi Mondal **1805389**

Sohita Kumari **1805346**

in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in **Computer Science & Engineering** is a bonafide record of the work carried out under my(our) guidance and supervision at School of Computer Engineering, Kalinga Institute of Industrial Technology, Deemed to be University.

Signature of Supervisor 1

NAME OF THE SUPERVISOR 1

Academic affiliation

Organization

.....
The Project was evaluated by us on DD/MM/YYYY

EXAMINER 1

EXAMINER 2

EXAMINER 3

EXAMINER 4

ACKNOWLEDGEMENTS

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

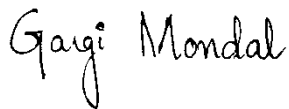
We would like to express our deep and sincere gratitude to our research supervisor, Dr. Pradeep Kumar Mallick, School of Computer Engineering, KIIT University for giving us the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation has deeply inspired us. He taught us the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. We are extremely grateful for what he has offered us.

We would like to say thanks to Dr. Bhabni Shankar Prasad Mishra, Dean School of Computer Science Engineering & Dr. Samareesh Mishra, Professor & Director School of Computer Science Engineering for his genuine support throughout this research work.

Finally, our thanks go to all the people who have supported us to complete the research work directly or indirectly.



Ayan Mukherjee



Gargi Mondal



Sohita Kumari

ABSTRACT

MPT or Modern Portfolio Theory also known as mean-variance analysis is a mathematical modeling technique that is deployed in constructing portfolios that can maximize the portfolio return for a given amount of risk. In this paper we optimize portfolios in accordance to the Modern Portfolio Theory for US based equity instruments using Monte Carlo Simulations.

For a given Portfolio 'P' having 'n' number of stocks, with each stock 'i' having weight of ' w_i ' we compute the mean and risk (standard deviation) and optimize our portfolio by optimizing the weights ' w_i ' for the equity instruments using Monte Carlo simulation.

TABLE OF CONTENTS

Abstract	:	1
Table of Contents	:	2
List of Figures	:	3
List of Tables	:	4
 CHAPTER 1: INTRODUCTION	:	 5
 CHAPTER 2: BACKGROUND/BASIC CONCEPTS		 6-7
2.1 Expected Return	:	6
2.2 Portfolio Return Variance	:	6
2.3 Portfolio Return Volatility	:	7
2.4 Sharpe Ratio	:	7
2.5 The Efficient Frontier	:	7
 CHAPTER 3: PROJECT ANALYSIS/ PROJECT IMPLEMENTATION		 8-14
3.1 Annual/Yearly Returns and Risk /Standard Deviation(S.D)	:	8
3.2 Monte Carlo Simulation	:	8
3.3 Python Implementation	:	9
3.4 Front End Development	:	12
3.5 Flask Integration	:	13
 CHAPTER 4: RESULTS AND DISCUSSION		 15-18
4.1 Python Implementation Results (executed in jupyter)	:	15
4.2 Web Application Results for different Ticker Symbols	:	18
 CHAPTER 5: CONCLUSION & FURTHER WORK		 19-20
5.2 Conclusion	:	19
5.2 Future Work	:	19
5.3 Planning And Project Management	:	19
 REFERENCES		 21

LIST OF FIGURES

Figure ID	Figure Title	Page
2.1	Efficient Frontier and CAL	7
3.1	Historical Data(Stock Prices) of selected Ticker symbols	9
3.2	Mean Returns & Covariance Matrix	9
3.3	Ret, Stdev & Sharpe of the selected instruments stored in a dataframe for various weights	10
3.4	Output of Portfolio with lowest risk & max expected return	11
3.5	Plot of 10000 Portfolios Constructed	11
3.6	Info Page	12
3.7	About us Page	12
3.8	Application Page	13
4.1	Output for Portfolio 1	15
4.2	Output for Portfolio 2	16
4.3	Output for Portfolio 3	16
4.4	Output for Portfolio 4	17
4.5	Output for Portfolio 5	17
4.6	Output for case 1 (ABC JPM AMZN)	18
4.7	Output for case 2(A ABC JPM BAC MS)	18
5.1	Gantt Chart	20

LIST OF TABLES

Table ID	Table Title	Page
Table 5.1	Showing details about project planning and management	24

CHAPTER 1

INTRODUCTION

The modern portfolio theory was devised by Henry Markowitz in 1952, for which he won the 1990 Nobel Prize in Economics.

The Modern Portfolio Theory is based on the assumption that the nature of investors is that of being risk averse, i.e given 2 portfolios that offer same expected returns, a rational investor shall prefer the one with minimal risk. Thus, an investor is willing to take a higher risk only if the same is compensated by a higher expected return. Similarly, if an investor aims for greater expected returns he/she must be willing to accept a higher associated risk^[1].

The explicit trade-off between risk/standard deviation and Expected Returns cannot be generalized for all investors.^[3] Depending on the risk aversion characteristics on an individual basis, different investors are bound to evaluate the trade – off differently. The implication being a rational investor shall not decide invest in a said portfolio given that there exists another portfolio with a more advantageous expected return-risk profile.

CHAPTER 2

BACKGROUND

Under the model

The proportion- weighted mix of the constituent assets of a given portfolio is the expected return of the portfolio.

The function defined as the the correlations ρ_{ij} of the constituent assets, asset pairs (i, j) is called the portfolio volatility.

2.1 Expected return:

$$E(R_p) = \sum_i w_i E(R_i)$$

In the above expression, R_p is the expected return on the portfolio 'P', R_i is the expected return on asset i with w_i as the associated weight of the component asset i.

2.2 Portfolio return variance: It is the measure of risk associated with a portfolio in this model. A higher variance is indicative of a higher risk associated for the given asset class and the portfolio.

