IMPLEMENTATION OF DATA SCIENCE TECHNIQUES IN STOCKS PORTFOLIO MANAGEMENT

Thesis

To Fulfil the Requirement to Obtain a Master Degree

Master of Management

Submitted by
Muhammad Rizky Bryan Eddy Noerrahman
19/452508/PEK/25480

To
FACULTY OF ECONOMICS AND BUSINESS
UNIVERSITAS GADJAH MADA
2021

STATEMENT OF AUTHENTICITY OF WRITTEN THESIS

I, the undersigned, state that this written thesis entitled:

IMPLEMENTATION OF DATA SCIENCE TECHNIQUES IN STOCKS PORTFOLIO MANAGEMENT

and submitted for examination on December 25, 2021, is my own work.

I hereby solemnly state that this thesis does not contain, in whole or in part, the work of any other person, in the form of plagiarized or copied phrases or symbols showing the ideas or thoughts of another writer presented as if they were my own; nor does it contain, in part or in whole, writing that I have copied, plagiarized or taken from the writing of another person without recognizing the original author.

In the event that, knowingly or otherwise, I have undertaken the above, I shall withdraw the thesis I have presented as my own work. If, henceforth, it is proven that I copied or plagiarized the work of another person presenting it as if it were my own, the degree and diploma awarded by the university shall be revoked.

Yogyakarta, December 25, 2021
Signed,

Muhammad Rizky Bryan Eddy Noerrahman

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

ABSTRACT

IMPLEMENTATION OF DATA SCIENCE TECHNIQUES IN STOCKS PORTFOLIO MANAGEMENT

Portfolio management can be a formidable task that comprises of portfolio construction, selection and assessment, nonetheless there are tools and techniques in data science that can help with these tasks and in creating optimal portfolios. This quantitative study aims to create optimal portfolios using a set of techniques and methods derived from data science. The study relied on intraday data of KOMPAS100 stocks and market prices along with bond yields for risk-free rate from January 2010 to December 2021.

The study tested two clustering analysis techniques, *k-means* and *agglomerative hierarchical clustering;* to select and group stocks into portfolios. Two portfolio weights optimisation methods were also evaluated, the classical *mean-variance optimisation* (Markowitz, 1952) and a more recent, machine learning based *hierarchical risk parity* (Lopez de Prado, 2016). The portfolios created by these techniques were tested against the market and a set of randomly-selected stocks portfolios.

The results showed that optimal portfolios can be created from the clusters produced the clustering analysis methods combined with portfolio weights optimised by *hierarchical risk parity.* In the validation stage, the optimal clusters from *k-means clustering* and *agglomerative hierarchical clustering* provided holding returns of 63.19% and 58.38%, with risk-adjusted Sharpe of 0.676 and 0.678 respectively. This study is intended to be a reference for portfolio managers, investment managers, individual investors and other researchers in the field of financial management with interest in financial data science.

Keywords: portfolio management, optimal portfolio, financial data science, mean-variance analysis, hierarchical risk parity, clustering analysis.

INTISARI

IMPLEMENTASI TEKNIK DATA SCIENCE DALAM MANAJEMEN PORTOFOLIO SAHAM

Manajemen portofolio bisa menjadi tugas berat yang terdiri dari konstruksi portofolio, seleksi dan penilaian, namun ada teknik dalam ilmu data yang dapat membantu proses ini dan menciptakan portofolio yang optimal. Studi kuantitatif ini bertujuan untuk menciptakan portofolio yang optimal dengan menggunakan seperangkat teknik dan metode yang diturunkan dari ilmu data. Kajian ini mengandalkan data harga harian saham KOMPAS100 dan harga pasar serta imbal hasil obligasi untuk tingkat bebas risiko dari Januari 2010 hingga Desember 2021.

Penelitian ini menguji dua teknik analisis klaster yaitu, *k-means* dan *agglomerative hierarchical clustering*; untuk memilih dan mengelompokkan saham menjadi portofolio. Disamping itu, pengoptimalan bobot portofolio dilakukan menggunakan metode klasikal *mean-variance optimisation* (Markowitz, 1952) dan metode lebih terkini yaitu, *hierarchical risk parity* yang berbasis pembelajaran mesin (Lopez de Prado, 2016). Portofolio yang dihasilkan oleh rangkaian metode ini adalah diuji terhadap pasar dan portofolio dibentuk dari saham yang dipilih secara acak.

Hasil penelitian menunjukkan bahwa portofolio optimal dapat dibuat dengan menggunakan klaster yang dihasilkan oleh metode analisis klaster yang dikombinasikan dengan bobot portofolio yang dioptimalkan menggunakan *hierarchical risk parity*. Pada tahap validasi, klaster optimal dari *k-means* dan *agglomerative hierarchical clustering* memberikan *holding return* sebesar 63,19% dan 58,38%, dengan Sharpe yang disesuaikan dengan risiko masing-masing sebesar 0,676 dan 0,678. Penelitian ini ditujukan untuk menjadi referensi bagi manajer portofolio, manajer investasi, investor individu dan peneliti lain di bidang manajemen keuangan yang tertarik pada ilmu data keuangan. .

Kata Kunci: manajemen portofolio, portofolio optimal, ilmu data keuangan, analisis klaster, analisis mean-variance, hierarchical risk parity

1. PRELIMINARY

1.1. Background

Tremendous amounts of financial data are being produced in every ticking second. From the global markets to everyday individuals, endless streams of financial data are created. Not limited to the values and prices of markets, these data are created from the macro scale of public finance to the micro-scale of personal finance. With the digital shift ahead, there can only be more data to be produced. Akin to finding a new oil reserve, the financial industry is a massive data reserve where new data will always replenish this reserve. Hence, the massive potential that can be tapped into with data science as our tools.

Nowadays, it's hard to imagine fields of study not adopting data science into their rigour. Finance is no different, it was in fact an early adopter of data science starting as early as the 1990s when banks implemented data-driven fraud control and consumer credit decisions (Provost & Fawcett, 2013). Data science techniques and methods have cemented new financial sectors such as fintech and with the recent release of publications such as *The Journal of Financial Data Science* it has established itself as a new field of study, *financial data science*.

One of the most interesting applications of data science in finance is in investment and assets management. Largely due to the fact that the data in this sector are relatively more accessible and the prospect of earning real financial gains if the models or proposed techniques are proven to work. Markets create vast

amounts of information and are not limited to prices and volumes. When we are presented with such large information, our brain will start seeking patterns in the data. Our brain is unique in its superior pattern-recognition ability (Mattson, 2014). Nevertheless, this is also its weakness, our brains are hardwired to seek patterns in meaningless information or simply find patterns simply to confirm our prior beliefs as confirmation bias would suggest (Shermer, 2008). This is the main issue with investors that use their 'gut feeling' when investing, they fall prey to pattern-matching rather than quantitatively analysing these data. To a certain extent this is also a criticism towards the technical analysis approach of investing. This is where data science techniques can help, where investment decisions are driven by data, quantitative statistical analysis and empirical results.

In investment, one of the main topics of discussion is portfolio management. A portfolio is simply a set of assets we want to invest in, yet choosing the right assets with the right proportions for the right investment strategy can be daunting and some investors will simply follow their instinct and succumb to the pattern-matching cycle described earlier. The study proposed some data science techniques such as *clustering, machine learning, data analysis* and *statistical simulation* using the *Monte Carlo method* and evaluated their use for the purpose of portfolio management in creating the optimal portfolios.

1.2. Problem Formulation

The main challenges in creating a portfolio of stocks are selecting the potential candidates of stocks (*portfolio construction*), finding the right proportion or weights of these stocks (*portfolio optimisation*) and assessing the outlook of the portfolio (*portfolio assessment*) (Fabozzi & Pachamanova, 2016). This study addressed these issues and seek to provide an outlook on how they can be made less burdensome with the use of data science techniques.

With the recent advancements such as more robust computational modelling and more effective machine learning methods, these innovations hold tremendous potential for finance. Financial data are also becoming more easily accessible, with stock market data readily available on the internet at any time. Combined with data analysis, valuable information regarding the stocks can be further unveiled.

For the task of selecting the stocks for the optimal portfolios, the study tested two *cluster analysis* methods. Following the stocks selection, the portfolio optimisation frameworks of this research were *mean-variance analysis* of the Modern Portfolio Theory (MPT) proposed by Markowitz (1952) and *hierarchical risk parity* proposed by Lopez de Prado (2016). These methods allow a more statistical approach to stocks portfolio management based on historical data. By exploiting the large amount of stock market data available, portfolios can be constructed, optimised and analysed through these frameworks. For the portfolios created, risk-free rate was accounted for and short-selling was not considered as short-selling is prohibited in Indonesia.

1.3. Research Questions

This research aims to investigate how the proposed techniques can help investors or potential investors in building and managing their stocks portfolios. In detail, this research studies what data-driven solutions can be implemented to help investors in different parts of portfolio management. The research questions can be defined as:

1. What techniques can be used in each stage of portfolio building?
2. How effective are these techniques in each stage of portfolio building?
3. Should investors or potential investors add these techniques to their financial toolbox when investing to create an optimal portfolio?

1.4. Research Objective

The main objective of this study is to create optimal portfolios with the proposed techniques and to evaluate and test the viability of the techniques by comparing the results with randomly selected stocks portfolios.

1.5. Research Contributions

This research aims to contribute to the field by implementing the proposed techniques on the Indonesian stock market. Studies on the implementation of data science techniques have been widely conducted, but the data used in these studies were from foreign stock markets, as of the time of writing there is yet a study to be done on the Indonesian stocks market. This study aims to contribute to the broader

field by implementing these techniques in a different set of data, specifically the Indonesian stocks market. In addition, this study is mainly focused on portfolio management, rather than predicting the financial market which a majority of the studies related to machine learning and data science are concerned with. This study also evaluated the effectiveness of the machine learning and data science techniques in making decisions regarding portfolio management. Lastly, the study aims to introduce a novel combination of readily-available data science techniques or tools incorporated together to address some of the challenges in portfolio management. With that, different techniques were evaluated for different activities or tasks within portfolio management.

1.6. Research Scope

The scope of this research revolves around a set of secondary data collected from two main online sources, Investing.com and Yahoo Finance. The data collected consist of KOMPAS100 market data, KOMPAS100 composite data and Indonesian Government 1-year bond yields in time-series format from 2010-2021. The variables represented by these data are stocks' prices, market prices and risk-free rates, respectively. The research focused solely on capital gains, no data on dividend income was included.

The set of stocks (*stocks universe)* was only limited to stocks that are in KOMPAS100 under the assumption that these stocks are some of the top performers in the IDX market and to reduce the computing resource needed for

bigger data size. Further restrictions on which stocks are included were implemented, the study only accounts for stocks that have consistently been in KOMPAS100 for at least 10 years.

The study should not be considered as a full, generalised investment strategy since the data was limited to a small set of stocks and for a thorough and complete machine learning investment strategy to be fully implemented by a single individual rather than a team of *data curators, features analysts, strategists, backtesters* and *deployment team* is unfeasible (Lopez de Prado, 2018). This study only concerns are portfolio selection, optimisation and assessment with the ultimate goal of creating optimal portfolios.

1.7. Systematic of Writing

The research study is structured around five main chapters consisting of:

CHAPTER 1: INTRODUCTION

This chapter introduces the broader perspective of the research consisting of background, problem formulation, research questions, research objectives and research scope. The intended purpose of this chapter is to provide a sweeping view of the research.

CHAPTER 2: THEORETICAL FRAMEWORK

The second chapter of the research opens a more in-depth perspective of the theories and elements of the research. The chapter consists of a theoretical

framework, detailed explanations of the theories used in the research, exploration of previous studies and the research model.

CHAPTER 3: RESEARCH METHOD

This chapter describes the method and process of the research by explaining the research design, the methods taken for data collection and data cleaning; analytical tools and instruments that are being used. The intention of this chapter is to provide an extensive account of how the research was conducted and how it can be replicated empirically.

CHAPTER 4: FINDINGS AND DISCUSSION

With the data collected and instruments assembled, this chapter presents the results along with their analysis. The aim of this chapter is to discuss the outcome of the research itself.

CHAPTER 5: CONCLUSION

Serving as the closing remarks of the research, this chapter presents the conclusion of the research along with the limitations faced during the research and provides open suggestions for further studies.

2. LITERATURE REVIEW

2.1. Modern Portfolio Theory (MPT)

In 1952, Markowitz introduced a portfolio framework based upon mean-variance analysis and became the foundation of Modern Portfolio Theory (MPT). Markowitz' model relies on two factors, expected returns and volatility represented by variance derived from historical data of the assets. Under the assumptions that investors are risk-averse, they tend to choose portfolios with the lowest variance and for a given expected return. MPT seeks to maximise the expected return for a given level of risk by selecting a portfolio from the *efficient frontier* of the risk-return plot (Markowitz, 1952). This is also called mean-variance optimization (MVO). Figure 2.1 shows an example of an efficient frontier using two stocks sampled from the data.

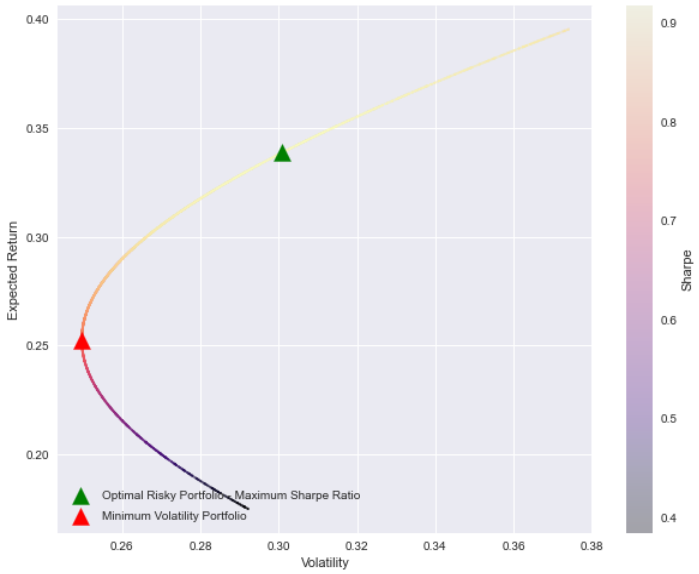

Figure 2.1 An Efficient Frontier of Two Sampled Stocks

2.1.1. Expected Return

Return is the financial reward acquired through investing. An investor would want to maximise their return yet predicting the returns gained from an investment is difficult. The expected return is a statistical estimation of returns that could be gained in the future. The most generally used expected return estimation is the *historical mean expected return* where the expected return is derived from the mean of the historical returns (Markowitz, 1952). Some of the expected return estimates used in this study are *exponentially weighted expected returns, linear regression expected returns* and *CAPM expected return.*

2.1.2. Variance as Risk Estimation

Variance is the mathematical estimation of risk. As with expected returns, future risks are not easily obtained and thus statistical estimation using variance of returns from historical data is a practical approximation (Markowitz, 1952). Nonetheless, variance is not a one-way measure, such that the variability of returns can take on positive and negative values. A high variance (high risk) on an asset suggests that the variability of the returns is large, meaning that it has experienced either large positive or negative returns compared to its mean returns historically. However, variance can be adjusted to be a one-way measure as with semi-variance or downside risk, where variance is only calculated for returns below an expected rate (Roy, 1952).

2.2 Hierarchical Risk Parity (HRP)

While Markowitz's theory was a breakthrough in the field of portfolio construction, it is not without flaws. MPT and mean-variance optimisation relies on expected values and any deviation in the expected values of risk or return will result in a different portfolio such that an equal weighted portfolio could outperform a mean-variance optimised portfolio, implying that the results from MVO may be unstable (Michaud, 1989). HRP attempts to address this by deviating from expected values and looking at the covariance matrix and the hierarchical structure of the assets within the portfolio and by exploiting more recent mathematical methods: graph theory and machine learning. HRP can allocate assets based on a singular covariance matrix by combining assets into a hierarchical structure of clusters by using a *tree graph* or *tree cluster*. A hierarchical structure is analogous as to how stocks in a market can be split into sectors, industry groups and sub-industry, but calculated mathematically. Given the covariance matrix of the assets, HRP then sorts the covariance matrix such that the largest values lie along the diagonal. Using inverse-variance allocation on the covariance matrix, weights allocation is distributed using the tree cluster where the further down the asset is in the tree cluster, the lesser the ratio of it will be allocated (Lopez de Prado, 2016).

First, *tree clustering* is implemented to create a hierarchical structure of clusters to the portfolio such that weights allocation can flow according to the tree graph as shown in Figure 2.2. From the figure, we can see that the sampled stocks

have three separate hierarchy structures, thus it can be determined that for this sample there can be three groups or clusters of stocks.

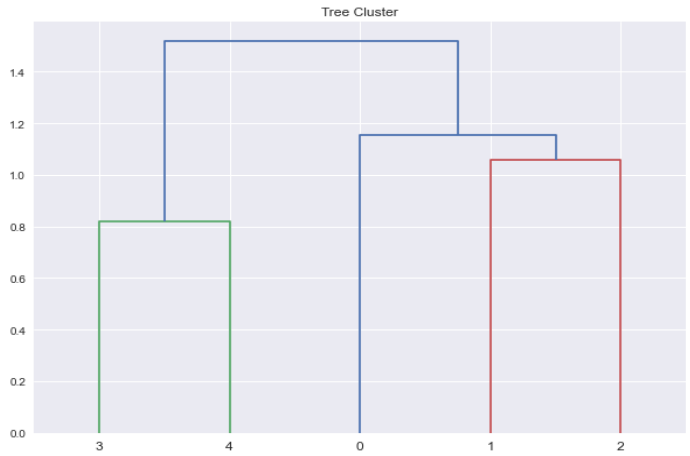

Figure 2.2 Tree Cluster of Sampled Stocks (n=5)

Second, *quasi-diagonalisation* rearranges the covariance matrix such that the largest values lie along the diagonal, therefore similar stocks are closer and dissimilar stocks are further. As demonstrated in Figure 2.3, the covariance matrix of the sampled stocks was rearranged along the main diagonal, creating the said clusters earlier.

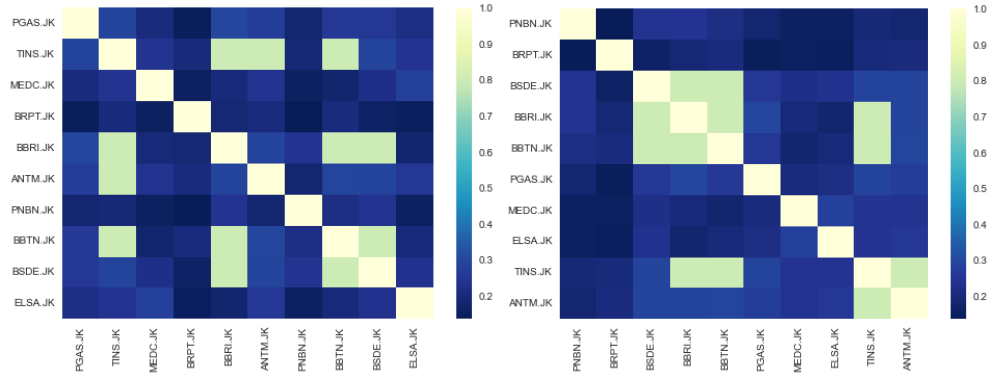

Figure 2.3 Covariance Matrix of Sampled Assets (n=10) before quasi-diagonalisation (left) and after (right)

Lastly, the *recursive bisection* takes the covariance matrix and the tree cluster and applies inverse-variance weights allocation such that stocks closer to the top of the tree cluster have more weight allocation and stocks further down the tree cluster.

