|  |
| --- |
|  |
| MGMT59000 - 044 |
| Computing for Analytics |
|  |
|  |
| **10/12/2019** |

Portfolio Optimization and Predicting Stock Price

|  |
| --- |
|  |

Team 1:

Arun Ramakrishnana

Sesha Sai Tumma

Abhinav Chanda

Cheng Cheng

Sagar Kurada

Maharshi Dutta

Introduction

A portfolio is any combination of financial assets such as cash, stocks and bonds. Portfolio optimization is the process of choosing the best asset distribution from a set of portfolios held to a particular objective. It is an extension of the Modern Portfolio Theory introduced by Harry Markowitz, which assumes that an investor wants to maximize the expected return of a portfolio contingent to any given constraint for risk. For achieving higher returns, investors are faced with the problem of taking on more risk. Portfolios which fall under the constraints of expected returns and risk are called efficient portfolios. The expected returns-risk relationship of efficient portfolios is represented by a curve known as the efficient frontier, and these portfolios are said to be well-diversified. The objective of any algorithm is to clearly plot this efficient frontier which gives the investor a clear indication of the volatility versus returns of risk.

Our objective as part of this project was to utilize the concepts of data structures, optimization, simulation and visualization clubbed with multiprocessing to come up with an effective model for portfolio selection and prediction of stock price for efficient returns.

#### Business Objective:

#### The main objective in today’s financial landscape is very simple – to get maximize returns at lesser risk. At first, this is would sound straightforward, however it is this balance which people are still thriving to achieve. Also construction and portfolio selection is an investment strategy that allows investors to optimize the overall rate of return investment for the risk. Hence a solution which works to resolve these challenges is of utmost importance and greatly valued in the financial management. This forms our core outline for picking this topic.

#### Results and Findings:

#### As part of the project we tried to solve the business problem highlighted above with the application of modern analytical tools and computational processes.

#### For step 1 we retrieved the data of multiple securities for various companies like Apple, Google and Netflix etc. by application of web scrapers. The data for the previous 10 years was picked up from Yahoo Finance and then processed by our application. The idea behind building up an analytical solution was to randomize the generation of portfolio’s comprising of securities from various companies to best fit constraints specified by an investor.

#### Once we were able to build multiple sets of random portfolios using simulation – the goal was to get the optimized portfolio with a balance between high returns and low risk. This was done using optimization techniques to return portfolio’s with high sharp ratio’s or low volatility. Random weights were assigned at each simulation of portfolio generation and the model was a built factoring into account all possible combinations. Once this was achieved, the application would then go on and predict stock price from the optimal solution over a period of time to give the investor a complete picture of the return on investment done.

#### Main Findings:

1. Maximum sharp ration or minimum volatility were selected as the 2 best parameters for optimal portfolio selection
2. Widgets, the backbone for this dynamic implementation of the interface. The tab ipy-widgets would bring quality and operationally effectiveness, fetching results in milliseconds.
3. Multiprocessing was found to be extremely efficient over large results
4. The Modeling with LSTM returned accurate prediction of stock price when fed in with specific portfolio.

#### Other Studies:

#### One of the important works, i.e. distance function is the Luenbgerger (1992, 1995) one. This function measures the distance of an in-eﬃcient unit (production plan) from the eﬃcient frontier with a given direction. Chambers, Chung and Färe (1996) introduced the directional distance function which measured technical eﬃciency based upon simultaneously reducing inputs and increasing outputs of production. In that way, distance function represents an indicator of ineﬃciency. Some of their key findings include - preferences of investor’s which play one of the most major roles in choosing the direction in which improvements should be made. This research shows some possibilities by changing the risk aversion of the investor, as well as using additional transaction costs and annualized return as starting points on how to choose the direction in which the restructuring portfolios.