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij},$$

where σ_i is the standard deviation of periodic returns of the given asset, and ρ_{ij} is the correlation coefficient between the returns on assets i and j. Alternatively the

expression can also be written as:

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij}$$

where σ_{ij} for $i = j$, or

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij}$$

where σ_{ij} is the co-variance of periodic expected return of the 2 assets, alternatively denoted as cov_{ij} or $\text{cov}(i,j)$.

2.3 Portfolio return volatility(standard deviation):

$$\sigma_p = \sqrt{\sigma_p^2}$$

2.4 Sharpe Ratio : It is the measure of ROI in relation to the Treasury rate/ Risk free rate & its risk profile. In general a higher value of SR indicates a better & more lucrative investment. It can be said that when comparing 2 portfolios with similar risk profiles, all else equal it would be better to invest in the portfolio having a higher Sharpe Ratio.^[2]

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

R_p is the return of portfolio 'P'
 R_f is the risk free rate
 σ_p Standard deviation or risk associated with the portfolio's excess return.

2.5 The Efficient Frontier :

It is the plot measure of risk vs expected returns and is deployed to identify the most optimum portfolio to invest into upon considering the risk profile and the characteristics of the given investor. The efficient frontier is an essential part of the curve in the 1st and 2nd quadrants that depends on investor objectives and characteristics.

The CAL(Capital Allocation line) is a tangent drawn to the efficient frontier and the intersection point is appraised to be the optimal investment i.e its the portfolio that has highest expected returns for a given level of risk, under normal circumstances.^[3]

The MPT is instrumental for investors keen on diversifying their portfolios. As a matter of fact the growth of ETFs (Exchange Traded Funds) made the theory more relevant by allowing investors to have an easier access to different asset classes.^[4]

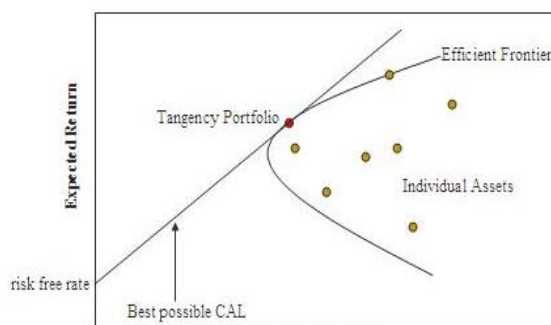


Fig 2.1 : Efficient Frontier & CAL

Equity Investors manage risk using MPT by putting a small chunk of their portfolios in Govt. EFTs. This way the variance of the portfolio becomes significantly lower since govt. bonds have a –ve correlation with equity instruments. This ensures that there's no large impact on the expected return because of this loss reducing effect.^[5]

CHAPTER 3

PROJECT ANALYSIS/ PROJECT IMPLEMENTATION

3.1 Annual/Yearly Returns and Risk /Standard Deviation(S.D)

In this paper we have considered the quotidian returns and corresponding standard deviation for 3 month of stock data. However, in practice, institutions work with yearly returns and S.D. The formulae used for converting quotidian returns and S.D to a yearly basis is as shown :

$$R_A = R_d * 252 * \sigma_A = \sigma_d * 252$$

Where,

R_A is the annual returns

R_d is the daily returns , σ_A is the annual standard deviation & σ_d is the daily standard deviation.

The no. of active trading days is taken to be 252 for a given year.

A data set containing the % change of adj. Closing price of ~200 equity instruments was first created.

Initially random weights are assigned to the above stocks selected for a particular portfolio keeping the sum of weights to be 1. The expected return and S.D of the portfolio is then calculated and stored.

Monte Carlo Simulations is then used on the portfolio to get optimal weights for each equity instrument in our portfolio in python.

3.2 Monte Carlo Simulation

Monte Carlo simulations is a statistical method of computation that deploys a vast number of random samples to obtain results. In this simulation, random weights are assigned to the equity instruments keeping the sum of weights to be 1. The expected return and S.D is then calculated for every combination of these weights and stored. Then weights are then again changed, assigned randomly and the process is repeated. The number of iterations depends on the error that the investor is willing to accept. With the increase in number of iterations, the accuracy of the optimization will also increase but at the cost of computation and time.

In this paper we have considered 10000 such iterations, out of these 10000 results of Expected return and risk profile generated, portfolio optimization can be achieved by identifying the portfolio which satisfies any of these 3 conditions:

1. A portfolio that has the lowest risk associated for desired level of expected return.
2. A portfolio that gives the highest expected return for desired risk level.
3. A portfolio that has the utmost Expected return to S.D/risk ratio, also known as Sharpe ratio.

3.3 Portfolio Optimization process in Python

The stock price data was fetched from the data set using pandas library.

```
Enter Ticker Sybmols :GS MS BAC
      GS      MS      BAC
0  0.000000  0.000000  0.000000
1 -0.011453 -0.023563 -0.013283
2  0.021241  0.023497  0.013462
3 -0.012042 -0.022544  0.000415
4  0.006799  0.015023  0.004564
..      ...      ...      ...
58 -0.009410 -0.007577 -0.005550
59  0.061331  0.056871  0.037321
60 -0.026811 -0.017544 -0.017821
61  0.025431  0.023559  0.028757
62 -0.001131 -0.001759 -0.002995
```

Fig 3.1 – Historical Data(stock prices) of selected Ticker Symbols

The mean return of all stocks in the portfolio and the corresponding covariance matrix was then generated.

```
#Calculate mean returns and covariances of all four the stocks
mean_returns = stock_ret.mean()
cov_matrix = stock_ret.cov()
print (mean_returns)
print (cov_matrix)
```

```
GS      0.004277
MS      0.005801
BAC     0.003896
dtype: float64
      GS      MS      BAC
GS  0.000372  0.000386  0.000388
MS  0.000386  0.000504  0.000472
BAC 0.000388  0.000472  0.000696
```

Fig 3.2 – Mean Returns and Covariance Matrix

The results of each iteration were stored in an array. The number of columns in the array will change with the no. of stocks in the portfolio, since the weights of all the stocks are to be stored.