2.3. Capital Asset Pricing Model (CAPM)

In 1963, in conjunction with Markowitz, Sharpe extended MPT with a model of portfolio analysis named *diagonal model*, where Sharpe described that any return can be determined by,

$$R_i = A_i + B_i I + C_i. \quad (1)$$

Where $\boldsymbol{R_i}$ is the expected return of the securityA_i, B_i are the parameters for the given security $\boldsymbol{i}$ while $\boldsymbol{C_i}$ represents the random variable of the security with an expected value of zero and variance Q_i . $I$is described as the level of some index, which may represent the whole market index, price index or econometric index such as Gross National Product (Sharpe, 1963). The model above was a specialised case of Capital Asset Pricing Model (CAPM), CAPM extends the previous concept into a general case theory. A generalised CAPM was introduced later, independently in a number of studies among them are Sharpe (1964), Lintner (1965) and Mossin (1966).A general CAPM was formalised as,

$$E[R_i] - r_f = \beta_i(E[R_m] - r_f) \quad (2)$$

where $E[R_i]$is the expected return of the asset, $r_f$is the risk-free rate, $E[R_m]$is the expected return of the market and $\beta_i$is the asset's beta. In CAPM the beta of the asset is defined as,

$$\beta_i = \frac{Cov(R_i, R_m)}{Var(R_m)} \tag{3}$$

where $Cov(R_m, R_i)$ is the covariance of the asset returns and the market returns while $Var(R_m)$ is the variance of the market returns. This is an *ordinary least squares* solution to CAPM beta. Beta in itself is a representation of the asset's non-diversifiable, systematic risk. From this definition of CAPM, statistical estimates of these values can be calculated by running a linear regression between $E[r_i] - r_f$, risk premium and $E[R_m] - r_f$, market premium produces a regression line called *security characteristic line* defined as,

$$E[R_i] - r_f = \alpha_i + \beta_i(E[_m] - r_f) + \epsilon_i \tag{4}$$

where α_i is the asset's alpha, derived from the y-intercept of the regression line, β_i is the asset's beta or the slope of the regression line and ϵ_i are the residuals from the regression representing non-systematic, non-market risk. Figure 2.4 shows the security characteristic line regression of BBCA using data from 2019 derived from its risk premium and market premium data. Given the regression calculated, the asset's alpha and beta can be deduced.

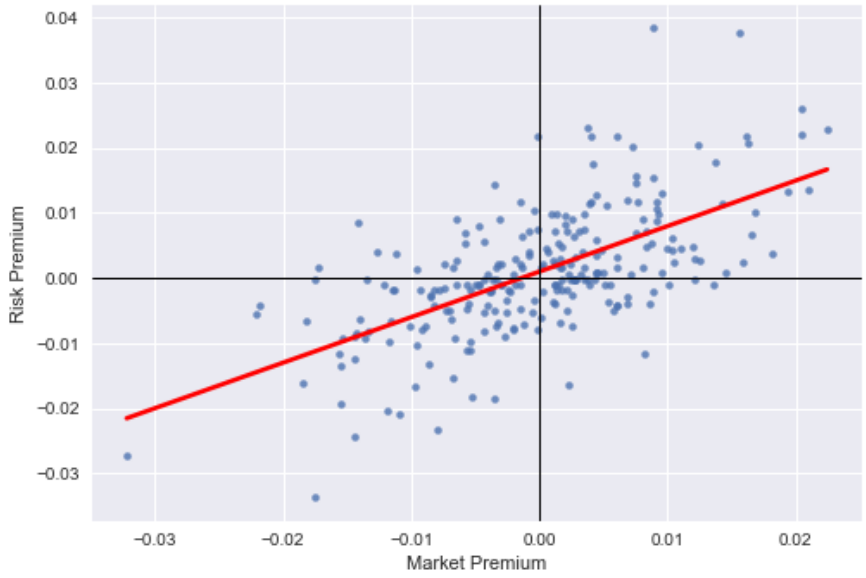

Figure 2.4 An Example of SCL Regression on BBCA Stock Data from 2019

2.4. Data Science Techniques

While the theories above provide perspectives on analysing expected return volatility in optimising portfolio weights, they do not take into consideration how the assets are selected for the portfolios, this is the other problem investigated in this study. The methods and techniques of data science such as the Monte Carlo method, machine learning through cluster analysis that are used in this study are based on statistical methods. Most of these methods were either readily available in the Python libraries used or their implementations were provided in the literatures.

2.4.1. Data Visualisation

While numerical values and results are important, large data can be hard to visualise. For example, a correlation matrix in its numerical form is important for calculations in quantitative finance, but in a business management setting, a visualisation method such as a heatmap may help expressing the insights within this correlation matrix much more visually. In this study, data will be provided with visualisations as much as possible.

2.4.2. Feature Engineering

Features are measurable values derived from existing dataset. Feature engineering is the process of composing new measurable values or metrics from the dataset. Time-series of stocks' prices in itself may not have much interpretable insights, but more features such as price variability, rate of returns, moving averages and movements can be further extracted to provide broader perspective into the stocks.

2.4.3. Principal Component Analysis (PCA)

Principal component analysis is dimensionality reduction or decomposition technique. PCA reduces the multiple variables (*dimensions*) of the data into principal components by projecting the data into a lower dimensional space. PCA compresses the data into components that matter the most and in large, high dimensional datasets such as financial data, PCA allows reduction in redundancy

and noise (Tatsat, Puri & Lookbaugh, 2020). Figure 2.5 shows how every variable of each stock such as it's expected return, volatility, risk-adjusted returns, alpha, beta and other variables can be reduced and projected into a 2-dimensional x-axis and y-axis plot by reducing all of the variables into two principal components.

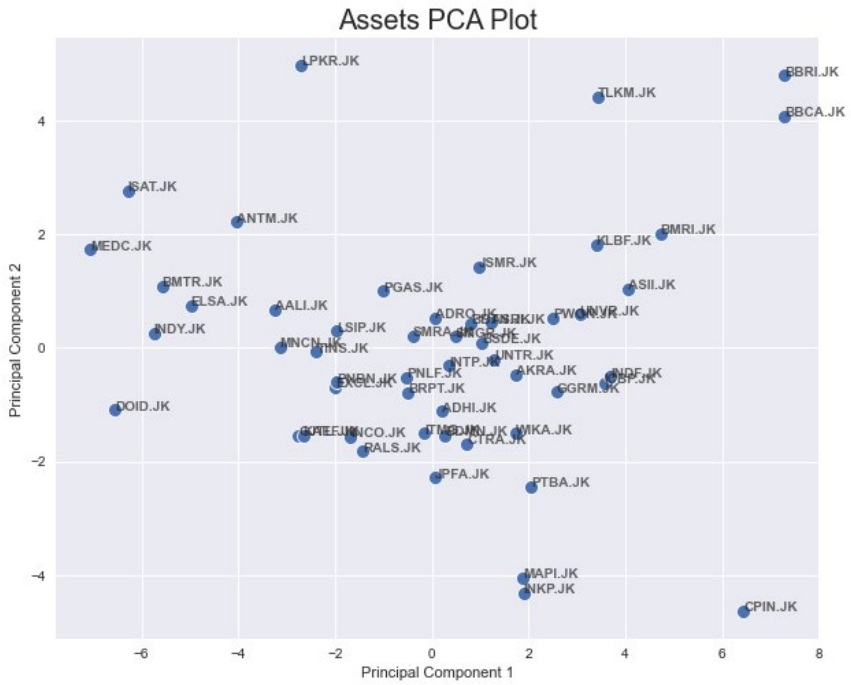

Figure 2.5 Principal Component Analysis Plot on Assets Metrics

2.4.4. Cluster Analysis

Essentially, cluster analysis or clustering is the process of grouping objects or data in a statistically or mathematically rigorous way. The concept of using clustering analysis in asset management has been discussed extensively in the recent work of Lopez de Prado (2020) with some blueprints from Tatsat, Puri & Lookbaugh (2020). Cluster analysis is also considered as an *unsupervised machine*

learning method, meaning that it classifies or groups data without prior labels regarding the classes or groups. There are multiple models and algorithms of cluster analysis and it would be a whole study on its own to test out all the possible cluster analysis models and algorithms.

The study is only testing two models, a *centroid model* specifically *k-means clustering* and a *connectivity model,* hierarchical agglomerative clustering. Since these models are readily available on ***scikit-learn*** library, the cluster analysis of this study relied on the implementations from the library and the detailed description of the algorithms can be found on ***scikit-learn*** user guide. As a centroid model, k-means clustering seeks to group these objects by finding the centres of these clusters by implementing an algorithm that minimises the *inertia* within the cluster, where the *inertia* is the sum-square of the value of each object or data minus the mean of the cluster. These can be expressed in an 'Elbow Curve' plot as shown in Figure 2.6.

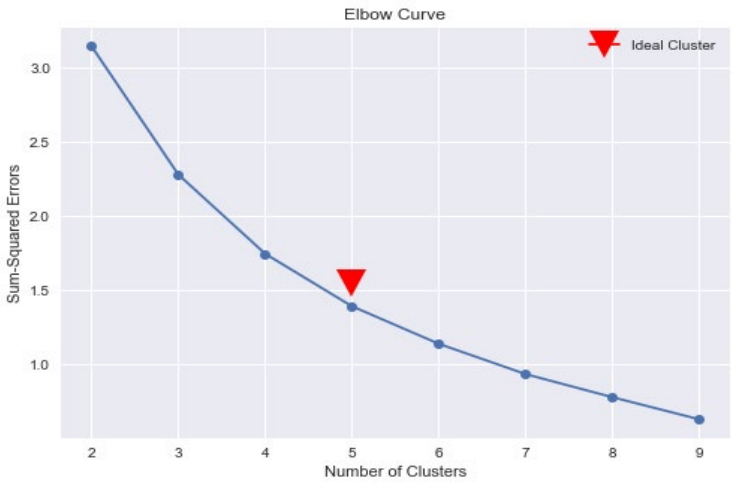

Figure 2.6 An Example of Elbow Curve Plot

In addition, Figure 2.7 shows an example of k-means clustering with expected return and volatility data from 30 sampled stocks as input. With each cluster expressed in different colours, we can see how the algorithm creates boundaries between these clusters and groups stocks with similar characteristics in their own clusters.

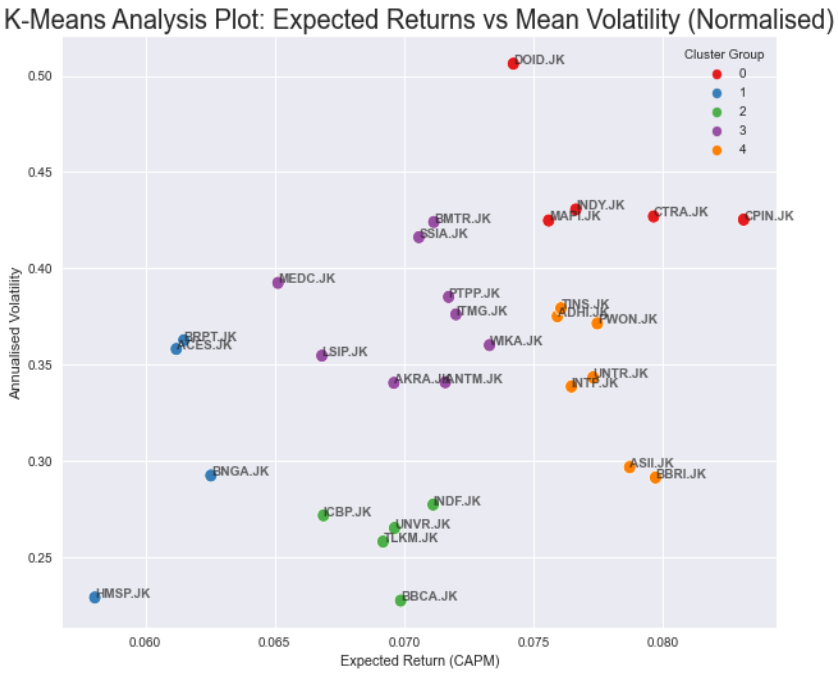

Figure 2.7 An Example of k-means Clustering from Sampled (n=30) on Normalised Expected Return and Annualised Volatility

On the other hand, hierarchical clustering attempts to find a hierarchical structure within the data and groups the data according to their hierarchy derived from the data's distance matrix. Hierarchical structures can be visualised through a dendrogram, as shown in Figure 2.8. From the same figure, it can be deduced that from the 10 sampled stocks, there are two distinct *roots* to the tree structure. Each

node in these *roots* is a stock from the samples and where these *roots* gather signifies a cluster. The number of clusters can be determined by 'cutting' the structure at a certain level. As hierarchical agglomerative clustering takes distance matrix as an input, Figure 2.9 shows how a distance matrix transformed when hierarchical agglomerative clustering was applied. Along the main diagonal of the distance matrix, we can see different stocks are grouped up or clustered together.

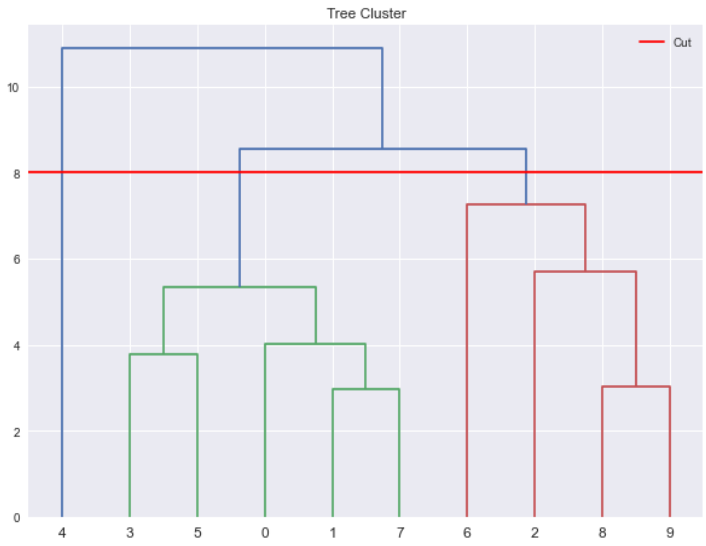

Figure 2.8 Tree Cluster Dendrogram of Sampled Stocks (n=10)

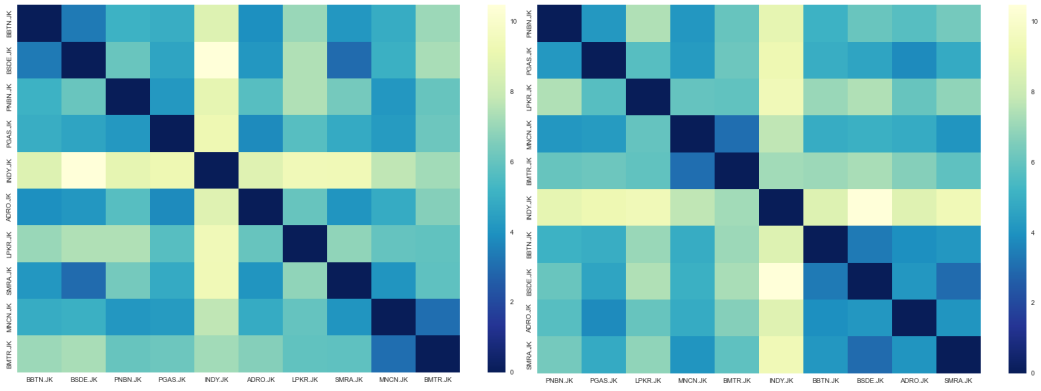

Figure 2.9 Distance Matrix Before and After Hierarchical Agglomerative Clustering

With both of these clustering methods, the study proposes that the clusters of stocks produced were to be considered as portfolios, such that the stocks in a cluster belongs in a single portfolio. This is meant to be a solution to the assets selection problem when constructing a portfolio.

2.4.5. Monte Carlo Method

Monte Carlo method or Monte Carlo simulation is a computational method based on randomness. Major literatures used in this study all agree that Monte Carlo method is an indispensable tool of financial data science (Hilpisch 2020; Lopez de Prado 2018, 2020) Given a set of possible inputs, the Monte Carlo method finds the answer by randomly generating these possible inputs. Essentially, the Monte Carlo method is a series of random experimentations. In this study, to find the optimal portfolio in a mean-variance optimization, a large number of portfolios with random weights were simulated. Monte Carlo method seeks to find an approximate solution based on multiple random sampling. Figure 2.10 shows how Monte Carlo method was implemented in finding the efficient frontier and thus the optimal portfolio.

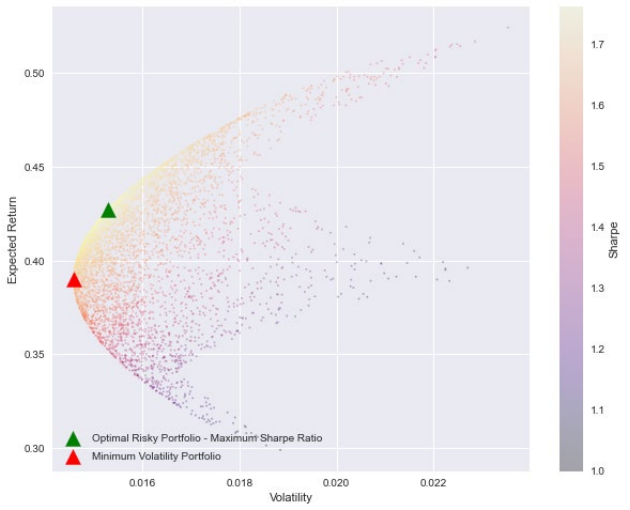

Figure 2.10 Monte Carlo Portfolio Optimisation Example on Sampled (n=5) Assets with 5000

2.5. Previous Studies

The implementations of data science techniques in the financial field and on the subject of investment are widely studied. With machine learning for forecasting and trading being some of the highly discussed topics, exemplified by studies such as machine learning for hedging and derivative pricing (De Spiegeleer et. al., 2018) , volatility forecasting through machine learning (Zhai, Cao & Liu, 2020) and stocks selection based on machine learning forecasting (Rasekhschaffe & Jones, 2019). Even in the field of computer science, finance has become a topic within the study of machine learning (Siami-Namini, Tavakoli & Namini, 2018; Karim et. al., 2019).

However, for this study the focus was less on forecasting and more on data-driven portfolio management. There were significantly less studies on this topic compared to the topics stated earlier. The techniques and methods utilised in this study were accessible and readily implementable from the literatures (Hilpisch,

2020; Lewinson, 2020; Lopez de Prado, 2018, 2020; McKinney, 2017; Tatsat, Puri & Lookbaugh, 2020) or available as functions from the Python libraries used. The novel approach of the study will be the combination of these proposed techniques. Below is a non-exhaustible list of previous studies on some parts of the methods and techniques implemented in this study.

Table 2.1 Previous Studies.

Authors	Title	Summary
León, D., Aragón, A., Sandoval, J. et al. (2017)	Clustering algorithms for Risk-adjusted Portfolio Construction	Constructed portfolios from Russell 1000 Index. Hierarchical clustering algorithm with Ward linkage produced a portfolio with the best performance.
Raffinot, T. (2018)	Hierarchical Clustering-Based Asset Allocation	Implemented Hierarchical Risk Parity (HRP) with different linkage methods and compared the performance with equal-weighted, minimum variance and most diversified portfolio. For portfolios consisting of individual stocks, HRP implementations produced significantly higher returns.
Nanda, S.R., Mahanty, B., Tiwari, M.K. (2010)	Clustering Indian stock market data for portfolio management	Presented that clustering algorithms in conjunction with mean-variance optimisation can create diversified portfolios.
Silva, B., Marques, N. (2010)	Feature Clustering with Self-organizing Maps and an Application to Financial Time-series for Portfolio Selection	Implemented feature clustering on financial time-series to create portfolios based on the clusters produced.

2.6. Research Framework

The study was fully-implemented in the Python language using the Jupyter Notebook. The framework can be divided into two main units of tasks, pre-processing and modelling.