Computational Steps

The following computational steps were carried out for the implementation of the project:

1. **Interactive UI** – An interactive UI was built to give the user a feasibility to select the stocks from the 500 stocks scraped from the Yahoo. The tab ipy widget was used.
2. **Data Retrieval** -- This involves looking up financial data of stocks from Yahoo Finance for multiple companies. Since the data was already in a clean, structured, and in standardized format, no additional data cleaning methods. The data timeliness was daily level. The data was read using pandas web data reader.
3. **Random Portfolio generation using Montecarlo simulation** – This function involves generation of random portfolio’s by assigning a weight of 1 across the number of stocks selected by the user. Trading days was kept to 252 for the year– and an additional function would take these stocks into account and generate the annual return and volatility of each portfolio. Random weights were assigned using the python numpy library and executed with multiple simulations.
4. **Optimizatio*n*** *–* Following the random portfolio generation, a function was developed which would take into consideration all the above information like number of portfolio’s, stocks considered, risk free rate. The Daily returns and covariance of the matrix of returns are calculated also as function input. Using the same, the function would output the 2 most optimal portfolios – one with maximum sharp ratio and second with minimum volatility. It would also clearly plot the efficient frontier for the all combinations thus helping resolve the investor’s dilemma*.*
5. **Multiprocessing** – One of the most important aspects while working on this project, was the application of multiprocessing – this was done both on the random generation of portfolio’s and well as the optimization process. The python multiprocessing library was used. Also since jupyter IDE is incapable of executing parallel processing – a Randomport.py package was also built which needs to be imported for execution and a total of one thousand random portfolio’s was generated for one thousand simulations.
6. **Data Visualization –** The data visualization takes into account python interactive plotting and generates various graphs starting from performance of stocks for previous years to cumulative returns over each day, a portfolio distribution chart and also prediction charts for return. Bokeh and plotly library was used for building interactive plots and 3D charts.
7. **Modeling –** Post data generation and optimization of portfolio’s a predictive model was built to forecast stock price across a given time series. Keras library was used for this purpose. The Long Short-Term Memory (LSTM) – is an advanced neural network architecture in the field of deep learning.

**Computational challenges:**

In terms of computational challenges the following challenges were faced:

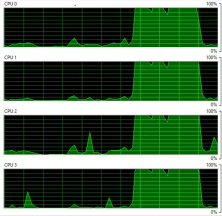
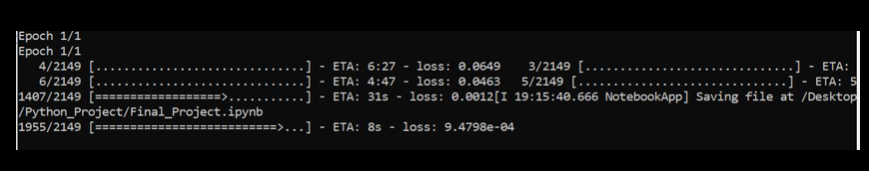
1. During our analysis, we have tried multiple widgets- all were time consuming and weren’t viable for computation. Not suited for the “list” data type which we’ve used for storing the stocks
2. The other challenge we faced was while executing the simulation across a large number of portfolio generations where the randomization and optimization was slow. Multiprocessing was a significant difference in the performance of the code
3. Automation of the Data Model for prediction of returns was not very feasible. Any automation led on random portfolio was sub optimal.

**Code Performance:**

All part of the project were duly optimized and code performance was taken into consideration while development. In order to run simulations for a significantly large number of portfolio selections – multiprocessing was applied. The application of multiprocessing was observed to make notable increase in performance as compared to processing without parallel processing.

However, even with multiprocessing over 8 cores – and running 1 Mn combinations of simulations – the randomization and optimization of portfolio’s was slow and lead to considerable decrease in system functioning.