The len function has been used while defining the array. The no. of rows in the created array is equal to the no. of iterations generated.

*#Set the number of iterations to 10000 and define an array to hold the simulation results;
initially set to all zeros*

```
num_iterations = 10000
simulation_res = np.zeros((4+len(stock)-1,num_iterations))
for i in range(num_iterations):
#Select random weights and normalize to set the sum to 1
weights = np.array(np.random.random(4))
weights /= np.sum(weights)
#Calculate the return and standard deviation for every step
portfolio_return = np.sum(mean_returns * weights)
portfolio_std_dev = np.sqrt(np.dot(weights.T,np.dot(cov_matrix,
weights)))
```

#Store all the results in a defined array

```
simulation_res[0,i] = portfolio_return
simulation_res[1,i] = portfolio_std_dev
#Calculate Sharpe ratio and store it in the array
simulation_res[2,i] = simulation_res[0,i] / simulation_res[1,i]
#Save the weights in the array
for j in range(len(weights)):
simulation_res[j+3,i] = weights[j]
```

The output was saved in a 'pandas data frame' for easy analysis and plotting.

	ret	stdev	sharpe	GS	MS	BAC
0	0.005209	0.021289	0.244708	0.230923	0.643291	0.125787
1	0.005156	0.020715	0.248911	0.375565	0.586376	0.038058
2	0.004659	0.021251	0.219256	0.333497	0.334078	0.332425
3	0.004862	0.020633	0.235662	0.411207	0.425012	0.163782
4	0.005152	0.020589	0.250244	0.414418	0.576573	0.009009
...
9995	0.004623	0.023068	0.200402	0.079997	0.365612	0.554391
9996	0.004187	0.024830	0.168618	0.036315	0.145487	0.818197
9997	0.004799	0.021802	0.220102	0.206152	0.432601	0.361246
9998	0.004487	0.023450	0.191339	0.078892	0.294470	0.626638
9999	0.004080	0.022674	0.179926	0.334370	0.029544	0.636087

[10000 rows x 6 columns]

Fig 3.3 – Ret, Stdev & Sharpe of the selected instruments stored in a dataframe for various weights

The above figures show the output where s some rows of the simulation results are shown.

```
ret      0.573113
stdev    2.222583
sharpe   25.785902
GS       4.278413
MS       95.477021
BAC      0.244566
Name: 8661, dtype: float64
ret      0.427049
stdev    1.931963
sharpe   22.104406
GS       97.060515
MS       0.230914
BAC      2.708570
Name: 3654, dtype: float64
```

Fig 3.4 –Output of Portfolio with lowest risk & max expected return

Portfolios having utmost Sharpe ratio or lowest associated risk was then computed.

The output was then plotted using the matplotlib library as the relevant points can be highlighted as shown

In the output plot generated, the red star indicates the portfolio with the highest Sharpe ratio, the blue star indicates the portfolio with lowest standard deviation or risk.

From the plot generated above, we deduce the composition of the optimal portfolio required on the basis of any of the above mentioned 3 criteria.

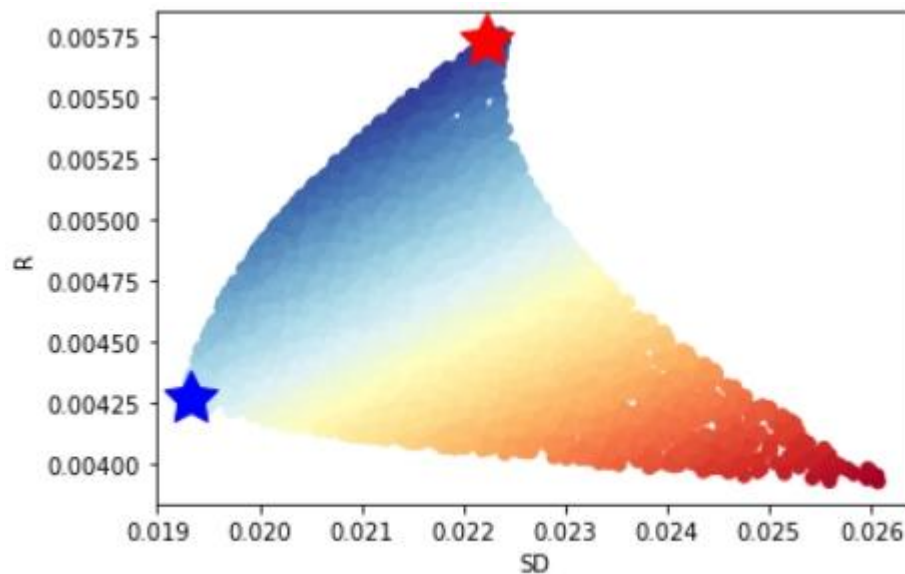


Fig 3.5 – Plot of 10000 portfolios constructed

3.4 Front End Development

The Front End Application has been made using HTML & CSS. The Front End of the Application consist of 3 pages. An Intro Page, an 'about us' page and the application page.

Intro Page

This is the introductory page of the web application. The page comprise of a brief description of the project and buttons which act as hyperlinks to other pages, this

Page also has a slider comprising of 3 images.