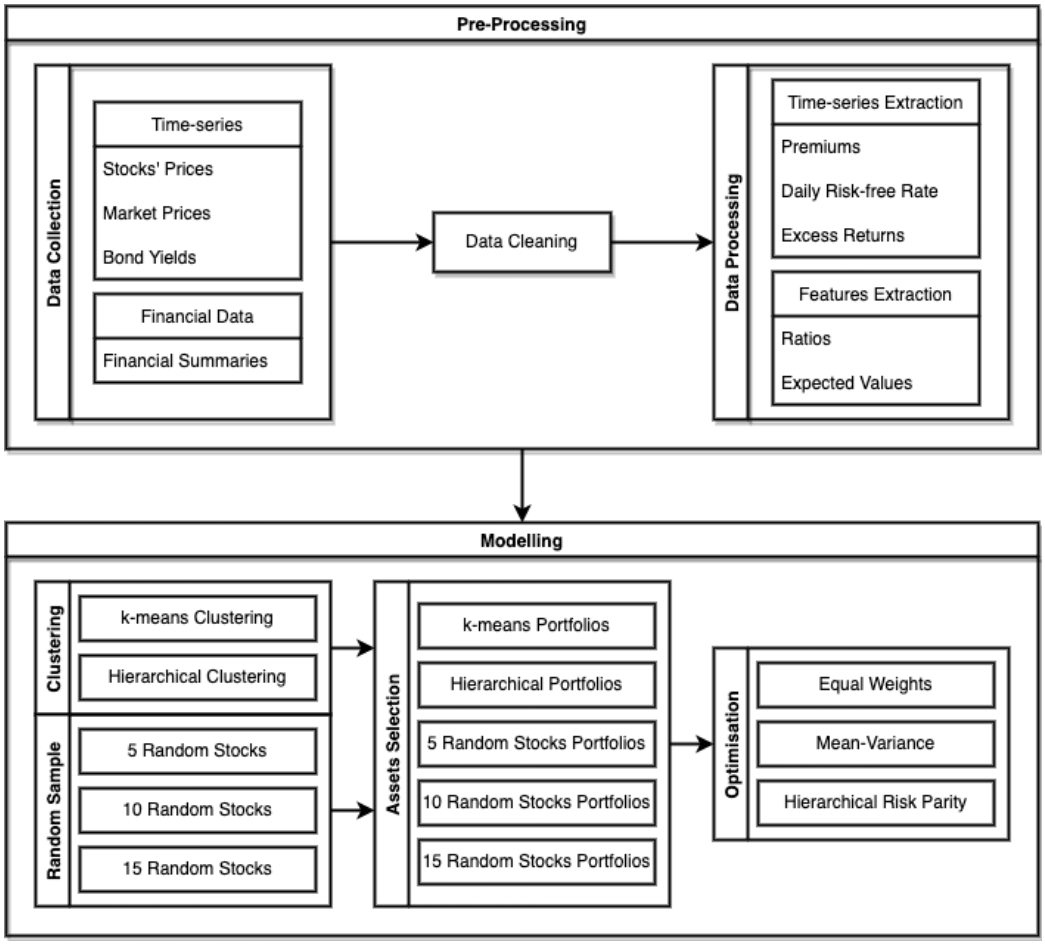

Figure 2.11 Research Framework.

3. RESEARCH METHOD

3.1. Research Design

The research is limited to the set of methods and techniques proposed. While there are numerous asset management strategies based on data science and machine learning, given the limited computational resources and time, only a handful were selected. Financial data from the companies, external factors, other portfolio theories and other classes or types of assets are not to be taken into consideration in the study. Through restricting the study on just time-series data of the stocks, market and bond yields, allowed the research to expand on the data analysis methods used to gather more valuable information from the given limited data and create a controlled setting where the proposed models and techniques can be tested.

In portfolio construction, the task of focus was choosing a group of *n* stocks from a larger set of *N* stocks from the chosen KOMPAS100 stocks. The methods implemented were *clustering analysis* such as *k-means clustering* and application of *Principal Component Analysis* (PCA).

The portfolio optimization methods were based on Markowitz' efficient set (Markowitz, 1952) and Lopez de Prado's *hierarchical risk parity* (HRP) (Lopez de Prado, 2016). For Markowitz' model, the distribution of the weights of the stocks in the portfolio and finding the optimal risky and minimum risk portfolios, Monte Carlo method was implemented, based on the approach used by Hilpisch (2020). The Python implementation of HRP was laid out in detail (Lopez de Prado, 2018),

thus it only needed to be added to the model. To validate the results provided, backtesting was implemented and random assets portfolios were constructed to compare whether the techniques could perform better than naive asset allocation. This concludes the portfolio assessment phase.

3.2. Data and Sample

All the computations and simulations for the study were implemented in Python programming language with Jupyter Notebook for the programming environment, along with its statistical and machine learning libraries such as ***pandas, sci-kit, sklearn, numpy, seaborn, yfinance*** and ***statsmodels*** to name a few. The first stage of this study is the data collection. For the stocks in the KOMPAS100 index the data was collected from Yahoo Finance through ***yfinance*** Python library from the time period of January 1, 2010 to December 13, 2021.

The list of stocks in the index KOMPAS100 is taken from Kontan.com. The data collected consisted of daily 'Open', 'Close', 'Adjusted Close', 'High' and 'Low' prices of the stocks. The study mainly focused on the Adjusted Close price data. In calculating metrics such as Sharpe ratio, beta and alpha, market returns (KOMPAS100) data and bond yields (1-Year Indonesian Government Bond Yield) were required. These two data are collected from Investing.com through ***investpy*** Python library instead, as Yahoo Finance did not have these data. As with the stocks, the data that were studied from these two sources are the Close price data.

While the stocks data collected were up-to-date with the most current KOMPAS100 listed stocks since 2010 there are companies that were only listed in the IDX as recently as 2019. With stocks such as IPTV (PT MNC Vision Networks Tbk) that have only been listed on IDX as early as June 2019 being in the KOMPAS100, this skews the distribution of the data by a lot by only having 563 days in the time series data while other stocks have more than 2800 days. To mitigate the issues from having time series data of different lengths, only stocks that have been in IDX for a certain consecutive period of time were chosen. This was done by only keeping stocks that have more than the median number of days in their time-series data. This resulted in only 52 stocks in the study.

The financial summaries from ***investpy*** were only able to provide data from 2017 onwards, hence only market capitalization, revenue and income values were taken from these summaries.

Table 3.1 Data Collection Summary

Name	Description	Source	Method	Data Type	Period
KOMPAS100 Stocks List	List of stocks in KOMPAS100 index.	Kontan.co.id	pandas	List	4 August 2021
KOMPAS100 Stocks Data	Intraday values of open, close, high, low, adjusted close prices and volumes of each stock currently in the KOMPAS100 index.	Yahoo Finance	yfinance	Intraday time series	1 January 2010 to 31 October 2021
KOMPAS100 Composite Data	Intraday values of open, close, high, low, adjusted close prices and volumes KOMPAS100 index composite.	Investing.com	investpy	Intraday time series	1 January 2010 to 31 October 2021
Indonesia Government 1-Year Bond Yield	Intraday values of open, close, high, low and adjusted close of Indonesian Government 1-year bond yields.	Investing.com	investpy	Intraday time series	1 January 2010 to 31 October 2021
Financial Summaries of KOMPAS100 Companies	Summary consisting of Revenue, EPS, Market Cap, P/E Ratio, Beta and Shares Outstanding.	Investing.com	investpy	Annual cross-sectional data	2017 to 2020

3.3. Variables and Measurements

The study was implemented in Python and a number of libraries to support data wrangling, statistical analysis, mathematical modelling or simulation, feature engineering, machine learning and data visualisation. Python in itself is a general-purpose programming language, therefore it is not a field-specific programming language such as MATLAB or R, with specific focus on matrix manipulation and

statistical computing, respectively. Python can be specified to support specific tasks with the help of its extensive libraries. Table 3.2 shows the libraries that were used for this study and their purpose.

Table 3.2 Python Programming Libraries Used in the Study

Name	Developers	Library Documentation	Purpose
NumPy	Harris, Millman, van der Walt et al. (2020)	https://numpy.org/doc/stable/	Numerical analysis, matrix manipulation.
pandas	McKinney (2010) Pandas Dev. Team (2020)	https://pandas.pydata.org/docs/	Data manipulation, data analysis, matrix calculations.
Matplotlib	Hunter (2007)	https://matplotlib.org/stable/	Plotting.
seaborn	Waskom (2021)	https://seaborn.pydata.org/	Data visualisation.
statsmodels	Seabold & Perktold (2010)	https://www.statsmodels.org/stable/	Statistical modelling, statistical tests.
scikit-learn	Pedregosa, Varoquaux, Gramfort et al. (2011)	https://scikit-learn.org/stable/	Machine learning.
scipy	Virtanen, Gommers, Oliphant et al. (2020)	https://scipy.github.io/devdocs/	Statistical analysis.
investpy	del Canto (2021)	https://investpy.readthedocs.io/	Retrieving data from Investing.com.
yfinance	Aroussi (2021)	https://github.com/ranaroussi/yfinance	Retrieving data from Yahoo Finance.

All the data stated in Table 3.1, calculated statistics, metrics (e.g., returns, Sharpe ratio) and the results are collected as pandas DataFrames which allow extensive data manipulation through its large libraries of functions. DataFrames are also cross-compatible with all the other libraries listed in Table 3.2, making the

collected data much more flexible in terms of analysing, visualising and manipulating. For data analysis, the main reference point is the work of McKinney (2017) as the author is also the developer of the ***pandas*** library.

3.3.1. Return and Rate of Return

For most of this study, the returns and rate of returns (e.g., daily, monthly, annual) for stocks or market rates will be *logarithmic returns* and *logarithmic rate of returns* unless stated otherwise. Logarithmic return or *log return* is defined as,

$$R_{log} = ln\frac{v_{t+1}}{v_t} \tag{2}$$

where $\boldsymbol{v}$ is the value or price of the stocks or the market and t signifies the time, where $t+1$ is one period (e.g., day, month, year) after t. Continuing on, the logarithmic rate of return or *log rate of return* is defined as,

$$r_{log} = \frac{ln\frac{v_{t+1}}{v_t}}{T} \tag{3}$$

where $\boldsymbol{T}$ is the length of the time period. Given the two equations above, logarithmic rate of return can be also be defined as log return R_{log} divided by T,

$$r_{log} = \frac{R_{log}}{T}. \tag{4}$$

Using log-returns and log rate of return has a few advantages. First, returns are continuously compounded and become time-additive such that for a sequence of *n* returns, the compound returns become,

$$R_{log} = R_{log,1} + R_{log,2} + \cdots + R_{log,n}. \tag{5}$$

This allows much easier calculation of cumulative returns for different periods of time as the returns for the given period only needed to be summed up.

Second, it allows the returns to be normally distributed and allows classic statistical analysis that presumes normal distribution. Lastly, the annualization of rate of return becomes much more simplified. By using Equation 4 for 252 trading days in a year where T is $\frac{1}{252}$ the annualised rate of return is,

$$\bar{r}_{Annualised} = \bar{R}_{log} \times 252. \tag{6}$$

3.3.2. Correlation Coefficients of Stocks

With the returns data calculated, further statistics and calculations can be derived. To gain insight into the correlation between the stocks, *Pearson correlation coefficients* between the stocks were derived from the returns data. This was done with the help of ***pandas*** library by implementing the `pandas.DataFrame.corr()` function on the stocks returns DataFrame.

Combining the result with ***seaborn*** gave a heatmap visualisation of the correlation matrix.

3.3.3. Variance and Standard Deviation of Returns

The variances and standard deviations of the stocks' returns were calculated in a similar manner to the correlation coefficients. Functions from ***pandas*** library were used to calculate these statistics from the returns. For variance, `pandas.DataFrame.var()` was used and for standard deviation, `pandas.DataFrame.std()` function was used.

For this study, two types of variances and standard deviation were used, regular variance and standard deviation along with semi-variance and semi-standard deviation. The difference between the regular statistics and the semivariance (or semi-standard deviation) is that for the latter, only values below certain levels were calculated, in this case returns below zero. The functions become `pandas.DataFrame[DataFrame<0].var()` for semivariance and `pandas.DataFrame[DataFrame<0].std()` for semi-standard deviation. Analogous to how variance and standard deviation represent two-way risks (upside and downside), semi-variance and semi-standard deviation were used to represent *only* the downside risk or the risk of loss, hence the returns that were considered were only negative returns or loss. Variances and standard deviations were used as representations of volatility and risk in the study, in accordance with MPT.

3.3.4. Covariance of Stocks and Market Prices

As with variance, using ***pandas*** covariance can be calculated using a function provided in the library, `pandas.DataFrame.cov()` and implemented on the desired DataFrame. Covariance was used to calculate theoretical Beta based on the CAPM model.

3.3.5. Security Characteristic Line (SCL) Regression

Linear regressions based on SCL were used to estimate regression Beta and Alpha of the stocks. Given that SCL is defined as,

$$r_{i,t} - r_f = \alpha_i + \beta(r_{m,t} - r_f) + \epsilon_i \tag{7}$$

where $r_{i,t}$ - r_f and $r_{m,t}$ - r_f are *risk premium* and *market premium* respectively. This interpretation of SCL allowed linear regressions to be carried out on each of the stocks, with *market premiums* as the regressors and *risk premiums* as regressands. Following these regressions, the regression Beta and Jensen's Alpha of each stock were derived. Linear regressions were implemented using ***scipy*** library and its `scipy.stats.linregress()` function.

3.3.6. Risk-adjusted Return Ratios

Three risk-adjusted returns ratios were calculated to assess the performance of the portfolios and the stocks, consisting of Sharpe ratio, Sortino ratio and

Information ratio. For Sharpe ratio and Sortino ratio, the before holding period or *expected* (ex-ante) and after holding period or *realised* (ex-post) values were calculated. The Sharpe and Sortino ratios are calculated based on the risk premiums of the stocks, where Sharpe is defined as:

$$Sharpe = \frac{E[R_i] - r_f}{\sigma_{if}} \tag{8}$$

Where $E[R_i]$ is the expected return of the portfolio or stock, r_f is the risk-free rate and σ_{if} is the standard deviation of the risk premium. Similarly, Sortino ratio takes

$$Sortino = \frac{E[R_i] - r_f}{DR} \tag{9}$$

$E[R_i]$ and r_f but replaces σ_{if} with downside risk, *DR*. Downside risk is the variability of the stocks returns below a required rate of return, in this study this required rate was set as zero, thus the downside risk measured the risk of loss. On the other hand, the Information ratio adjusts the expected return with the expected return of a benchmark, in this case the expected return of index composite of KOMPAS100.

$$Information\ Ratio = \frac{E[R_i] - E[R_m]}{\sigma_i} \tag{10}$$

3.3.7. Principal Component Analysis (PCA)

PCA uses the same library as *k-means* clustering, with the function `sklearn.decomposition.PCA` and specifying the number of components `n_components=2`.

3.3.8 Distance Matrix

Hierarchical clustering takes *pairwise distances* as its input metrics. Analogous to a correlation matrix, distance matrix describes the similarity and dissimilarity of a dataset in terms of distances of how close or further apart two samples are based on their variables. Distance matrix is a square matrix with zeros on the main diagonal as the distance between the same sample should be zero. Distance matrix is required for the agglomerative clustering. To transform the dataset into a distance matrix, ***scikit-learn*** library was used with its `sklearn.metrics.pairwise_distances()` functions.

3.3.9. k-means Clustering

The *k-means* clustering algorithm was implemented for assets selection stage after features were calculated. In essence, as an *unsupervised machine learning algorithm* the *k-means* clustering grouped the stocks into clusters with similar features by finding centroids of each cluster. The implementation came in two stages, with finding the ideal number of clusters first and then fitting the *k-means* clustering afterwards.

Here, ***scikit-learn*** library was utilised as it has an extensive library of machine learning functions. To find the ideal number of clusters, an *Elbow Curve* plot for the given set of data was looked at. The plot was produced by fitting the `sklearn.cluster.MiniBatchKMeans.fit()` function over the data and a set of numbers of clusters. The clustering function *mini-batch k-means* is essentially the same as normal *k-means*; the only difference is that it uses small randomly sampled batches from the dataset instead of the full dataset which is meant to increase the computational speed. With the number of clusters deduced, using the same function as above but with a specified number of clusters to be fitted to the dataset *n*, to group the data into clusters using `sklearn.cluster.MiniBatchKMeans(n_cluster = n).fit()`. The resulting clusters were treated as independent portfolios of grouped stocks from the clusters.

3.3.10. Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering tries to create clusters from data by analysing its hierarchical structures such that a cluster is dissimilar from another cluster and each of the elements within the cluster belong in the same hierarchical group. This clustering is also a type of *unsupervised machine learning* method. The input to this clustering method is the distance matrix of the dataset. To implement *hierarchical agglomerative clustering*, ***scikit-learn*** was used and its `sklearn.cluster.AgglomerativeClustering(n_cluster = n).fit()`

was implemented. The default *linkage* for this clustering method in ***scikit-learn*** is the *Ward's method*, the same *linkage* used by León, Aragón, Sandoval et al. (2017). The clusters produced by the *hierarchical agglomerative clustering* were treated as portfolios consisting of stocks within those clusters. From here on, the term *hierarchical agglomerative clustering* and *agglomerative clustering* are used interchangeably throughout the study to express this method.

3.4. Data Analysis Method

3.4.1. Feature Engineering

Raw data such as prices on their own do not provide insightful information regarding the stocks or the market. Features such as rate of return, variance, Sharpe ratio and Beta are more relevant and significant in portfolio management as they provide insights on the performance of the stocks, the risk that they carry and how potentially profitable they are with regards to their risks. In this part of the data analysis stage, feature engineering was done in two stages, where the first stage consisted of calculating basic features such as log returns, variances, standard deviations and covariances. The next stage of feature engineering is creating more data from the previously calculated features to create new set of features such as Beta, Alpha, Sharpe ratios and Sortino ratios. The Python implementations for most of the calculations of these features are available in the works of Hilpisch (2020), Lopez de Prado (2020) and Lewinson (2020).

Table 3.3 Extracted Features

Features	Description
Annualised Return	Average log-returns x 252 trading days.
Annualised Volatility	The standard deviation of log returns x $\sqrt{252}$.
Annualised Downside Risk	The standard deviation of log-returns where log-returns < 0 x $\sqrt{252}$.
Exponential Weighted Mean (Exponential Weighted Expected Return)	Exponentially weighted mean of log returns with rolling 300 days window.
Exponentially Weighted Volatility (Exponential Weighted Expected Volatility)	Exponentially weighted standard deviation of log returns with rolling 300 days window.
Expected Return (LinReg)	Expected return from linear regression.
Expected Return (CAPM)	Expected return from CAPM calculation.
Alpha	y-intercept of linear regression between market premium and risk premium.
Beta	Slope of linear regression between market premium and risk premium.
Beta (CAPM)	Covariance of stocks returns and market returns divided by market variance.
Sharpe	Expected returns deducted by risk-free rate divided by volatility.
Sortino	Expected returns deducted by risk-free rate divided by downside risk.
Information Ratio	Expected returns deducted by expected market return divided by volatility.
Last 3 Months Change	Change of price within the last three months.
Last 6 Months Change	Change of price within the last six months.
Skew	The skewness of the returns distribution.
Kurtosis	The kurtosis of the returns distribution.
Mean Revenue Growth	Mean of revenue growth from 2017-2020
Mean Income Growth	Mean of income growth from 2017-2020

3.4.2. Portfolio Construction

In the portfolio construction process, the study will look at the implementation of clustering algorithms in selecting stocks. The use of k-means clustering in portfolio management has been studied (Nanda, Mahanty & Tiwari, 2016). Using hierarchical clustering in stocks selections have been studied by León, Aragón, Sandoval et al. in 2017. All of these clustering algorithms are readily available on scikit-learn libraries. To compare how these portfolio construction methods perform, baseline equal-weighted portfolios were calculated where each of the stocks in the portfolio has a weight of *1/n* where *n* is the number of stocks in the given portfolio.