When increasing data or the number of simulations, the process was seen to further slow down utilizing the entire CPU memory – as seen in the figure below

  Fig1: Computational difference post Multiprocessing

Results

The following plots were made to capture our results and findings:

* *Interactive UI* : This part explains the feature of the project where the user can select the stocks the he wants to generate the Portfolio’s for:

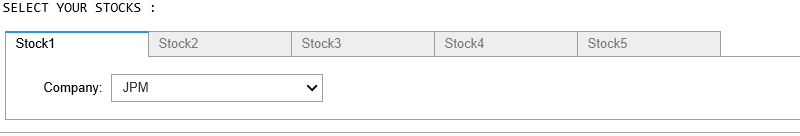


Fig 1: Interactive UI for stock selection

* *Data descriptive Statistics:* The plot captures the performance of the stocks over a period of 10 years time period for which the data has been scraped from Yahoo Finance. There is a slider which enables time comparison to view stock performance for the time period.
* *5 Stock Cumulative Return* : This plot enables to visualise the value of the portfolio when we select 5 stocks for $1 budget:

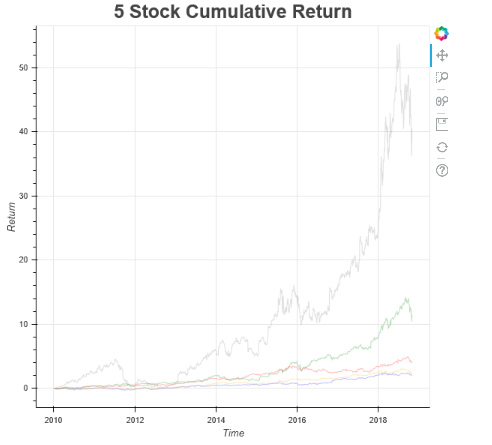
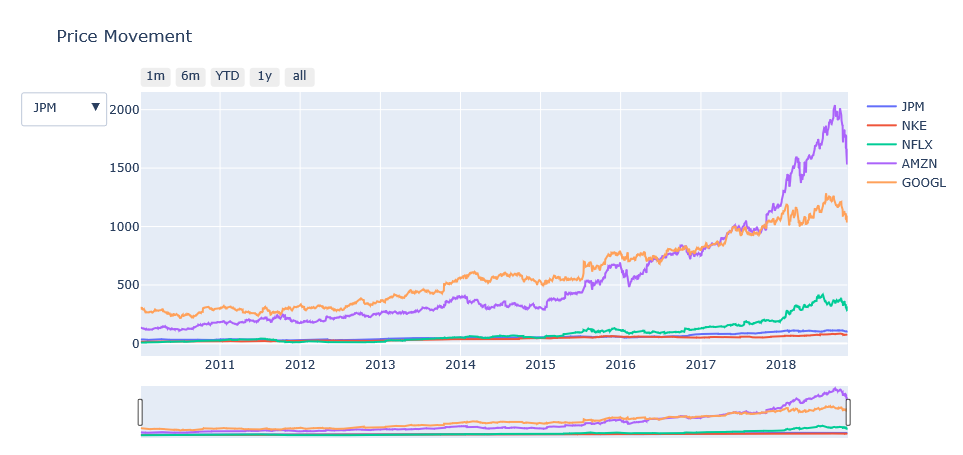


Fig 2: Stock Performance Fig 3: Cumulative Stock Return

* *Optimized Portfolio returns:* The plot gives a clear distribution of weights in terms of maximum sharp ration for selection of optimal Portfolio post running Monte Carlo simulation. Along with the same, we have the plot for the efficient frontier – which gives the point of maximum return versus minimum risk.