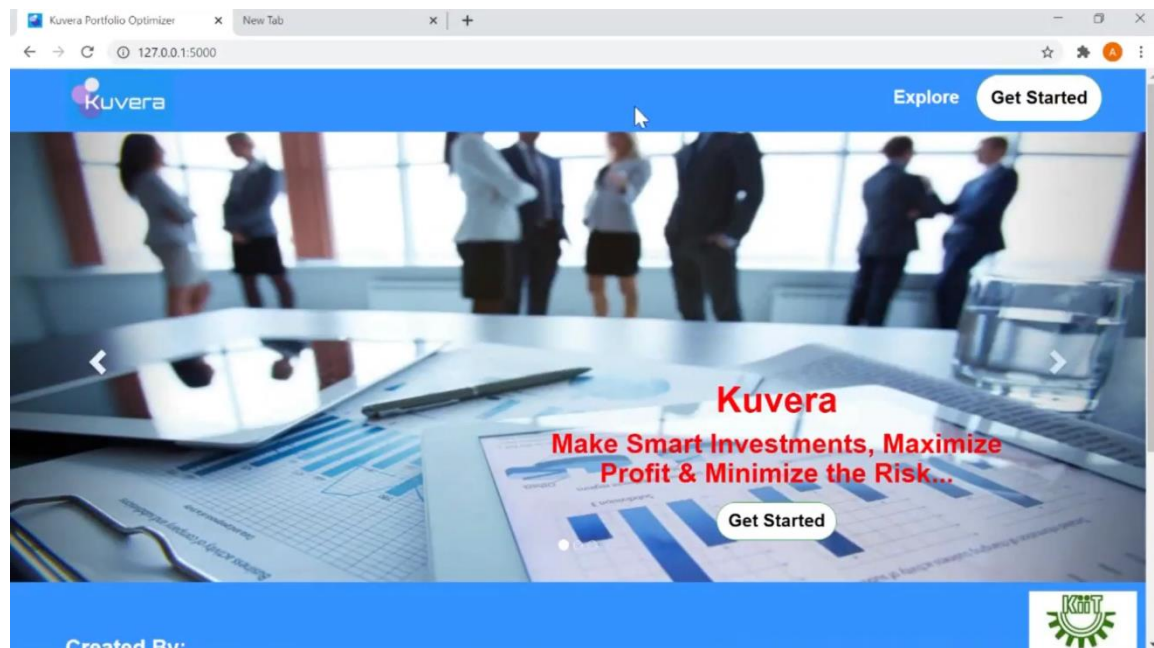


Fig 3.6 – Intro Page

About Us Page

This page consists of the abstract of our work and a button which acts as a link to the paper.

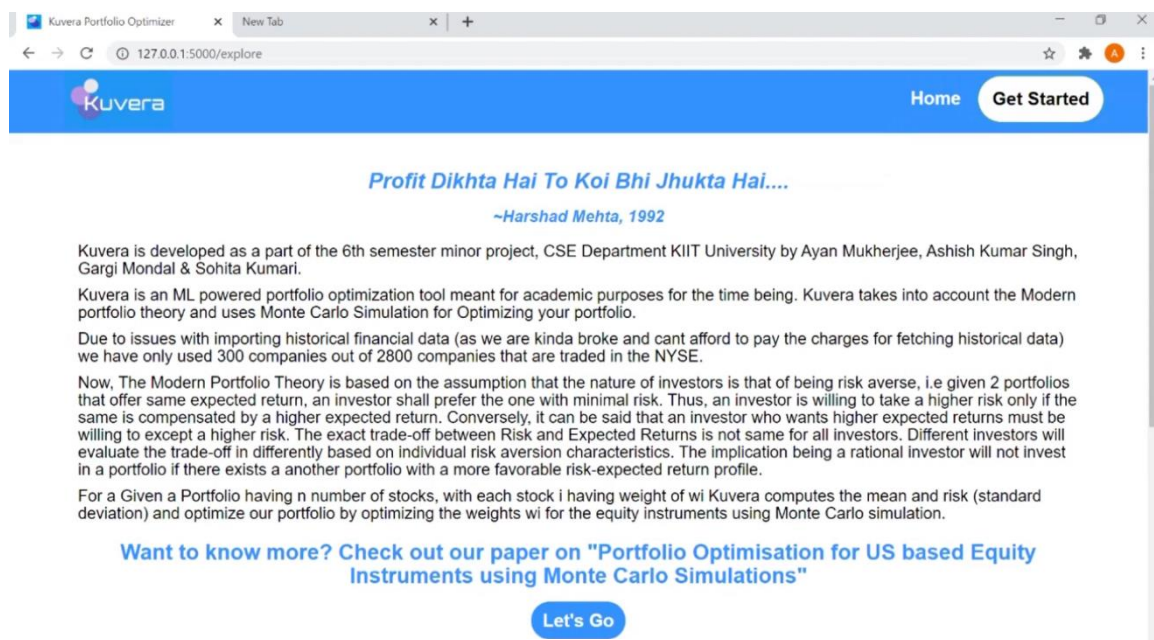


Fig 3.7 – About Us Page

Application Page

This is the application page, on the left hand side of the page is a widget from trading view which displays the major indices across the globe, on the left hand side is a

Input bar, here the user can input the ticker symbols of the equity instruments separated by spaces.

There are two buttons, 'submit' and 'reset', upon hitting the 'submit' button the Optimal portfolios based on MPT is computed in the backend and the results are displayed(the results are shown in the results chapter).

Upon hitting the 'reset' button the page is reset and the input bar is cleared.