3.4.3. Mean-Variance Portfolio Optimisation

The concept for this implementation is an extension of the examples provided from two works of literature that outline implementations of machine learning and data science on financial data (Hilpisch, 2020; McKinney, 2020). Here, the Monte Carlo method is used to find the *efficient frontier* of the portfolio by creating a large number of randomly weighted portfolios *P*. The randomly generated weights follow,

$$w_1 + w_2 + \cdots + w_n = 1 \tag{8}$$

where n is the number of stocks in the portfolio and w_i are random numbers between 0 and 1. Each portfolio was given a unique set of weights in the form of a weight matrix W_p where $p \in P$.

$$W_p = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{9}$$

The expected returns and the variance (*volatility*) of the portfolio were calculated according to the given random weights of the portfolio. Expected return of the portfolio is defined as,

$$E[R_p] = w_1 E[R_1] + w_2 E[R_2] + \cdots + w_n E[R_n] \tag{10}$$

while variance is defined as,

$$V_p = W_p \cdot cov[P, P] \cdot W_p^T \tag{11}$$

where $E[R_P]$ is the expected rate of return on the portfolio, V_P is the volatility of the portfolio, $cov[P,P]$ is the covariance matrix of the stocks in the portfolio and W_p is the vector of weights of the portfolio. These values were calculated during

the Monte Carlo simulation. For the study, 50,000 randomly weighted portfolios were iterated for each simulation.

Once the simulation is completed, to find the *minimum risk portfolio* and the *optimal risky portfolio* can be obtained by finding the portfolio with the minimum risk factor and the maximum Sharpe ratio respectively. For this study, the portfolio considered was the *optimal risky portfolio* also known as *tangency portfolio.* This portfolio is meant to give the optimal risk-adjusted return.

3.4.4. Hierarchical Parity Risk (HRP) Portfolio

Lopez de Prado introduced HRP in his 2016 paper (Lopez de Prado, 2016) and provided a full Python implementation of HRP in his book (Lopez de Prado, 2018). Thus, in this study, Lopez de Prado's algorithm of HRP was used. The algorithm takes a single covariance matrix of the portfolio and the weights are calculated in three stages described in the previous chapter.

3.4.5. Portfolio Assessment

Portfolio assessment is the final validation step in the study. Here, the performances of the portfolios created through the previous methods were assessed. Using one of the most common approaches in portfolio assessment, *backtesting*. In backtesting, the performance of the portfolios was tested on data from January 1 2020 to 13 December 2021. The holding period is set to two years since some preliminary testing showed that a period of one year skewed the clustering

portfolios towards a very small group of stocks that have abnormal returns for that period and period longer than two years would require larger datasets. With the given weights, the daily returns, holding period returns and log cumulative returns for the stated period were calculated. The daily returns of these portfolios were used in estimating their returns probability densities and Value-at-Risk. The results of these portfolios were compared with each other and the performance of the KOMPAS100 index. The study periods are shown as below:

Table 3.4 Study Periods and Datasets

Stage	Training Data	Testing Data
Model Construction	1 January 2010 to 31 December 2018	1 January 2019 to 31 December 2020
Model Validation	1 January 2010 to 31 December 2019	1 January 2020 to 13 December 2021

3.5. Case Profile

This study is implemented on KOMPAS100 stocks data from 2010-2021. Since some of the stocks currently in KOMPAS100 were only listed in the market as early as 2019, only stocks that have been listed in IDX for more than the median number of trading days of about 2750 days were studied.

The study combined a set of techniques for different stages in portfolio management. Three different asset allocation methods were implemented.

4. FINDINGS AND DISCUSSION

4.1. Data Descriptions

4.1.1. Stocks Selection

Stocks are constantly being listed into and dropped from the larger IDX market, as mentioned earlier with stocks such as IPTV being in the current KOMPAS100 but has only been listed in IDX since July 2019. If all stocks from the current list of KOMPAS100 were to be included in the study, there would be a large skew in the data size, with some stocks having more than 10 years' worth of intraday prices and some such as IPTV that only have just about three years' worth of data. This created an issue when the data were divided further into training and testing sets shown in Figure 4.1. Stocks that have only been in IDX for three years would only have one year worth of training data and two years' worth of testing data. To impose some uniformity to the data size, only stocks which have been in IDX for the median number of trading days of the current stocks in KOMPAS100 or above were selected. This reduced the number of stocks from 100 to 52 stocks listed in Appendix 1.

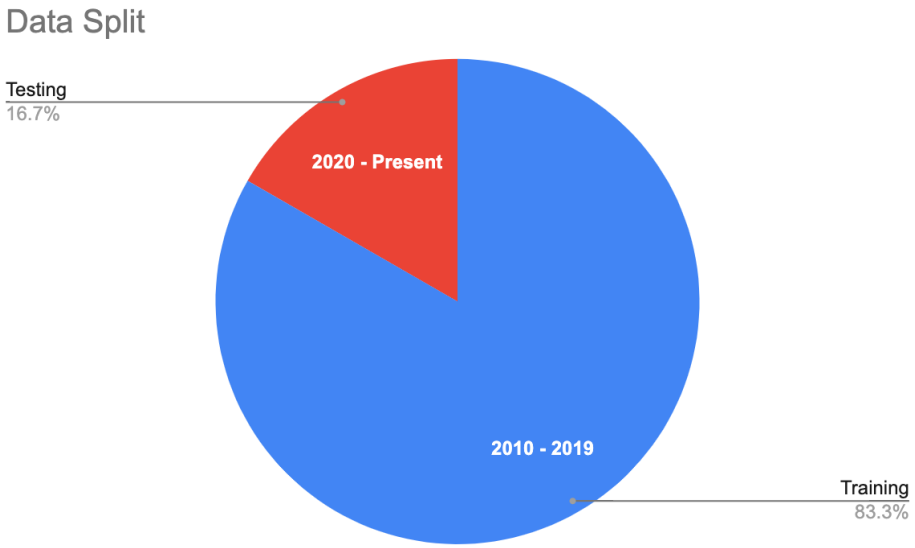

Figure 4.1 Train-Test Split on Data

4.1.2. Raw Prices Data

Data procured from Yahoo Finance and Investing.com were raw prices data and volume data. Table 4.1 shows the average statistics of the stocks data, while Figure 4.1 shows the changes of the adjusted close prices throughout the years. Similarly, Table 4.2 and 4.3 show the average statistics of the KOMPAS100 composite and Indonesian Government 1-year Bond Yields respectively. The trends of the composite and the bond yields are shown in Figure 4.2 and 4.3.

Table 4.1 Average Statistics of Stocks Data

	Adj Close	Close	High	Low	Open	Volume
count	2929	2929	2929	2929	2929	2929
mean	4060.19	4932.68	5004.33	4862.67	4936.06	30655870
std	761.14	686.10	688.02	683.38	686.28	15672646
min	1312.52	1335.00	1335.00	1335.00	1335.00	0
25%	3552.70	4512.68	4594.72	4442.67	4511.40	21929302
50%	4013.60	4980.50	5057.06	4916.69	4986.38	27293801
75%	4517.61	5360.98	5423.01	5291.40	5360.41	34910517
max	6124.31	6838.59	6838.59	6838.59	6838.59	187898445

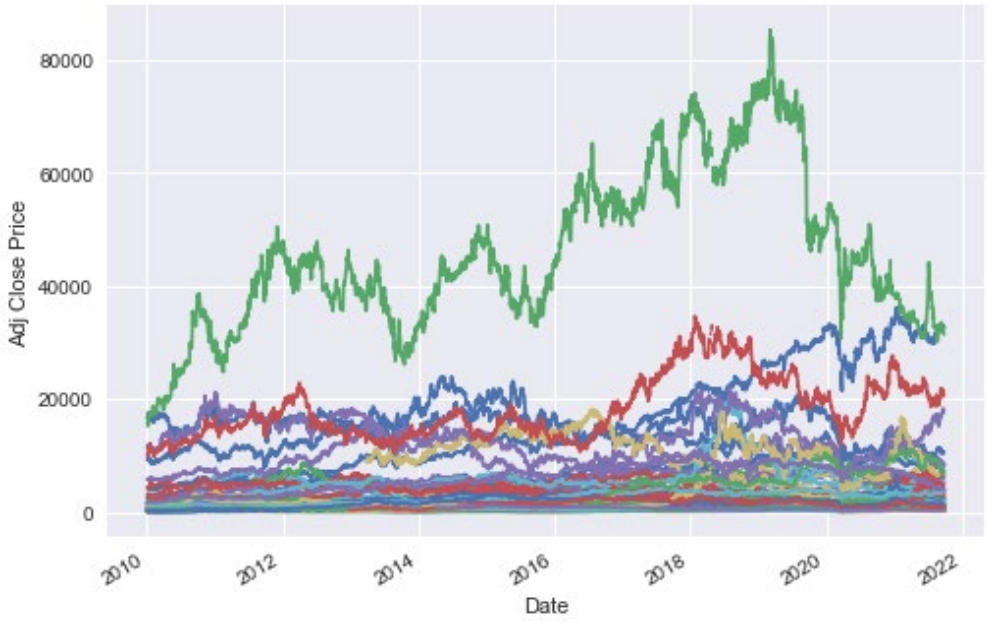

Figure 4.2 Adjusted Close prices of stocks in KOMPAS100 index from 2010 to 2021

Table 4.2 KOMPAS100 Data Statistics from 2010 to 2021

	Open	High	Low	Close
count	2869	2869	2869	2869
mean	1049.76	1056.30	1042.11	1049.39
std	173.59	173.12	173.45	172.95
min	591.86	595.92	580.38	592.09
25%	919.14	924.93	912.04	918.75
50%	1065.28	1071.50	1057.73	1064.68
75%	1192.49	1200.18	1186.20	1192.68
max	1420.94	1421.95	1406.81	1421.50

Figure 4.3 KOMPAS100 prices from 2010 to 2021

Table 4.3 Indonesian Government 1-Year Bond Yield statistics from 2010 to 2021

	Open	High	Low	Close
count	2852	2852	2852	2852
mean	0.0596	0.0601	0.0591	0.0596
std	0.0123	0.0123	0.0121	0.0122
min	0.000000	0.0240	0.000000	0.0240
25%	0.0506	0.0512	0.0503	0.0510
50%	0.0621	0.0625	0.0618	0.0622
75%	0.0685	0.0691	0.0677	0.0682
max	0.0904	0.0904	0.0888	0.0888

Figure 4.4 Indonesian Government 1-Year Bond Yield from 2010 to 2021

4.2. Data Processing and Model Construction

4.2.1 Model Construction

As described earlier in Figure 4.1, the raw data were split into testing data and training data. In building the model, data from the training split were used and data from the testing split were kept. This is to ensure that the model constructed during this phase has not seen the future data and evaluate whether the model can produce the same results with a different set of data. This is also meant to test whether the model constructed can be generalised with a whole set of different data. Splitting the data also prevents human bias, such that the model parameters are manipulated to produce results that skew towards the expectation of the researcher. For this model construction stage, the input datasets used were from 2010-2018 and tested against 2019 – 2020 data.

4.2.2. Daily Returns Data

From the raw stocks adjusted close and market close prices, daily log returns data were calculated. Some outlier values were found in the stocks prices when returns were calculated, where MAPI (PT Mitra Adiperkasa Tbk) price dropped to zero and rose back to its original value within a day. This resulted in a -230% loss and followed then by a 230% return within a day as shown in Figure 4.5. Upon checking with other sources of data from Google Stocks and Kontan, such returns did not occur and this was an error on Yahoo Finance side. Given such outliers, the stocks return data need to be cleaned by removing any values above 3 standard

deviations or ±3$\boldsymbol{\sigma}$. This was done by calculating the Z-score of each stock and removing any value with Z-score above ±3$\boldsymbol{\sigma}$. Figure 4.6 shows the cleaned data.

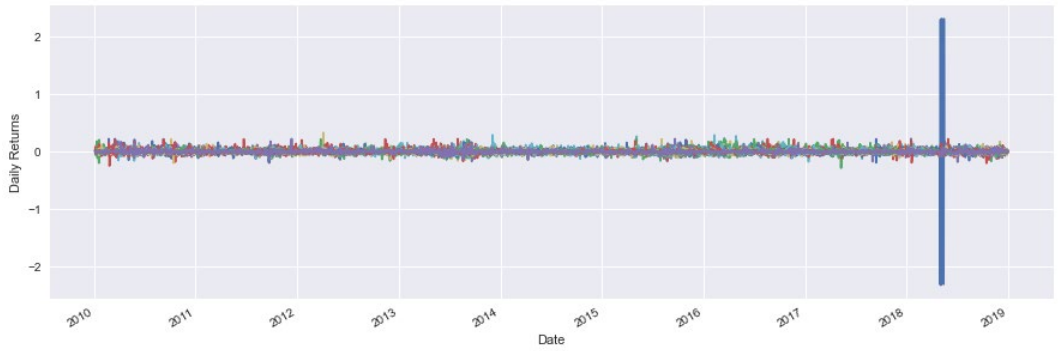

Figure 4.5 Daily Returns of Stocks

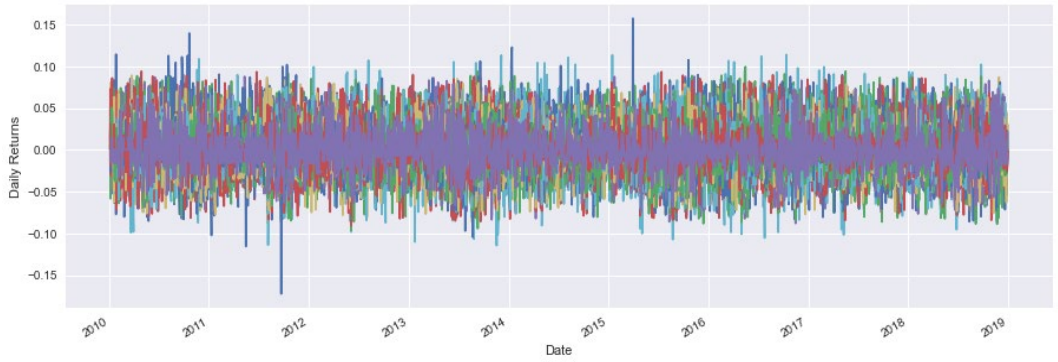

Figure 4.6 Cleaned Daily Returns Data

Figure 4.7 shows the daily, monthly and annual returns of stocks sampled. As the returns calculated were log daily returns and log returns are additive, to calculate annual or monthly returns, these returns can simply be added together for a given period of time. Figure 4.8 shows the distributions of all the returns of the stocks to show that the returns are relatively normally distributed.

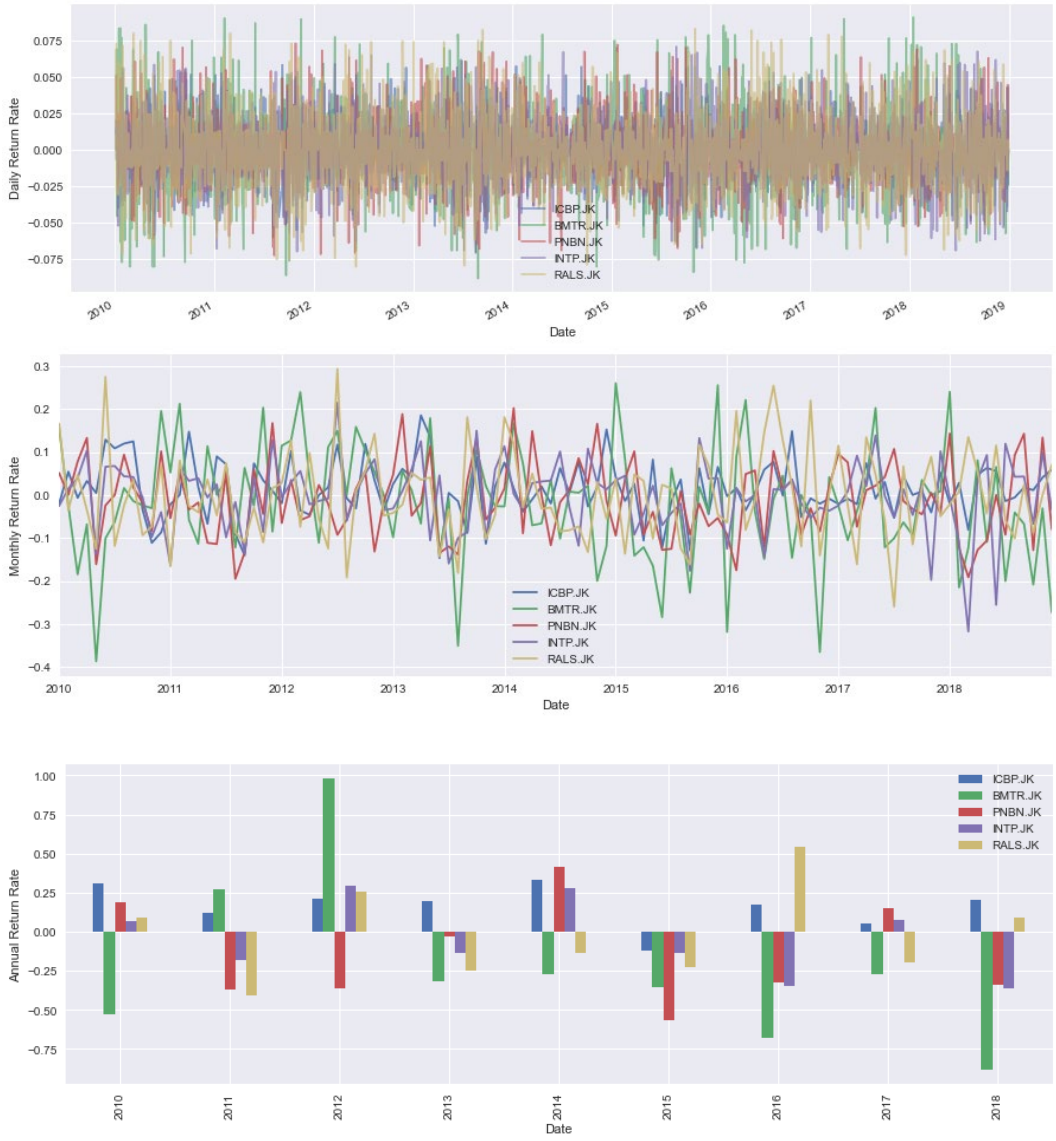

Figure 4.7 Daily, Monthly and Annual Returns of Sampled Stocks (n=5)