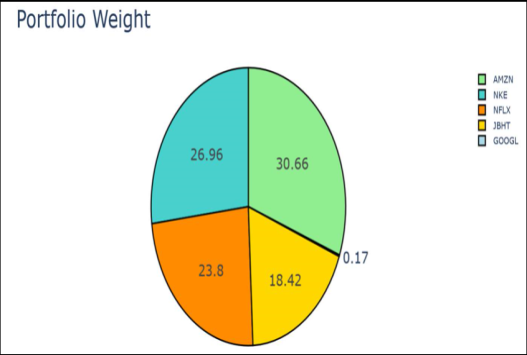
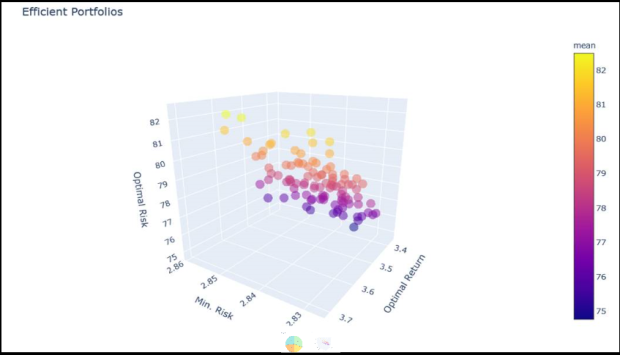
 

Fig 4: Max Sharp Ratio Fig 5: Efficient Frontier

* *Multiprocessing Time comparison:* This plot defines the time comparison between running parallel processing for simulating random portfolio
* *Stock Price forecasting model:* The plot displays the forecasting model which predicts the price of stock for various companies selected. The plot projects a 3 month prediction which helps investors get a quarterly view of the price.

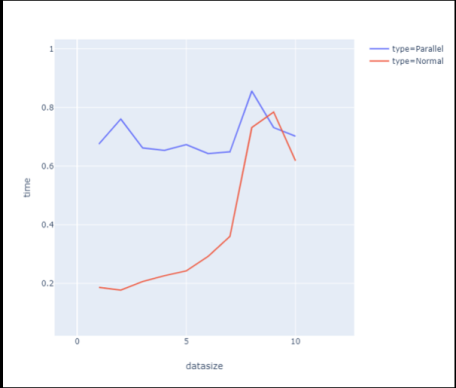
 

Fig 6: Multiprocessing of Simulation/Optimization Fig 7: Forecasted Stock Price

Conclusion Summary

As part of the project we made the following conclusions:

1. Based on the history of the stocks and an interactive UI with feature to select personalized stocks – the user an optimized portfolio was selected based on 2 parameters:
2. Maximum Sharp Ratio
3. Minimum Volatility
4. Simulation and optimization – This resulted in the selection of the most optimal solution as per the above logic as criteria over number of iterations
5. Multiprocessing resulted in significant efficiency increase in data processing
6. Predictive modeling resulted in accurate stock price predictions and this was visualized over a time frame.
7. Stock price forecasting model was able to project next quarter rates for portfolio selection

**Interesting Hypothesis/Questions for future analysis/Computational Challenge**

The below questions to be answered for future scope:

1. Automation of stock price prediction over large securities for accurate results
2. Including various market risk parameters like foreign exchange would bring in additional complexity while building predictive model
3. Random portfolio generation according to weighted average of stocks is optimal but needs to be scalable to include futuristic change in trading methodology
4. Currently available software runs into difficulty with more than about 600 securities to compute the efficient frontier
5. Ability of standard IDE’s to run multiprocessing in an efficient way

Appendix

Below are all the references for our code:

* https://www.academia.edu/40006464/PERFORMANCE\_GAUGING\_OF\_PORTFOLIO\_LUENBERGER\_DISTANCE\_FUNCTION\_APPROACH\_ON\_SARAJEVO\_STOCK\_EXCHANGE
* https://towardsdatascience.com/efficient-frontier-portfolio-optimisation-in-python-e7844051e7f
* https://www.investopedia.com/terms/s/sharperatio.asp
* http://web.stanford.edu/~wfsharpe/art/sr/sr.htm
* https://en.wikipedia.org/wiki/List\_of\_S%26P\_500\_companies
* <https://finance.yahoo.com/>
* https://stackoverflow.com