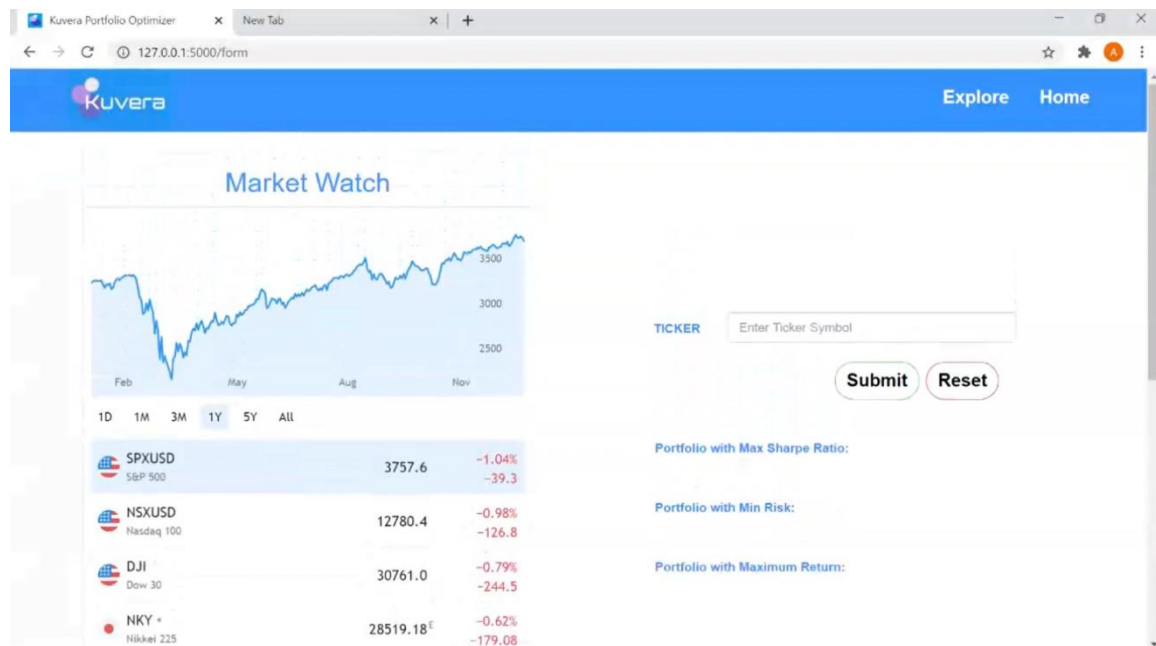


Fig 3.8 – Application Page

3.5 Flask Integration

The Front End and the ML code have been integrated using the flask framework

File Structure

```
/home/user/Projects/Minor_Project
├── templates/
│   ├── intro.html
│   ├── application.html
│   └── aboutus.html
├── static/
│   └── style.css
├── venv/
├── app.py
├── model.py
└── Data.csv
```


Venv : a Python virtual environment where Flask and other dependencies are installed.

Model.py : The Python model for portfolio optimization

Data.csv : the dataset

Templates : the html files/ front end of the application

App.py : it contains the instance of the **Flask** class. Everything about the application, such as configuration and URLs, is registered with this class. It acts as the central configuration object for the entire **application**. It's used to set up pieces of the **application** required for extended functionality.

The following code snippet illustrates the results generation functionality:

```
@app.route('/result', methods=['POST', 'GET'])
def result():
    name = request.form.get("var_2", type=str)
    sharpe, stdev = model(name)
    return render_template('form.html',
pred="{0}".format(sharpe), pred2="{0}".format(stdev))
```

App routing is used to map the specific URL with the associated function that is intended to perform some task. The **route()** decorator tells Flask what URL will trigger our function.

The function is given a name which is also used to generate URLs for that particular function, and returns the message we want to display in the user's browser.

The Ticker Symbol names which is the input is taken from the Result using 'GET' method using the command `request.form.get()` and stored in the 'name' variable.

The model is called by passing 'name' through the function `model()` which is declared in Model.py.

The output from the `model()` are stored in the variables 'sharpe' and 'stdev'. The **render_template()** is used to generate output from the template file based on the Jinja2 engine that is found in the application's templates folder.

CHAPTER 4

RESULTS & DISCUSSIONS

4.1 Python Implementation Results (executed in jupyter)

Multiple optimized portfolios have been generated using the same by taking equity instruments from namely 2 sectors, the results of which are shown below.

1. Technology Sector
2. Banking Sector

The equity instruments considered are tradable shares of companies from the above two mentioned sectors that are listed in the NYSE (New York Stock Exchange)

We have constructed 5 optimal portfolios, 3 from the Technology Sector and 2 from the Banking Sector.

Portfolio 1 : Technology Sector

Stocks selected :

AMZN (Amazon), ABC(Alphabet Inc./Google), ORCL (Oracle)

```
ret      0.147915
stdev    1.114092
sharpe   13.276734
AMZN     0.957212
ABC      0.585146
ORCL     98.457642
Name: 7079, dtype: float64
ret      0.088006
stdev    0.959656
sharpe   9.170534
AMZN     16.331346
ABC      28.927542
ORCL     54.741113
Name: 4573, dtype: float64
```

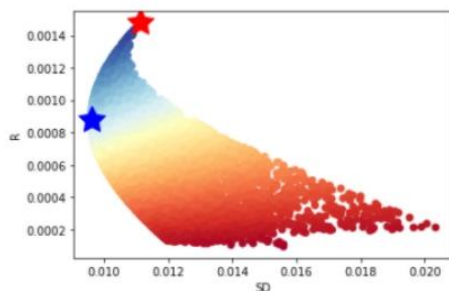


Fig 4.1 – Output for Portfolio 1

Portfolio 2 : Technology Sector

Stocks selected :