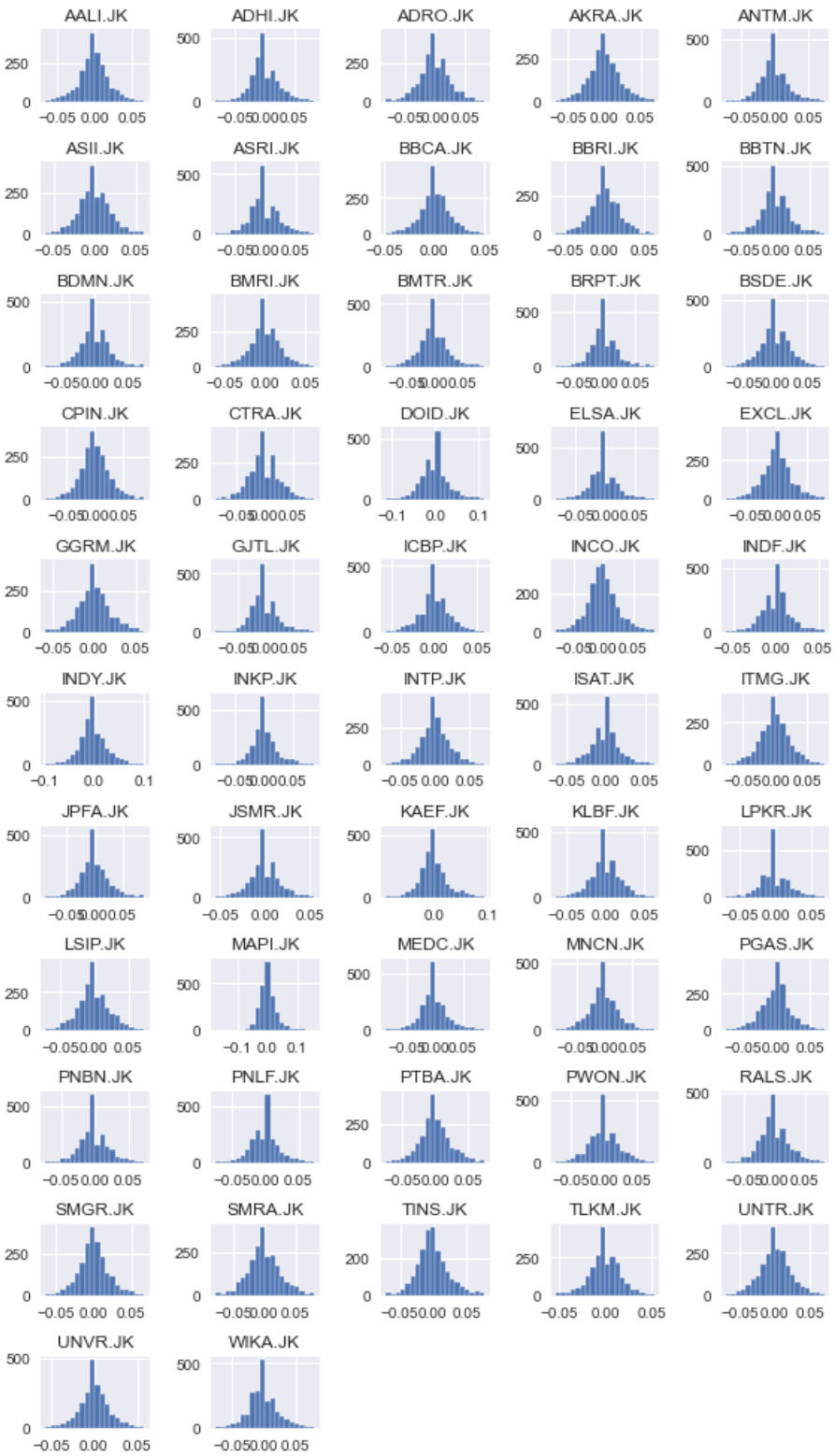

Figure 4.8 Returns distributions of all the stocks

Market daily returns were calculated in the same method as shown in Figure 4.9. In addition to this, the daily risk-free rate was calculated using the following definition, since bond yield is not confined to stocks market trading day, to get the daily risk-free rate, the bond yield was divided by the number of days until its maturity of one year.

$$DailyRisk - freeRate = \frac{DailyBondYield}{365} \tag{12}$$

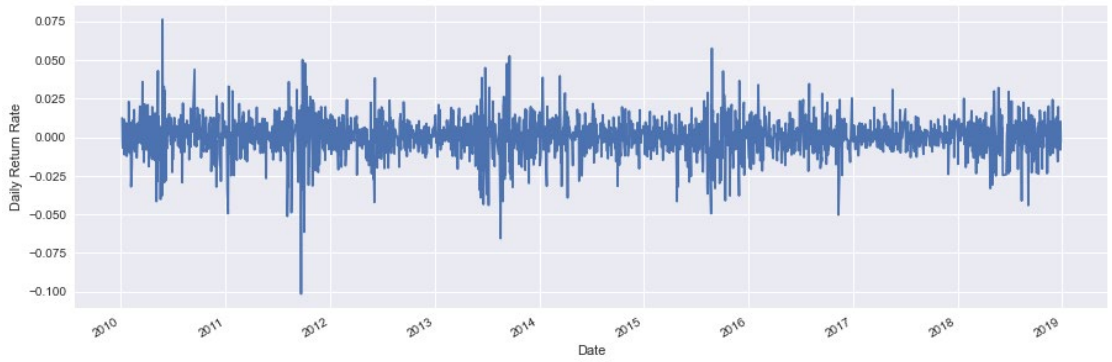

Figure 4.9 Market Daily Returns

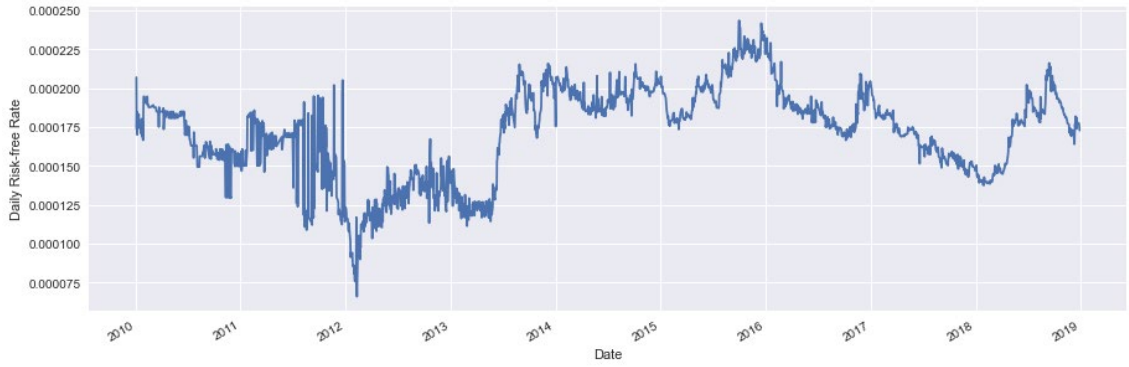

Figure 4.10 Daily Risk-free Rates

4.2.3 .Market Premium and Risk Premiums

Given the returns calculated, to further derive some of the metrics and features stated in 3.4.1, risk premiums and market premium were calculated by subtracting the daily risk-free rate from stocks returns and market returns respectively. These data were then used for linear regressions to derive regression alpha and beta.

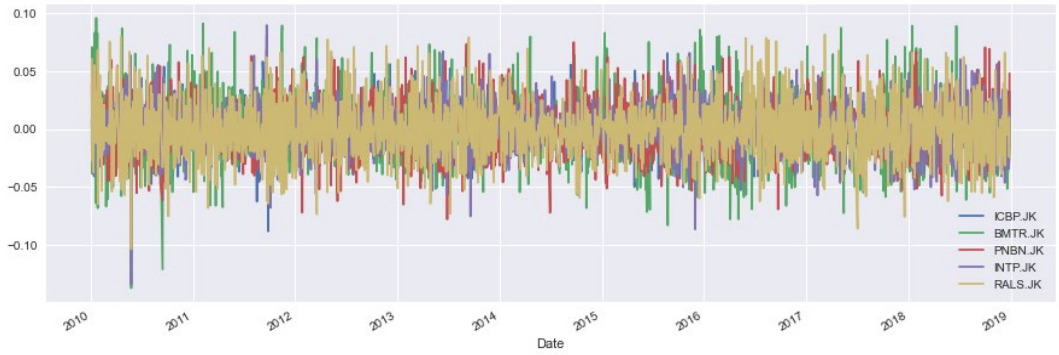

Figure 4.11 Daily Risk Premiums of Sampled Stock (n=5)

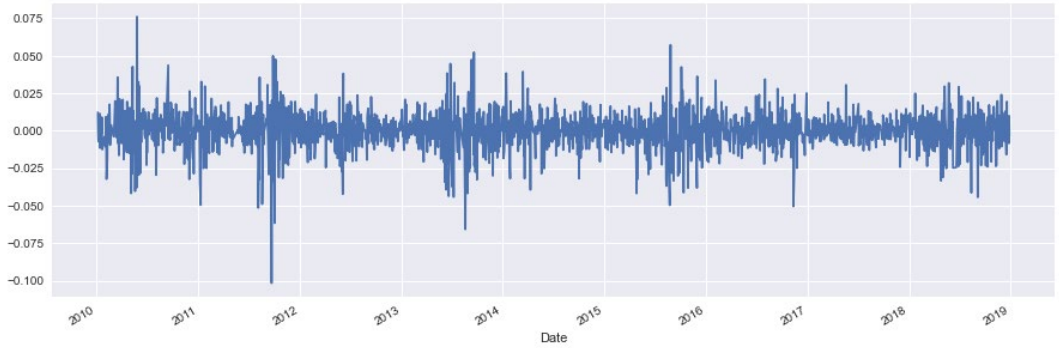

Figure 4.12 Daily Market Premiums

4.2.4. Expected Returns Testing

Five different estimates of expected return were calculated from the data. To find the estimation method that follows the realised return, p-value and t-statistics were calculated using the independent t-test with the realised return from a given year. The p-value of expected returns from CAPM ordinary least squares does not indicate that this estimate is significant in estimating realised returns as $p < 0.05$.

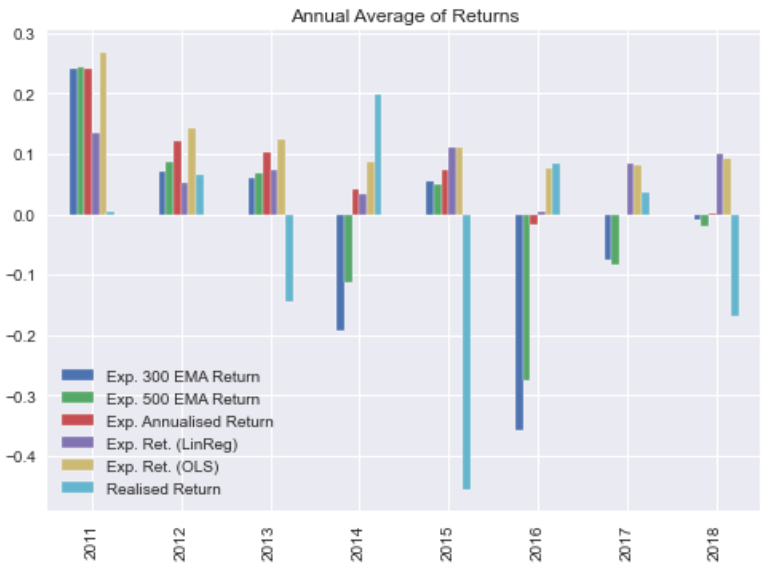

Figure 4.13 Annual Averages of Expected Returns and Realised Returns

Table 4.4 The p-value and t-statistics of Expected Returns from Independent t-test with Realised Returns

	Exp. 300 EMA Return	Exp. 500 EMA Return	Annualised Return	Exp. Ret. (LinReg)	Exp. Ret. (OLS)	Realised Return
t-statistics	0.222791	0.459279	1.510591	1.652822	2.248858	0.0
p-value	0.826916	0.653084	0.153131	0.120608	0.041142	1.0

4.2.5. Expected Volatility Testing

Volatility or risk is equally important in MPT, therefore some of the estimation methods used in estimating expected returns were implemented for volatility (standard deviation). Again, independent t-tests were done to compare them with the realised volatility. All the estimates have $p > 0.05$ therefore is significant in estimating the realised volatility.

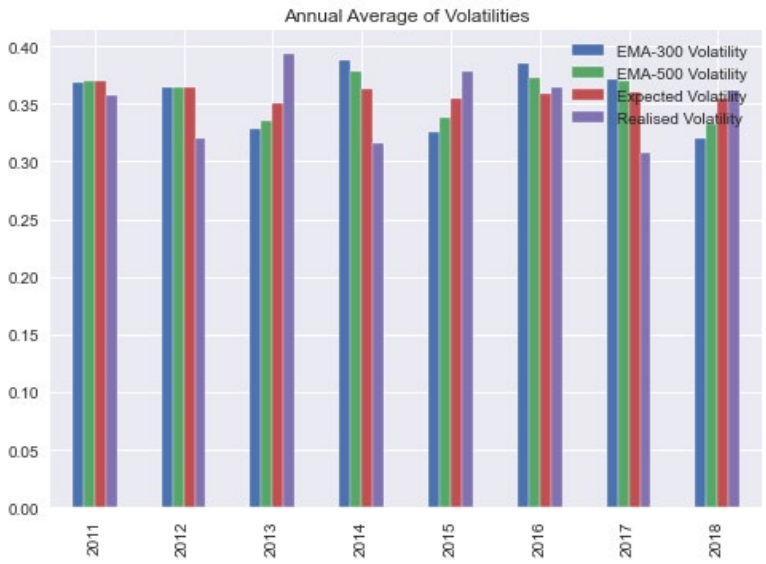

Figure 4.14 Annual Averages of Expected Volatilities and Realised Volatilities

Table 4.5 The p-value and t-statistics of Expected Volatilities from Independent t-test with Realised Volatilities

	EMA-300 Volatility	EMA-500 Volatility	Expected Volatility	Realised Volatility
t-statistics	0.430570	0.577279	0.822257	0.0
p-value	0.673336	0.572918	0.424714	1.0

4.2.6. Features from Model Construction Training Data

From prices and returns, the data were further processed into features specified in 3.4.1. These features were used as inputs for the clustering stage followed. The k-means clustering algorithm used the features dataset in its normalised form using `sklearn.preprocessing.StandardScaler()` where each value of the features was transformed into a range based on the feature's Z-score. For the agglomerative clustering, the features dataset had to be transformed into a distance matrix. This process required the features dataset to be (1) scaled into a range using `sklearn.preprocessing.StandardScaler()` and (2) transformed into a distance matrix.

Table 4.6 Features of Sampled Stocks (n=5)

Features	KLBF	ADRO	AKRA	ASII	TLKM
Expected Return (LinReg)	-0.0048	0.0604	0.0032	0.0141	0.0238
Expected Return (CAPM)	0.0696	0.0774	0.0696	0.0786	0.0688
Annualised Return	0.1919	-0.0060	0.2338	0.0909	0.0613
EMA-300 Return	-0.0511	-0.1270	-0.2087	0.1090	0.0305
Last 3 Months Change	-0.0017	-0.2405	0.0073	0.1597	0.0109
Last 6 Months Change	0.0407	-0.0729	-0.1247	0.2674	0.0655
Last 2 Year Change	0.1237	0.2077	0.0786	-0.0261	-0.0253
Annualised Volatility	0.2947	0.3980	0.3411	0.3008	0.2616
Annualised Downside Risk	0.1874	0.2501	0.2060	0.1869	0.1737

Table 4.6 Continued

Features	KLBF	ADRO	AKRA	ASII	TLKM
Sharpe	0.5044	-0.1237	0.5587	0.1583	0.0689
Expected Sharpe	-0.2950	-0.4481	-0.7523	0.2311	-0.0443
Sortino	0.7929	-0.1968	0.9253	0.2547	0.1037
Expected Sortino	-0.4330	-0.7537	-1.1221	0.3566	-0.0695
Info. Ratio	0.3787	-0.2167	0.4501	0.0352	-0.0727
Skew	0.0400	0.0474	0.1474	0.0579	-0.1041
Kurtosis	0.8203	0.7165	0.5633	0.5585	0.7538
Alpha	-0.0001	0.0002	-0.0001	0.0001	0.0001
Beta	0.3167	0.2241	0.2362	0.4084	0.3909
Beta (CAPM)	0.7130	0.9215	0.7123	0.9556	0.6909
R2	0.4722	0.4527	0.4095	0.6218	0.5189
Market Cap (IDR)	689000000 00000.0078	431800000 00000.0000	147600000 00000.0000	219620000 000000.000 0	329880000 000000.000 0
Volume-to-Market Ratio	0.0159	0.0141	0.0027	0.0085	0.0246
Realised Return	-0.1847	0.0008	-0.3920	-0.0420	-0.0757
Total Vol. Ratio	0.0159	0.0141	0.0027	0.0085	0.0246
Avg. Trading Vol. Ratio (Daily)	0.0159	0.0141	0.0026	0.0085	0.0246
Mean Rev. Growth	0.0442	0.1109	0.2876	0.1609	0.0197
Mean Net Inc. Growth	0.0223	-0.1357	0.3688	0.1499	-0.1857

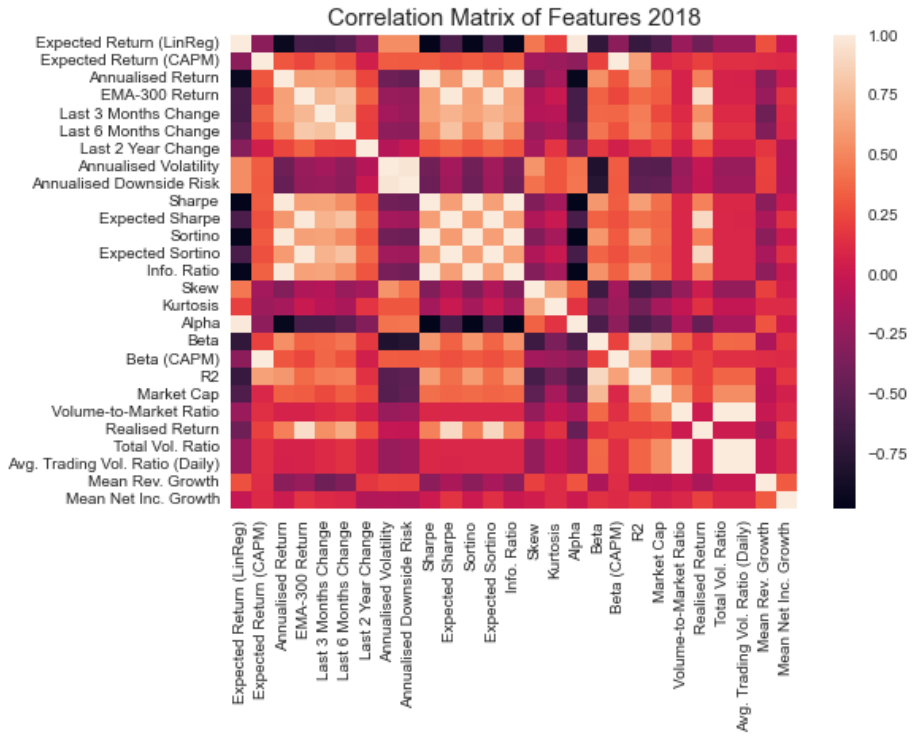

Figure 4.15 Correlation Matrix of Features from Model Construction

In total, for each iteration of the model using the model training and model testing datasets there were 27 features for each of the 52 stocks. A correlation matrix of the features was calculated to analyse the correlations between these features. To visualise these features in two dimensional axes, a PCA transformation was implemented to reduce all of the features into two principal components. A risk-return plot of stocks was produced from the 'Annualised Return' and 'Annualised Volatility' features.

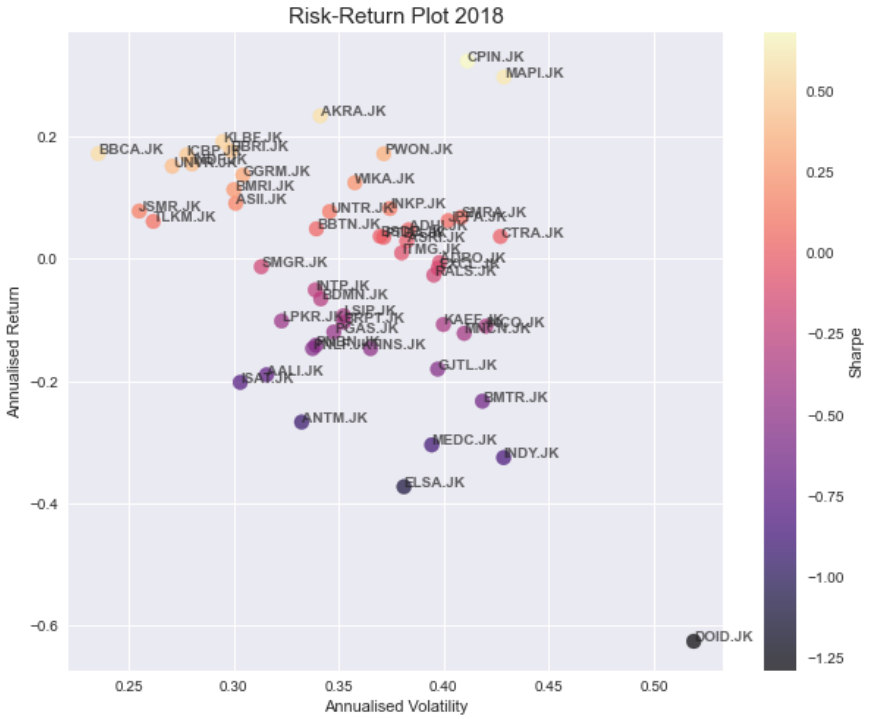

Figure 4.16 Risk-Return Plot from Model Construction

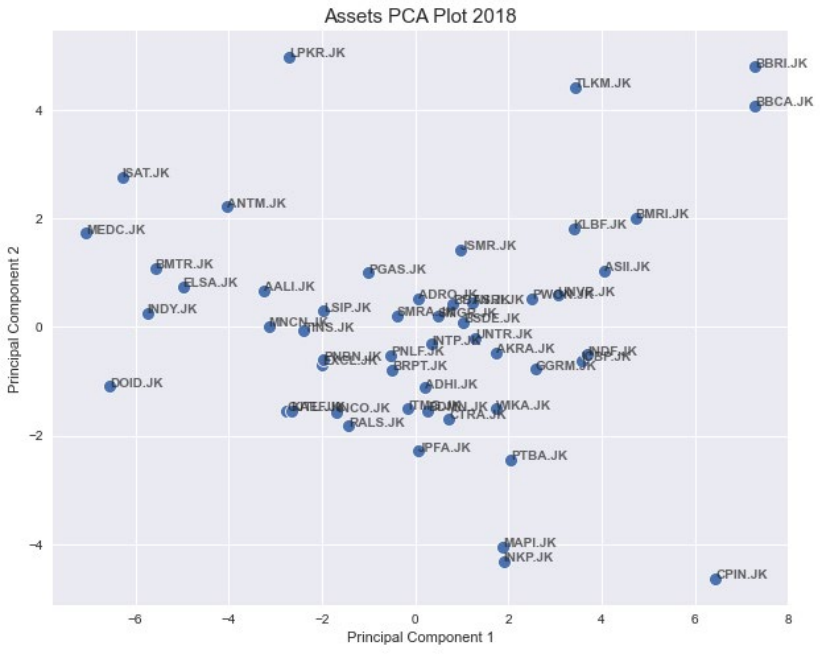

Figure 4.17 Principal Components Plot of the Stocks from 2018 Features

4.2.7. K-means Clustering Stocks Selection

One of the proposed clustering methods was k-means clustering. Utilising the features dataset constructed earlier as input, k-means clustering attempts to group selected stocks into clusters with equal variances. From the clusters of grouped stocks, the study tested whether they were viable portfolios, what each of the clusters represented and how to choose the most ideal cluster for an optimal portfolio.

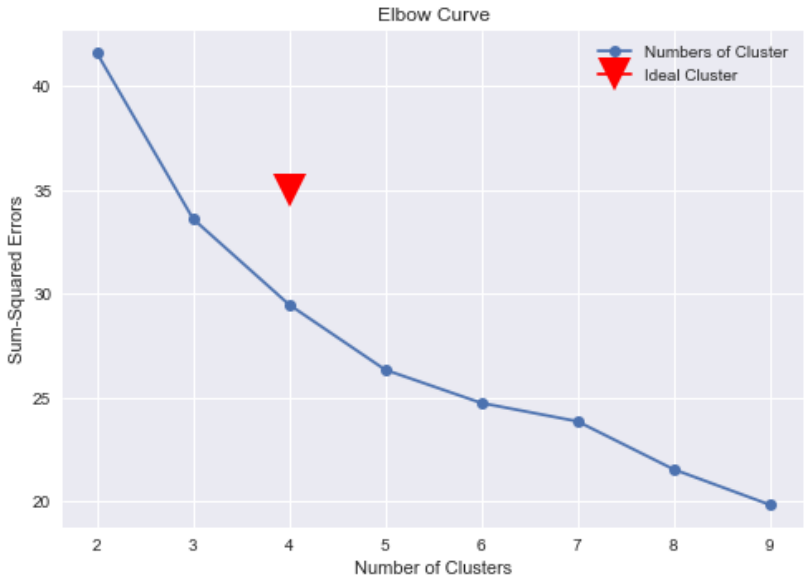

Figure 4.18 Elbow Curve of k-means Clustering on Model Construction Dataset

The sum-square errors plot Figure 4.18 suggests that four clusters are ideal since cluster numbers larger than four do not reduce the errors significantly and in preliminary tests, larger numbers of clusters overfit the data thus creating clusters with only one or two stocks, which is not very useful in portfolio construction. To visualise the clusters, PCA transformation was applied on the dataset.

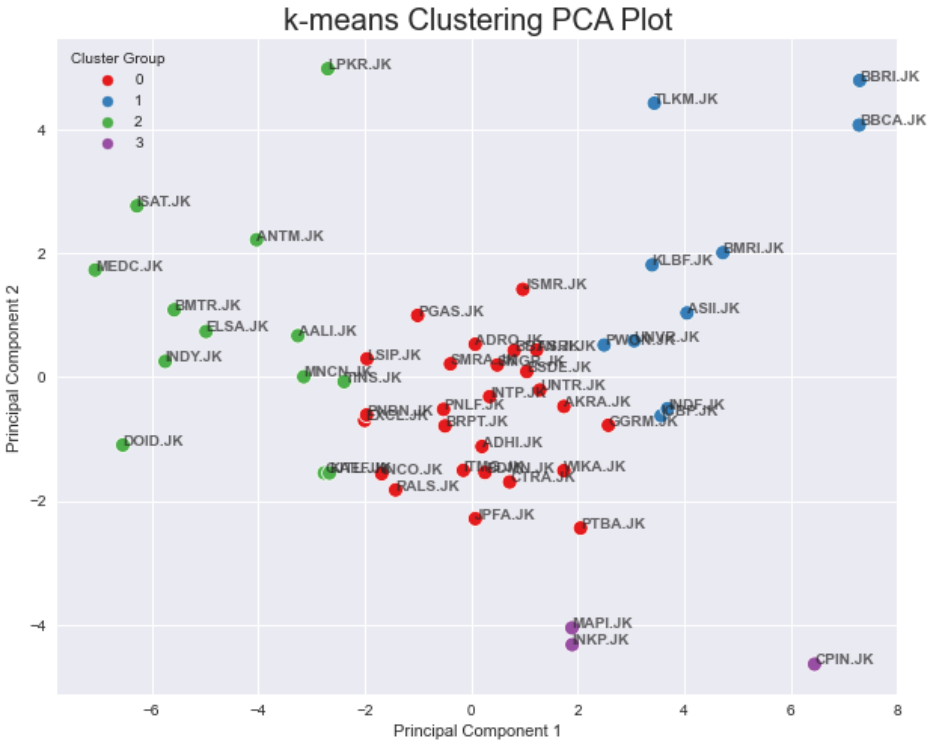

Figure 4.19 Principal Components Plot of k-means Clustering

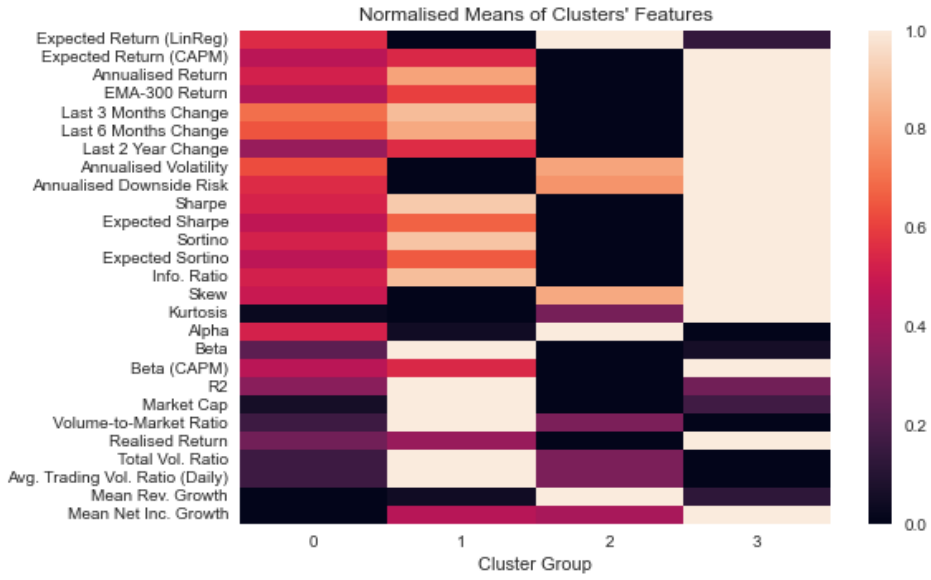

Figure 4.20 Average of Metrics for Each Cluster

To further analyse what these clusters actually represented, normalised means of the values of features in the clusters were calculated. By analysing Figure 4.20, we can observe that:

- Cluster 0 consists of stocks with medium-to-small market capitalisation with low income and revenue growth.
- Cluster 1 consists of high regression Beta stocks with large market capitalisation, large trading and market volume.
- Cluster 2 consists of high Alpha stocks with high average revenue growth and small market capitalisation.
- Cluster 3 consists of stocks with high risk-adjusted ratios, high CAPM Beta and high realised return in 2018.

To interpret these clusters more intuitively, we can express each cluster as:

- Cluster 0: Medium-Small Market Cap. Portfolio
- Cluster 1: High Beta, Large Market Cap. Portfolio
- Cluster 2: High Alpha, Small Market Cap. Portfolio
- Cluster 3: High Risk-adjusted Returns Portfolio

Testing these portfolios against data from 2019-2020 in the model construction phase shows which cluster portfolio is the most optimal. Portfolio with the same characteristic is chosen for the model validation stage with data from 2020-2021. The complete list of stocks in each cluster is provided in the Appendix 2, Table A.2.a.

4.2.8. Agglomerative Clustering Stocks Selections

Agglomerative clustering takes distance matrix as an input; thus, the features dataset of the stocks was transformed into a pairwise distances matrix of the stocks using the function provided by ***scikit-learn***. Here, to find the ideal number of clusters can be found using tree dendrogram.

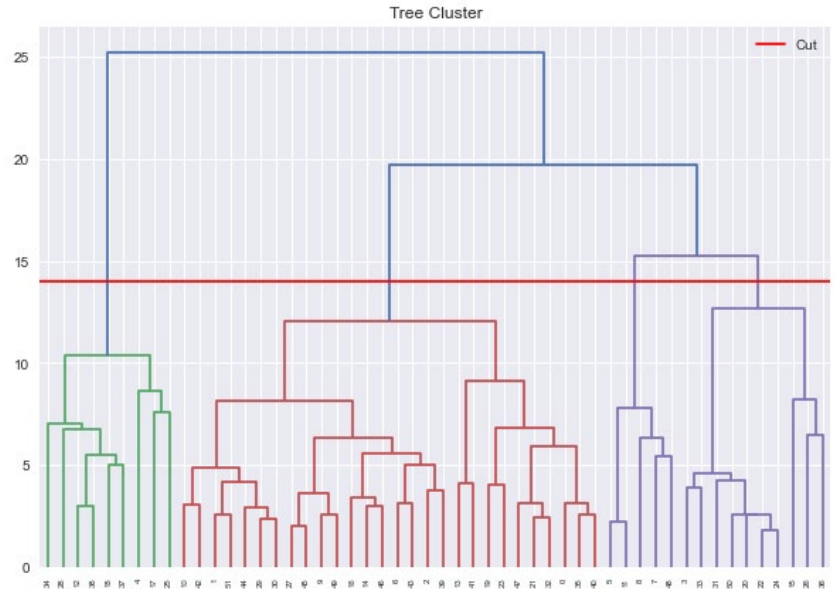

Figure 4.21 Clustering Tree Dendrogram

From the dendrogram, it can be inferred that four clusters can be implemented on the data for clusters with a decent number of stocks in them. Cutting the dendrogram lower may create clusters with only three stocks in them. Figure 4.22 shows the distance matrix before the clustering and Figure 4.23 shows the clustered distance matrix. From the clustered distance matrix, we can clearly see that clusters are indeed being formed along the main diagonal of the matrix, grouping stocks with similar hierarchical structures.

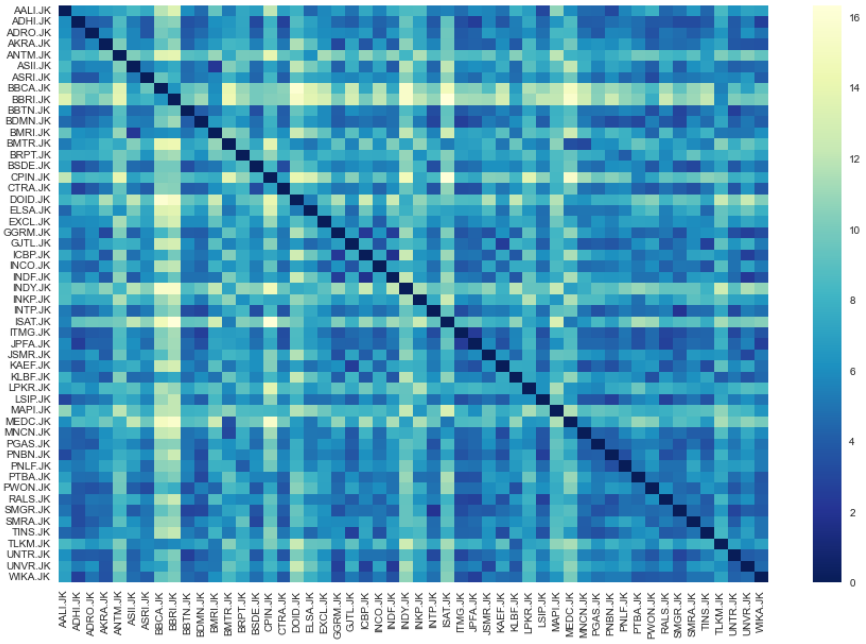

Figure 4.22 Distance Matrix Input

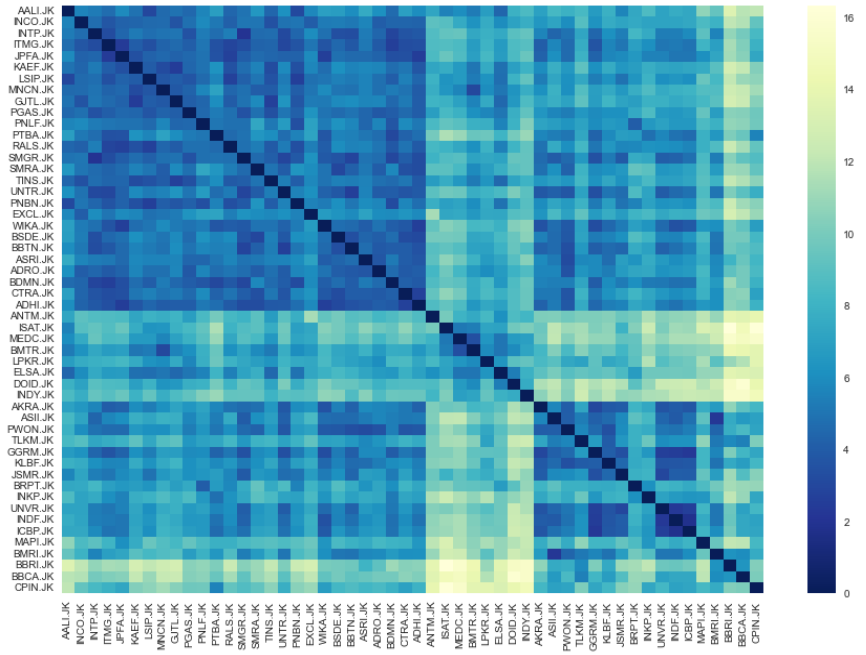

Figure 4.23 Distance Matrix after Clustering

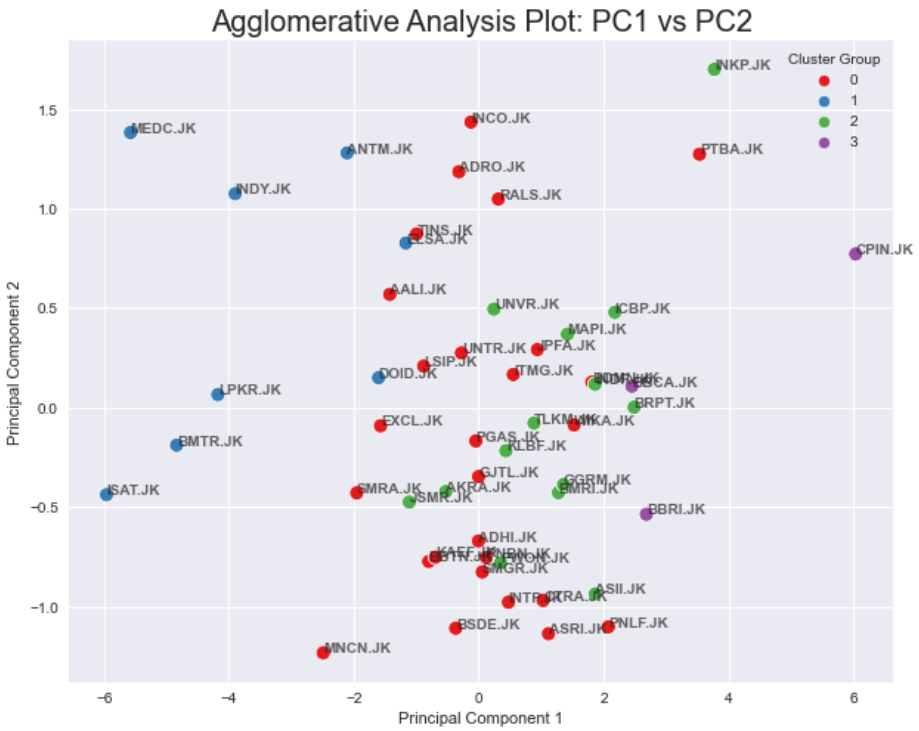

Figure 4.24 Principal Components Plot of Agglomerative Clustering

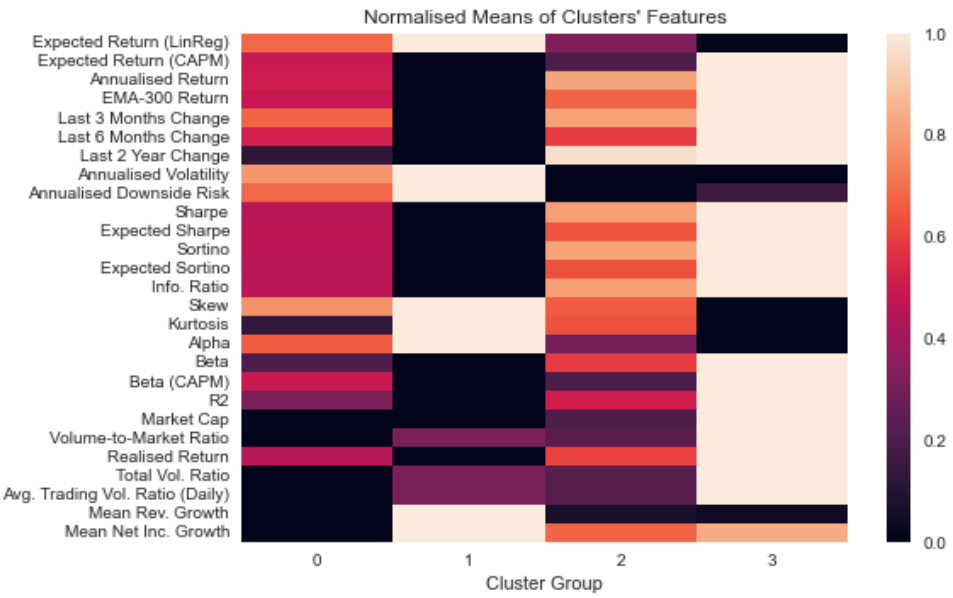

Figure 4.25 Normalised Average Features for Each Cluster

As with the previous clustering method, we can derive interpretations regarding each of these clusters from assessing the average features of each of the clusters. Looking at the normalised averages Figure 4.25, this clustering method produced a different set of portfolios. In this clustering method, cluster 0 has similar characteristics to the k-means' cluster 0. Here, cluster 1 is the high alpha cluster instead of cluster 2. Unlike the previous method where high market capitalisation and minimal volatility stocks are within the same cluster, agglomerative clustering separates the minimal volatility cluster from the high market capitalisation cluster.