AAPL(Apple), ABT (Abbott Labs), AMZN(Amazon), ABC(Alphabet Inc/Google)

```
ret      0.203204
stdev    1.809088
sharpe   11.232391
AAPL     81.279090
ABT      4.188146
AMZN     1.263229
ABC      13.269534
Name: 8226, dtype: float64
ret      0.033086
stdev    1.098877
sharpe   3.010923
AAPL     0.579554
ABT      31.033269
AMZN     22.201257
ABC      46.185920
Name: 5717, dtype: float64
```

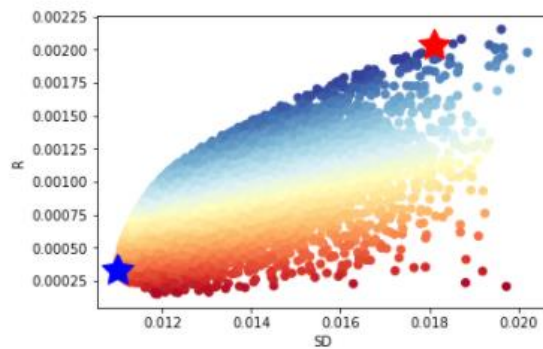


Fig 4.2 – Output for Portfolio 2

Portfolio 3 : Technology Sector

Stocks selected :

ADI (Analog Devices Inc.), ADSK (Autodesk Inc.), AAPL(Apple Inc.)

```
ret      0.395952
stdev    1.536657
sharpe   25.767136
ADI      47.724207
ADSK     52.188089
AAPL     0.087704
Name: 2234, dtype: float64
ret      0.367656
stdev    1.492286
sharpe   24.637080
ADI      52.428587
ADSK     34.463727
AAPL     13.107686
Name: 3042, dtype: float64
```

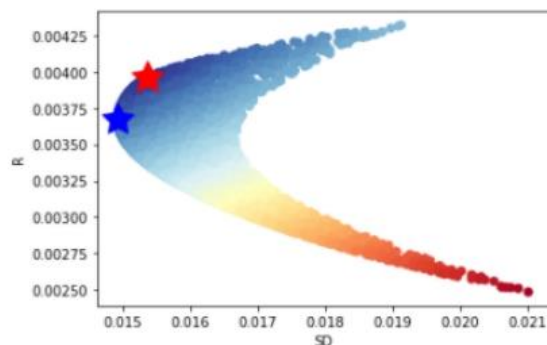


Fig 4.3 – Output for Portfolio 3

Portfolio 4 :Banking Sector

Stocks selected :

BAC (bank of America), JPM (J. P. Morgan), GS (Goldman Sachs), MS (Morgan Stanley)

```
ret      0.327406
stdev    1.231474
sharpe   26.586508
BAC      0.978128
JPM      49.574441
GS       1.795954
MS       47.651477
Name: 3860, dtype: float64
ret      0.115184
stdev    0.558013
sharpe   20.641878
BAC      1.097631
JPM      90.882820
GS       5.253932
MS       2.765617
Name: 5292, dtype: float64
```

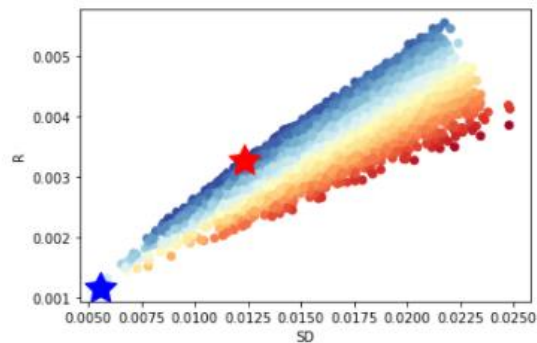


Fig 4.4 – Output for Portfolio 4

Portfolio 5 :Banking Sector

Stocks selected :

C (Citi Bank), WFC (Wells Fargo), BAC(Bank of America)

```
ret      0.578217
stdev    2.514889
sharpe   22.991769
C        97.577188
WFC      0.684877
BAC      1.737935
Name: 755, dtype: float64
ret      0.513693
stdev    2.485928
sharpe   20.664017
C        63.562355
WFC      10.898885
BAC      25.538760
Name: 1816, dtype: float64
```

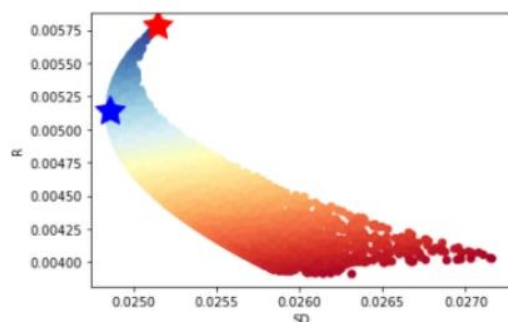


Fig 4.5 – Output for Portfolio 5

4.2 Web Application Results for different Ticker Symbols

Case 1:

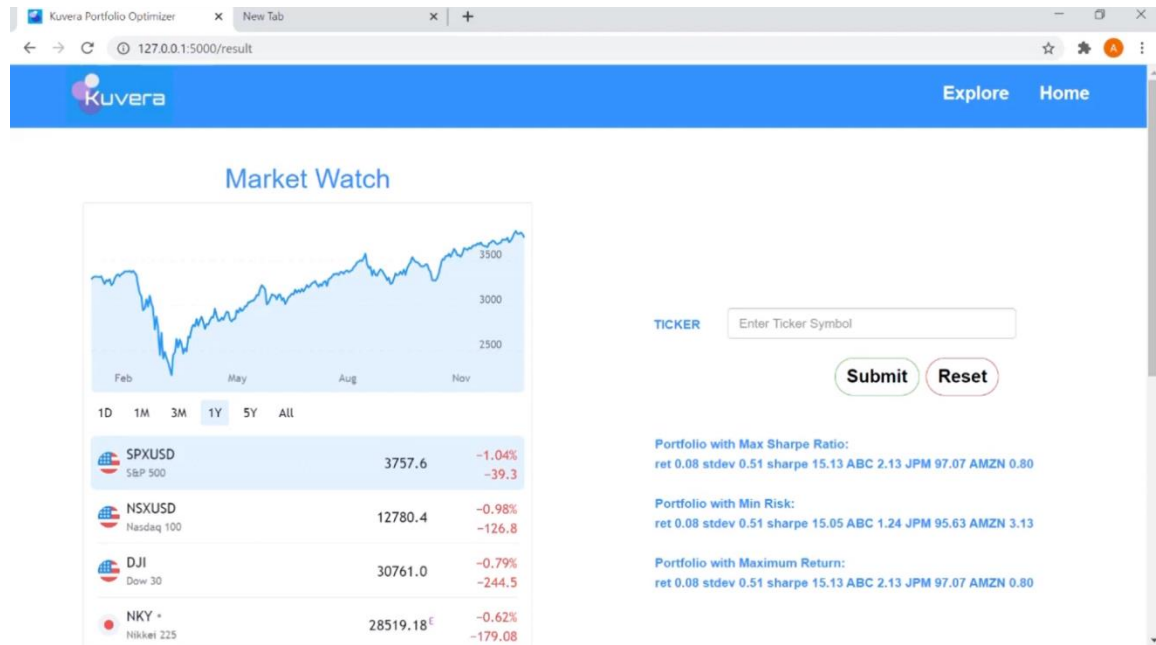


Fig 4.6 – Output for case 1(ABC JPM AMZN)

Case 2

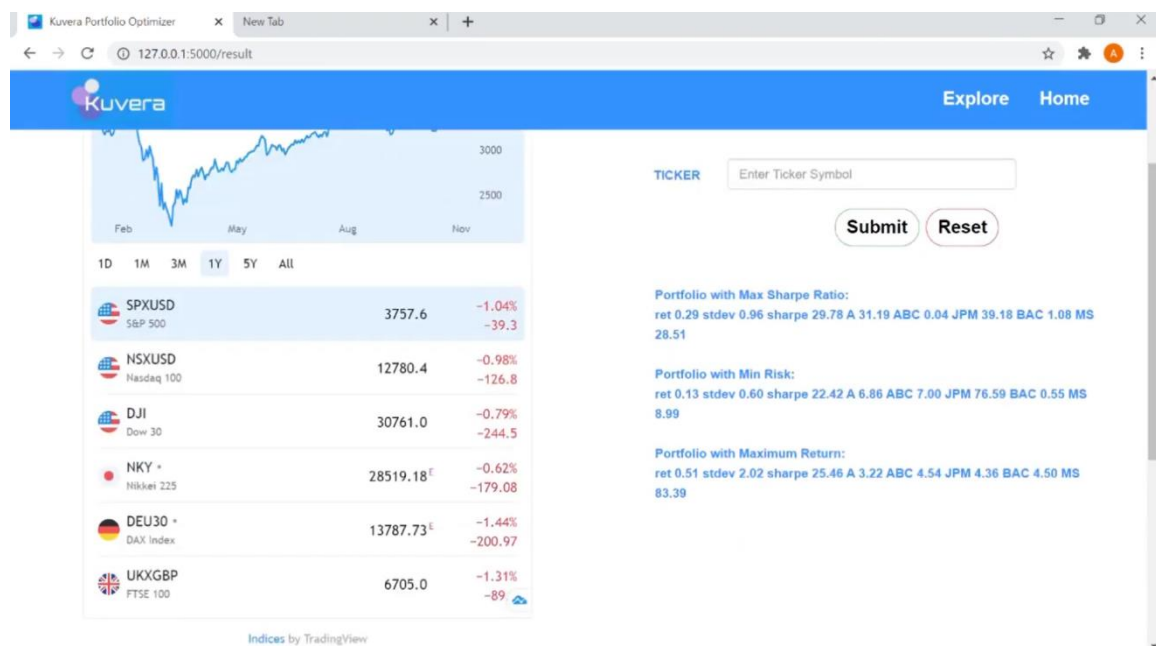


Fig 4.7 – Output for case 2(A ABC JPM BAC MS)

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The literature associated with portfolio optimization is rich and vast. There is a broad variety of variations and advancements upon the basic methods and a lot of active research that goes around it. In this paper we have used Monte Carlo simulation to optimize our portfolio that consisted of US based equity instruments from selected sectors, the same can be applied not only to equity instruments from wide array of sectors but also to different asset classes so as to diversify investor portfolio and improve the Risk to Expected Return Trade-Off.

5.2 Future Work

The application is currently hosted locally, deploying the application in cloud with and increasing the dataset will help cater to the needs of a larger group of audience. Currently the application has been made for US based Equity Instruments, the same can be extended to other financial markets globally like the NSE, BSE and LSE. Also including other asset classes than equity like commodities is a promising area of work.