- Cluster 0: Medium-Small Market Cap. Portfolio
- Cluster 1: High Alpha Portfolio
- Cluster 2: Minimal Volatility Portfolio
- Cluster: High Beta, High Risk-adjusted Returns Portfolio

Refer to Appendix 2, Table A.2.b for the complete list of stocks for each cluster.

4.2.9. Random Portfolios for Model Construction

To serve as baseline portfolios, three different randomly selected stocks portfolios were created. These portfolios were also used in assessing the performance of portfolio optimisation methods. The stocks were randomly chosen using a random sampler from ***pandas*** library. Three portfolios were created, each consisting of 5, 10 and 15 stocks. Given that the stocks of these portfolios were randomly selected, if these portfolios perform as well as the cluster portfolios, we can assume that the performance of the clustering techniques are not relevant since

a naively chosen set of stocks can perform as well as those techniques. The list of randomly selected stocks can be found in the Appendix 2, Table A.2.c.

4.2.10. Market Performance 2019 - 2020

Performance of the KOMPAS100 market during this period was calculated to provide an overall perspective of the market trends. During this period, referring to Table 4.7 and Figure 4.26, the market took a massive downturn after the first quarter of 2020 and could not return to the level that it was in 2019. This was mainly caused by the COVID-19 pandemic outbreak in Indonesia that became a public concern in the first quarter of 2020.

Figure 4.26 Market Cumulative Return 2019 - 2020

Table 4.7 Market Performance 2019 – 2020

Market 2019 - 2020	
Holding Return	-4.08%
Cumulative Return	-4.16%
Annualised Volatility	210.84%
Max Drawdown	-54.29%
Sharpe	-0.019

4.2.11. Backtesting: Equal Weights Allocation

Equal weighted portfolios are portfolios where the weight of each stock in the portfolio is $1/n$ where n is the number of stocks within the portfolio. This weight allocation is also called *naive allocation* since these weights are only proportional to the number of stocks and not adjusted to expected returns or risk. Naive allocations provide baseline performances on the portfolios. These portfolios were tested against the 2019 - 2020 dataset. Ex-post Sharpe and Sortino are risk-adjusted returns calculated from the cumulative returns instead of expected returns.

Table 4.8 Equal Weighted Random Portfolios

Random Portfolios with Equal Weights	Random 5	Random 10	Random 15
Holding Return	45.37%	26.13%	14.43%
Cumulative Return	21.87%	18.21%	9.30%
Annualised Volatility	38.80%	32.54%	30.86%
Annualised Downside Risk	25.45%	25.70%	24.17%
Max Drawdown	-74.39%	-59.55%	-54.86%
Ex-Post Sharpe	0.402	0.367	0.098
Ex-Post Sortino	0.613	0.465	0.126

Table 4.9 Equal Weighted k-means Portfolios

k-means Portfolios with Equal Weights	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	0.59%	-1.28%	51.21%	-3.22%
Cumulative Return	-4.90%	-2.61%	32.47%	-3.34%
Annualised Volatility	32.31%	26.29%	35.84%	41.48%
Annualised Downside Risk	25.97%	21.33%	27.28%	26.35%
Max Drawdown	-78.64%	-46.44%	-76.61%	-81.36%
Ex-Post Sharpe	-0.346	-0.338	0.731	-0.232
Ex-Post Sortino	-0.430	-0.416	0.961	-0.364

From Table 4.8, the 5 random stocks portfolio has the performance out of all the random portfolios. Since the stocks in these random portfolios are selected at random, these results do not reflect the fact whether smaller portfolios actually perform better, in fact a small set of assets suggests that the investment is not diversified and thus their risks are also not diversified.

Table 4.10 Equal Weighted Agglomerative Portfolios

Agglomerative Portfolios with Equal Weights	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	7.37%	51.58%	-0.52%	18.11%
Cumulative Return	2.12%	29.32%	-5.77%	15.78%
Annualised Volatility	32.74%	38.19%	27.80%	32.91%
Annualised Downside Risk	26.06%	27.48%	22.32%	23.39%
Max Drawdown	-79.60%	-79.79%	-59.69%	-34.58%
Ex-Post Sharpe	-0.126	0.604	-0.433	0.289
Ex-Post Sortino	-0.159	0.839	-0.539	0.407

Yet, this shows that through randomly sampling stocks, by chance some very profitable stocks can be selected, but this method does not guarantee replicability in the future.

On the other hand, looking at the clustered portfolios from Table 4.9 and 4.10 and their returns, portfolios constructed from agglomerative clustering seem to have better returns than those allocated by *k-means* clustering. Looking at the annualised volatility, agglomerative clustering also produced portfolios with lower risks than k-means clustering, with maximum annualised volatility topping at 27.48% while k-means cluster portfolios went up to 41.48%. Both of the best performing portfolios from the clustering techniques have quite similar performances. Referring back to Chapter 4.2.7 and 4.2.8, these two clusters were clusters that were interpreted as ‘High Alpha' portfolios. Given this, it can be inferred that to create the most optimal portfolio, we can select the cluster with the highest Alpha by average. Overall, these portfolios perform much better than the market condition, even with naively selected stocks.

4.2.12. Backtesting: Mean-Variance Optimisation (MVO)

Using the same dataset to test against as the previous backtesting, mean-variance optimisations were implemented using Monte Carlo method and portfolios with the highest Sharpe ratios were identified. Breakdown of each portfolio’s weights can be found in Appendix 2.

Comparing the performance of the random portfolios with equal weights in Table 4.8 and with mean-variance optimised weights in Table 4.11, MVO has significantly increased the holding and cumulative returns of the Random 10 portfolio while reducing its risks, which increases its risk-adjusted returns significantly as well, with an ex-post Sharpe ratio increase from, 0.367 to 0.707. The maximum drawdown (MDD) of the portfolio has also significantly decreased, from an MDD of -59.55% with equal weights to just -37.00% with MVO.

Table 4.11 Mean-Variance Optimised Random Portfolios

Random Portfolios with MVO	Random 5	Random 10	Random 15
Holding Return	-6.56%	42.43%	20.30%
Cumulative Return	-7.55%	28.13%	11.54%
Annualised Volatility	55.11%	30.94%	31.26%
Annualised Downside Risk	30.95%	24.47%	23.25%
Max Drawdown	-117.88%	-37.00%	-48.95%
Ex-Post Sharpe	-0.251	0.707	0.169
Ex-Post Sortino	-0.446	0.893	0.227

In contrast, for Random 5 portfolio, the performance becomes a stark opposite, from a holding and cumulative returns of 45.37% and 21.87% with equal weights, to a subpar level of -6.56% and -7.55% respectively for MVO. This portfolio actually performs worse than the market as a whole. Referring to the weights distribution from Table A.2.a from the Appendix, it is clear that for Random 5 portfolio, the optimiser leans very heavily on a single stock, giving it a weight of 81.5% in the portfolio.

Table 4.12 Mean-Variance Optimised k-means Portfolios

k-means Portfolios with MVO	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	-13.68%	12.84%	8.04%	35.37%
Cumulative Return	-17.81%	10.93%	-2.05%	18.46%
Annualised Volatility	34.11%	25.58%	44.52%	35.66%
Annualised Downside Risk	24.98%	20.95%	28.51%	28.59%
Max Drawdown	-87.94%	-26.71%	-123.00%	-60.80%
Ex-Post Sharpe	-0.706	0.183	-0.187	0.342
Ex-Post Sortino	-0.964	0.223	-0.292	0.427

As shown in Figure 4.12, the results of MVO on k-means portfolios are as unstable as the results previous. Immediately recognisable, for cluster 1 and 3 of the k-means portfolios, the MVO has turned the performance of these portfolios completely, from having negative returns and high MDDs to significantly large positive returns, with less volatilities and lower MDDs.

Table 4.13 Mean-Variance Optimised Agglomerative Portfolios

Agglomerative Portfolios with MVO	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	-2.18%	15.39%	5.78%	3.99%
Cumulative Return	-7.34%	4.42%	-1.20%	3.19%
Annualised Volatility	33.13%	44.03%	31.07%	45.75%
Annualised Downside Risk	26.71%	28.24%	23.33%	29.40%
Max Drawdown	-83.82%	-120.43%	-61.66%	-49.90%
Ex-Post Sharpe	-0.411	-0.042	-0.240	-0.067
Ex-Post Sortino	-0.509	-0.065	-0.320	-0.104

For a highly profitable portfolio such as the k-means cluster 2 with equal weights, MVO actually reduced the return while making the portfolio more volatile with higher MDD.

For agglomerative portfolios, MVO actually produced significantly worse portfolios than equal weights, except for portfolios from cluster 2, where the holding period return is now positive, yet the cumulative return remains negative. From these results, evidently this shows that MVO can be quite unstable as suggested by Michaud (1999) and Lopez de Prado (2016).

4.2.13. Backtesting: Hierarchical Parity Risk Portfolios (HRP)

With the same approach as the previous optimization, we can backtest how HRP would have performed under the same condition. Detailed information regarding the portfolios' weights is in Appendix 3. Table 4.14 shows the results of HRP on random portfolios.

Table 4.14 Hierarchical Risk Parity Random Portfolios

Random Portfolios with HRP	Random 5	Random 10	Random 15
Holding Return	40.61%	40.35%	10.13%
Cumulative Return	17.18%	27.22%	5.60%
Annualised Volatility	42.97%	34.13%	30.79%
Annualised Downside Risk	26.14%	26.55%	24.55%
Max Drawdown	-94.87%	-49.92%	-56.33%
Ex-Post Sharpe	0.254	0.614	-0.022
Ex-Post Sortino	0.418	0.789	-0.027

On random portfolios, the HRP has overall stable results, as all portfolios have positive returns, with Random 10 portfolio gaining a +14% holding return increase and around +9% cumulative return increase. Although, volatility for Random 5 portfolio has increased, its return actually decreased from the equal weights portfolio along with a staggering -94.87% MDD. For Random 15, the cumulative return rate is actually lower than the equal weights counterpart and is below the risk-free rate, alluding to the negative risk-adjusted returns.

Table 4.15 Hierarchical Risk Parity k-means Portfolios

k-means Portfolios with HRP	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	-9.54%	4.24%	39.23%	32.33%
Cumulative Return	-12.65%	2.85%	22.88%	21.04%
Annualised Volatility	33.52%	26.17%	36.98%	33.10%
Annualised Downside Risk	24.63%	21.99%	27.90%	27.07%
Max Drawdown	-86.07%	-40.59%	-71.44%	-64.57%
Ex-Post Sharpe	-0.564	-0.130	0.449	0.446
Ex-Post Sortino	-0.768	-0.155	0.596	0.546

For the *k-means* portfolios, interestingly both of the portfolios that resulted in negative returns have been turned into profitable portfolios, despite the returns on cluster 1 portfolio being lower than the risk-free rate. Again, similar to the random portfolios, HRP actually reduced the returns of the portfolio with the

highest returns from the equal weights portfolios, but overall, the results are more stable than MVO such that HRP generally produces portfolios with positive returns.

Table 4.16 Hierarchical Risk Parity Agglomerative Portfolios

Agglomerative Portfolios with HRP	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Holding Return	5.31%	39.23%	-0.31%	18.83%
Cumulative Return	0.50%	22.88%	-4.73%	16.58%
Annualised Volatility	33.31%	36.98%	28.57%	33.82%
Annualised Downside Risk	26.81%	27.90%	22.46%	24.26%
Max Drawdown	-83.89%	-71.44%	-62.96%	-34.95%
Ex-Post Sharpe	-0.173	0.449	-0.385	0.305
Ex-Post Sortino	-0.215	0.596	-0.489	0.425

With the *agglomerative* cluster portfolios, the results from HRP weights seem to reflect those from *k-means* cluster portfolios. In both *agglomerative* and *k-means* cluster portfolios, the cluster with the highest normalised Sharpe average performs the best.

4.3. Analysis and Model Validation

4.3.1. Model Validation

Model validation serves as an assessment to check whether the model can be generalised and is replicable with a different set of data. Taking on the information and specifications observed during the model construction stage, the constructed model was tested against a new dataset while the parameters and

methods from model construction remained unchanged. In the model construction stage, the model was tested against data from 2019 to 2020. For model validation, the model was tested against more recent data spanning from 1 January 2020 to 13 December 2021 with input data from 2010 - 2019. During model construction, some of the most important aspects discovered were:

1. Regardless of clustering methods, the cluster with the highest average of Alpha consistently outperformed other portfolios.
2. Both of the clustering methods returned quite similar groups or clusters of stocks, either method is completely viable.
3. HRP produced more stable and reliable portfolios than MVO, such that the returns from HRP portfolios were mostly positive.

With these specifications, the model can be formalised as,

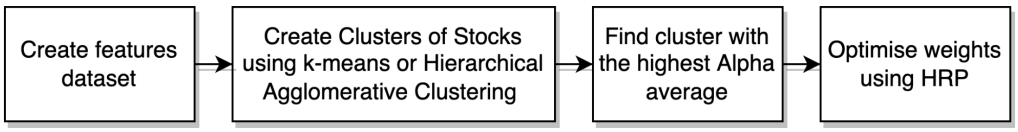

Figure 4.27 Model Outline

4.3.2. Features from Model Validation Training Data

The same approach to features extraction was implemented to the new set of data. The steps taken here were identical to those in 4.2.6. The features correlation matrix was relatively similar to the previous correlation matrix from the

model testing dataset. Figure 4.28 shows the assets PCA plot to serve as a reference for the clustering later.

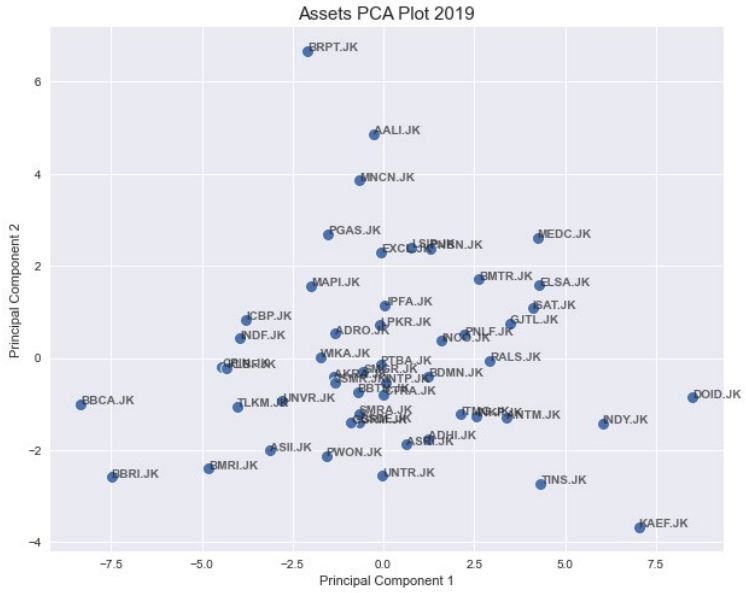

Figure 4.28 Principal Components Plot

4.3.3. k-means Clustering on Model Validation Dataset

To find the ideal number of clusters for this dataset, an error 'Elbow' curve was plotted. Evidently from Figure 4.29, for this dataset, six clusters seemed to be the ideal number. Figure 4.30 shows the different *k-means* clusters on a principal components plot. From this figure we can roughly see how *k-means* clustering groups the different stocks together in distinct groups or clusters. From the normalised features' means of the clusters shown in Figure 4.31, *k-means* cluster 3 is the 'High Alpha' cluster that we seek. For a detailed list of stocks in the clusters refer to Table A.2.d of Appendix 2.

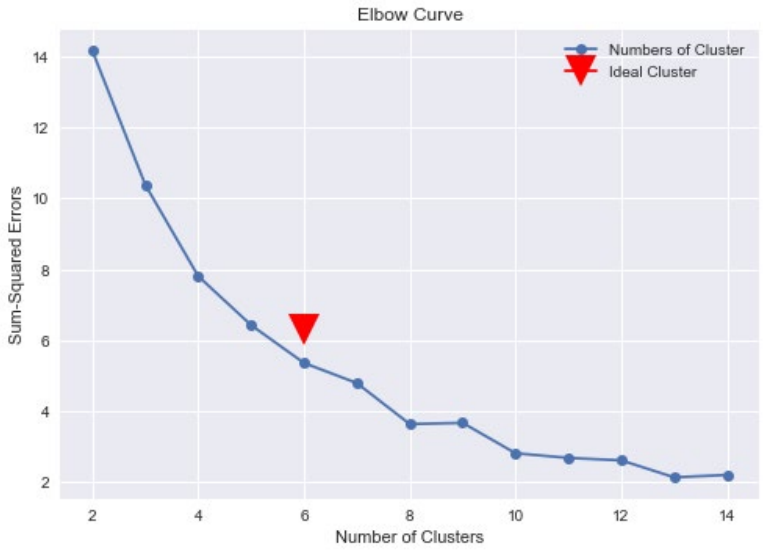

Figure 4.29 Elbow Curve of k-means Clustering on Model Validation Dataset

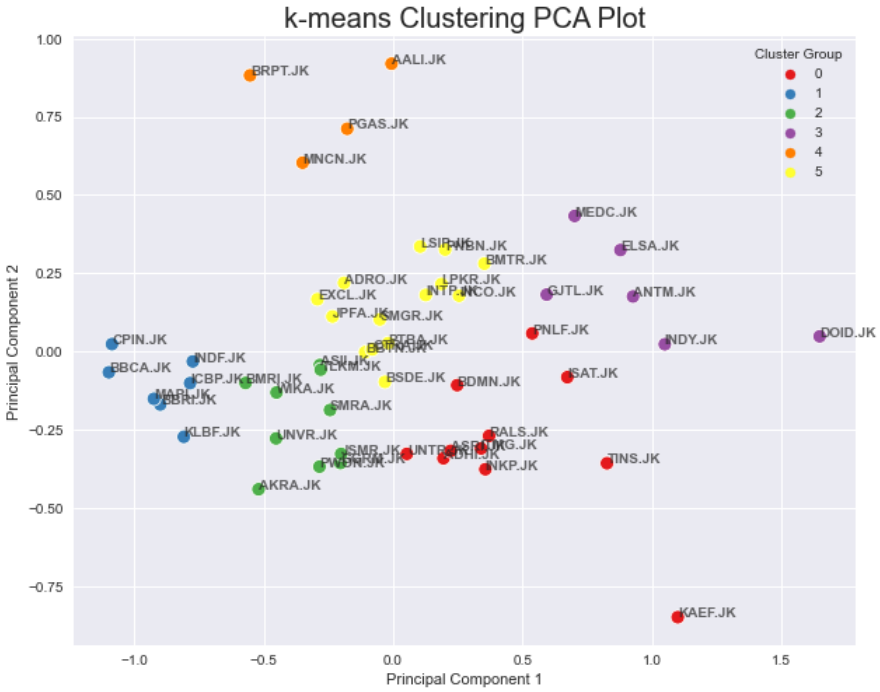

Figure 4.30 Principal Components Plot of k-means Clustered Stocks

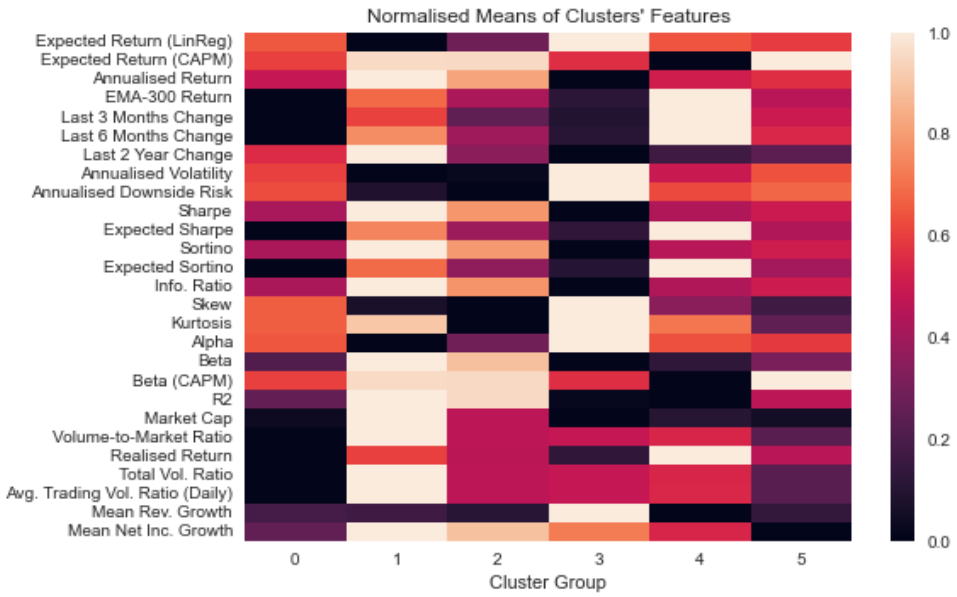

Figure 4.31 Normalised Features Means of k-means Clusters

4.3.4 Agglomerative Clustering on Model Validation Dataset

Repeating the same procedure from the previous model construction, to start hierarchical agglomerative clustering, a tree dendrogram from the stocks' features was plotted so that the different hierarchical structures can be analysed.