5. 3 Planning And Project Management

Table 5.1 showing details about project planning and management

Activity	Starting week	Number of weeks
Literature review	1 st week of October	4
Finalizing problem	1 st week of November	1
Excess and permission for required software	2 nd week of November	2
Dataset Creation using Yahoo Finance Historical data	1 st week of December	2
ML Model development for Portfolio Optimization in Python	3 rd week of December	2
Model Testing	1 st week of January	2

Front End Development for Web Application	3 rd week of January	2
Flask Integration	1 st week of February	2
Web Application Testing	2 nd week of February	2
Preparation of project report	1 st week of March	3
Preparation of project presentation	1 st week of April	2

The Gantt chart is shown below:

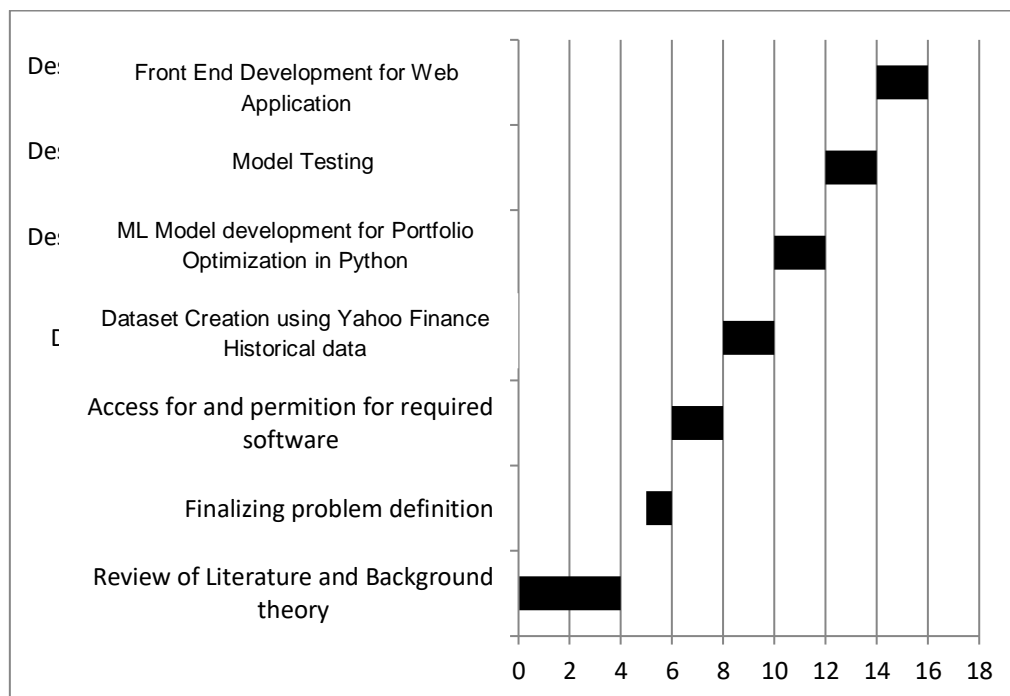


Fig 5.1: Gantt chart

REFERENCES

1. Markowitz, H.M. (March 1952). "Portfolio Selection". *The Journal of Finance*. 7 (1): 77–91. doi:[10.2307/297597](https://doi.org/10.2307/297597)
2. Markowitz, Harry (February 2000). *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Wiley. ISBN 978-1-883-24975-5.
3. Low, R.K.Y.; Faff, R.; Aas, K. (2016). "Enhancing mean–variance portfolio selection by modeling distributional asymmetries" (PDF). *Journal of Economics and Business*.
4. Risk Manager Journal (2006), "New Approaches for Portfolio Optimization: Parting with the Bell Curve — Interview with Prof. Svetlozar Rachev and Prof. Stefan Mittnik
5. Doganoglu, Toker; Hartz, Christoph; Mittnik, Stefan (2007). "Portfolio Optimization When Risk Factors Are Conditionally Varying and Heavy Tailed

INDIVIDUAL CONTRIBUTION REPORT:

**Flask based web application for portfolio optimization for US based
equity instruments using Monte Carlo Simulations**

Ayan Mukherjee
1805027

Abstract: MPT or Modern Portfolio Theory also known as mean-variance analysis is a mathematical modeling technique that is deployed in constructing portfolios that can maximize the portfolio return for a given amount of risk. In this paper we optimize portfolios in accordance to the Modern Portfolio Theory for US based equity instruments using Monte Carlo Simulations.

For a given Portfolio 'P' having 'n' number of stocks, with each stock 'i' having weight of ' w_i ' we compute the mean and risk (standard deviation) and optimize our portfolio by optimizing the weights ' w_i ' for the equity instruments using Monte Carlo simulation.


Individual contribution and findings: My contribution helped make the team project a success. I encouraged team members to brainstorm and I provided ideas and inputs for the project. I spent a considerable amount of time guiding other team members on specific tasks of the project, concerning my area of expertise, to assist the team in reaching project goals.

I demonstrated strong leadership skills throughout the project, in multiple project areas. In week two, I volunteered to be the team leader. As the team leader, I assigned each team member a portion of the assignment to accomplish, pulled together everyone's inputs and designed the project plan. I designed the primary dataset and the final dataset for the project, the final dataset comprising of historical price data of nearly 200 equity instruments from NYSE taken over a 3 month period. I worked on the development of the ML model for this project and the backend integration using flask.

Individual contribution to project report preparation: The areas concerning the python model development and flask integration along with testing results of chapters 2, 3 and 4 was contributed by me, additionally as the team leader I was entasked with the project planning and thus helped in finalizing the project plan as displayed in chapter 5. I proofed the report for errors, posted the proofed report for review and team members' approval, and posted the finished product to the project-write-only folder.

Individual contribution for project presentation and demonstration: Demonstration of the web application using flask was carried out by me. Explained the functionality of the web application and the back end codes during presentation. Also Explained the MPT and our work in brief manner along with the future scope of work.

Dr. Pradeep Kumar Mallick
School of Computer Engineering


Ayan Mukherjee 1805027