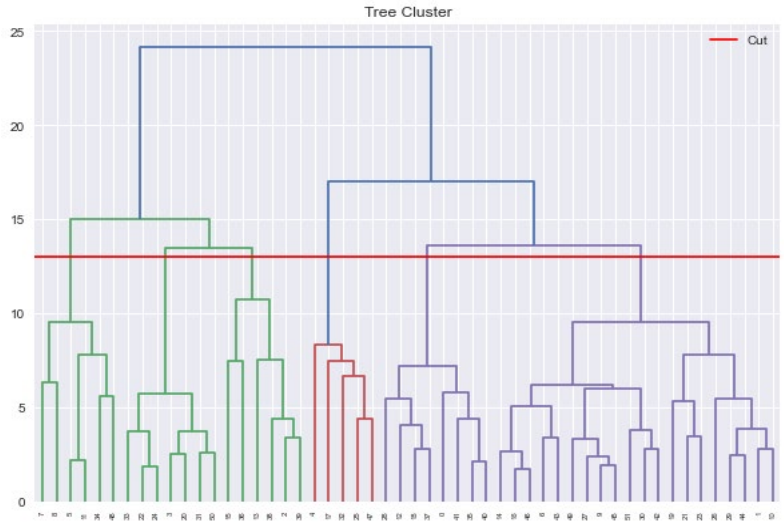

Figure 4.32 Tree Dendrogram on Model Validation Dataset

The dendrogram from Figure 4.32 suggests that the dataset can be clustered into six different clusters. Transforming the dataset into a distance matrix shown in Figure 4.33. In Figure 4.34 we can see how the distance matrix has been clustered along the main diagonal. Figure 4.35 provides a visualisation of how these stocks are clustered within the principal components plane. From Figure 4.36, for this iteration of the agglomerative clustering, the 'High Alpha' cluster was determined to be cluster 0. Table A.2.e of Appendix 2 has the list of stocks in each cluster.

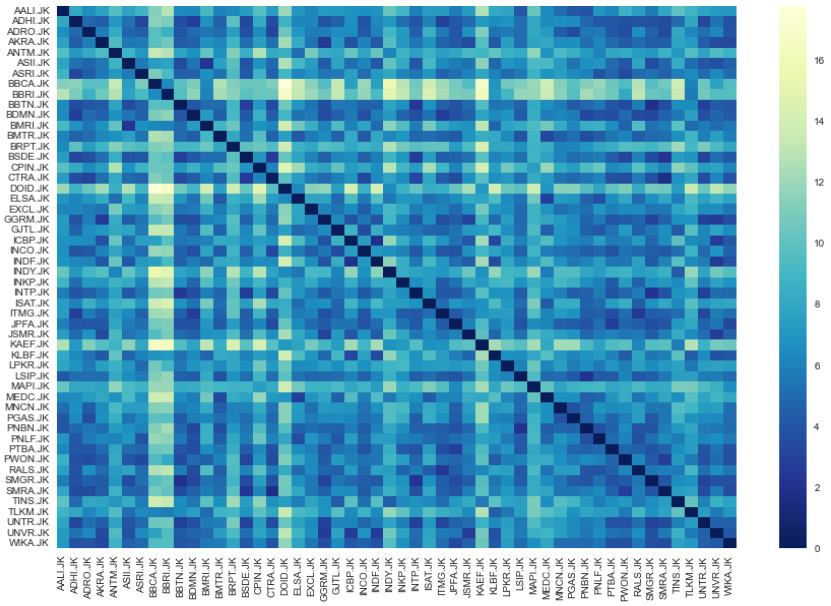

Figure 4.33 Distance Matrix from Model Validation Dataset

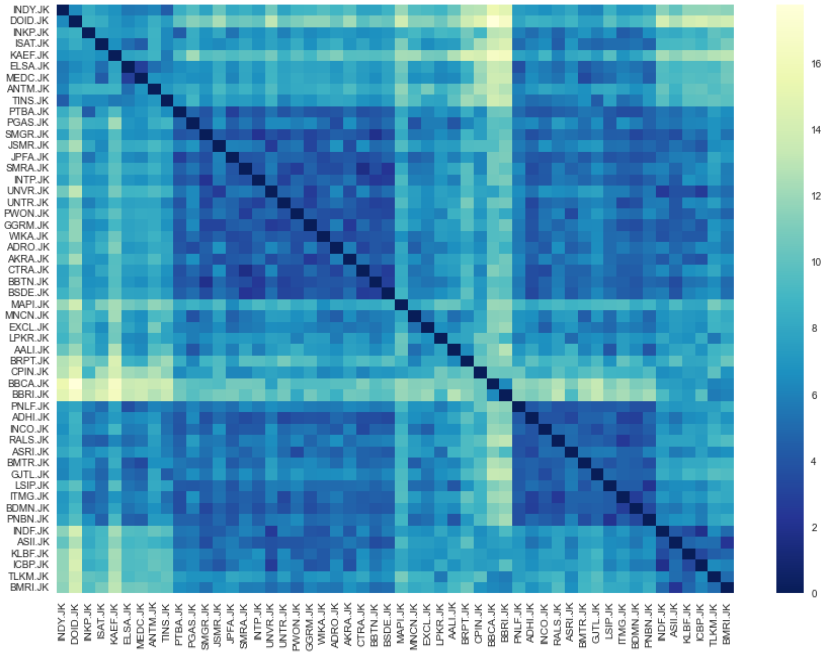

Figure 4.34 Clustered Distance Matrix

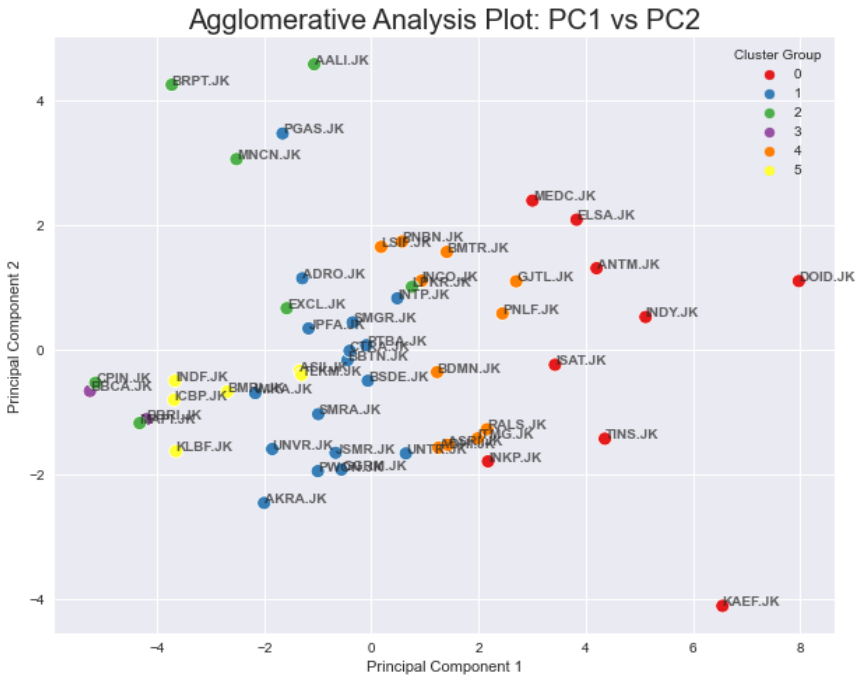

Figure 4.35 Principal Components Plot of Agglomerative Clustered Stocks.

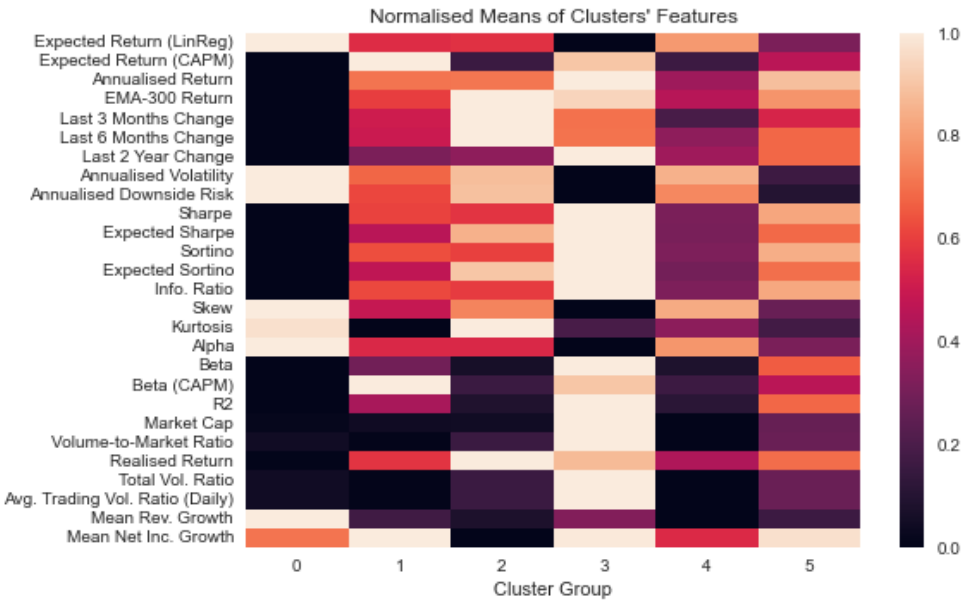

Figure 4.36 Normalised Features Means of Agglomerative Clusters

4.3.5. Market Performance 2020 - 2021

Looking at Figure 4.37, it is immediately noticeable that the KOMPAS100 market has not recovered fully from the impact of the pandemic. In January 2021, the market seemed to take on a more positive outlook, but not long after it went on a downturn again. The returns stated in Table 4.17 are not promising either. But, entering the last quarter of 2021, the market seemed to progressively improve from the previous condition. In assessing how the portfolios perform, market performance serves as a comparison.

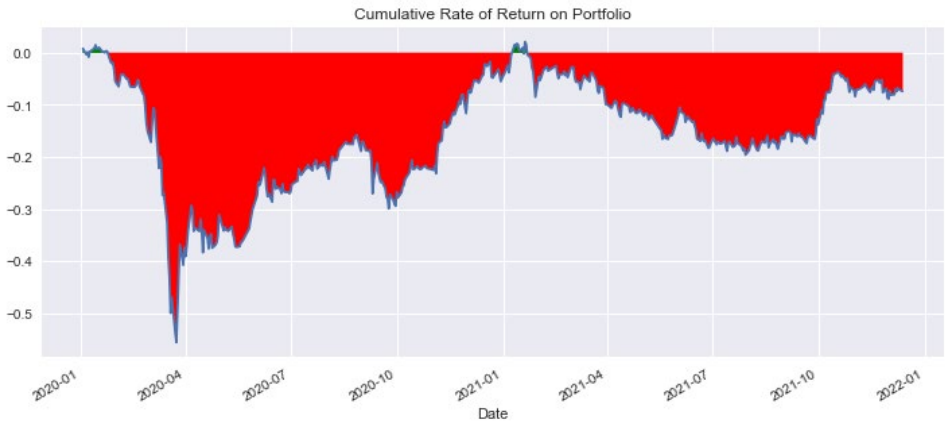

Figure 4.37 Market Cumulative Return 2020-2021

Table 4.17 Market Performance 2020-2021

Market 2020 - 2021	
Holding Return	-7.11%
Cumulative Return	-7.37%
Annualised Volatility	168.51%
Max Drawdown	-55.60%
Sharpe	-0.042

4.3.6. Validation Backtesting: Equal Weights Allocation

To provide naive baselines for the optimised portfolios, once again equal weights allocations were implemented. A new set of randomly sampled stocks portfolios was also generated. Thus, becoming the baselines to compare the optimised portfolios to. Implementing equal weights on the portfolios also allows us to test our assumption regarding the observation that 'High Alpha' clusters perform better. From Table 4.18, the random portfolios from 2020 - 2021 are not performing as well as random portfolios from 2019 - 2020. With small returns below or just close to the risk-free rate, their risk-adjusted returns are in the

negatives. Breakdown of portfolio weights can be referred to in Appendix 4, Table A.4.a and A.4.b.

Table 4.18 Equal Weighted Random Portfolios

Random Portfolios with Equal Weights	Random 5	Random 10	Random 15
Holding Return	2.57%	7.38%	5.75%
Cumulative Return	2.33%	4.61%	-0.79%
Annualised Volatility	36.08%	35.31%	33.97%
Annualised Downside Risk	23.48%	25.80%	24.53%
Max Drawdown	-72.71%	-80.55%	-73.11%
Ex-Post Sharpe	-0.109	-0.047	-0.208
Ex-Post Sortino	-0.167	-0.064	-0.288

In the previous section on k-means clustering, we determined that cluster 3 is the 'High Alpha' cluster that should give the highest return. Turning into Table 4.19, this assumption still holds true, cluster 3 or the 'High Alpha' cluster is the portfolio that results in the highest returns compared to other clusters. Nonetheless, there is one thing to consider, its MDD is the second highest compared to other clusters. Adding to this, the risk-adjusted return of this portfolio is also rather low with a Sharpe ratio of only 0.346.

Table 4.19 Equal Weighted k-means Portfolios

k-means Portfolios with Equal Weights	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Holding Return	30.15%	-6.71%	-14.09%	37.50%	-32.62%	-7.06%
Cumulative Return	13.89%	-7.89%	-18.24%	21.70%	-40.11%	-11.24%
Annualised Volatility	36.68%	27.04%	32.23%	44.68%	39.28%	33.81%
Annualised Downside Risk	25.04%	20.80%	24.52%	28.42%	29.56%	26.15%
Max Drawdown	-75.86%	-53.51%	-68.85%	-94.82%	-108.09%	-70.76%
Ex-Post Sharpe	0.208	-0.523	-0.760	0.346	-1.181	-0.518
Ex-Post Sortino	0.305	-0.680	-0.999	0.543	-1.569	-0.669

For the hierarchical agglomerative cluster, in Chapter 4.3.4, the cluster that was identified with the highest average Alpha was cluster 0. Again, the assumption holds true for this case as well, such that cluster 0 is the portfolio with the highest rate of returns with a high risk-adjusted returns for both the Sharpe and Sortino ratio. Thus, with these two pieces of evidence, the assumption regarding the 'High Alpha' is seemingly true, that 'High Alpha' clusters, based on construction and validation data, produce a set of stocks with high returns.

Table 4.20 Equal Weighted Agglomerative Portfolios

Agglom. Portfolios with Equal Weights	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Holding Return	60.99%	-10.25%	-25.40%	6.54%	-4.71%	-3.10%
Cumulative Return	38.05%	-14.44%	-31.21%	6.14%	-12.79%	-3.74%
Annualised Volatility	43.55%	34.44%	33.58%	33.49%	35.01%	27.60%
Annualised Downside Risk	27.90%	25.91%	25.27%	22.07%	25.89%	22.19%
Max Drawdown	-88.56%	-70.88%	-84.76%	-49.83%	-83.10%	-52.74%
Ex-Post Sharpe	0.730	-0.601	-1.116	-0.004	-0.544	-0.362
Ex-Post Sortino	1.139	-0.799	-1.483	-0.006	-0.736	-0.451

4.3.7. Validation Backtesting: Optimised Portfolios

Given the results from the equal weights allocation, to test how the MVO and HRP perform with a new dataset. The random portfolios and the 'High Alpha' clusters from both clustering methods were optimised. From the model construction, it was observed that the MVO provided erratic and unstable results, which turned profitable portfolios under equal weights allocation and HRP into portfolios with negative returns. Portfolios weights can be found in Appendix 4.

Table 4.21 MVO Portfolios from Model Validation

MVO Portfolios	Random 5	Random 10	Random 15	k-means Cluster 3	Agglom. Cluster 0
Holding Return	10.96%	11.79%	-4.01%	0.36%	33.29%
Cumulative Return	10.28%	8.86%	-5.39%	-13.17%	14.44%
Annualised Volatility	30.63%	34.17%	31.04%	54.92%	47.64%
Annualised Downside Risk	19.88%	26.51%	23.40%	33.49%	29.84%
Max Drawdown	-44.77%	-72.33%	-65.31%	-96.46%	-90.83%
Ex-Post Sharpe	0.131	0.076	-0.375	-0.354	0.172
Ex-Post Sortino	0.202	0.098	-0.498	-0.580	0.274

And again, here observed in Figure 4.21, the implementation of the MVO seems to produce unstable results with some portfolios being optimised and others simply returning a worse rate of returns. This indicates that the statement made by Michaud (1989) is evidently true.

Table 4.22 HRP Portfolios from Model Validation

HRP Portfolios	Random 5	Random 10	Random 15	k-means Cluster 3	Agglom. Cluster 0
Holding Return	6.95%	4.34%	10.70%	63.19%	58.38%
Cumulative Return	6.48%	2.32%	3.32%	37.01%	36.19%
Annualised Volatility	31.64%	33.78%	33.51%	45.48%	44.13%
Annualised Downside Risk	20.92%	24.88%	23.80%	28.50%	28.05%
Max Drawdown	-56.34%	-79.46%	-71.44%	-94.68%	-91.07%
Ex-Post Sharpe	0.007	-0.117	-0.088	0.676	0.678
Ex-Post Sortino	0.010	-0.159	-0.124	1.079	1.067

Contrary to MVO, referring to Table 4.22 shows that HRP portfolios are far more stable in performance. Compared with the randomly selected portfolios, the cluster portfolios from both k-means and hierarchical agglomerative clustering show significantly higher returns than the randomly selected portfolios. HRP also showed stable results over all the portfolios with all the portfolios providing positive returns. The returns from the HRP optimal clusters portfolios are significantly higher than the rest of the portfolios, suggesting that the model has indeed created optimal portfolios that outperformed the market and the random portfolios. In addition to this, the risk-adjusted returns Sharpe and Sortino are also significantly higher than the rest, with more than six times the rate of the random portfolios. Also, these risk-adjusted returns were calculated using cumulative returns, if holding returns were to be considered instead, these optimal portfolios would have Sharpe ratios > 1.0 and Sortino ratios closer to 2.0, such that their returns exceed their risks and signify their ability to outperform the market.

4.4. Discussions

The objective of this study is to construct the most optimal portfolio using a set of data science techniques implemented in an ensemble model. Two of the main tasks in portfolio construction are assets selection and assets allocation. While there are studies that tackle each of those tasks, such as using clustering algorithms to select stocks and construct portfolios by Nanda, Mahanty & Tiwari (2010) and León, Aragón, Sandoval et al. (2017) and a more robust asset allocation method through hierarchical parity risk introduced by Lopez de Prado (2016). This study aimed to use these two methods in conjunction with each other to create the most optimal portfolio.

Looking back at the results in the validation stage, these two methods worked in perfect tandem with each other. Referring to the performance of the cluster portfolios with HRP stated in Table 4.22, despite the downturn of the market in 2020 - 2021, these portfolios managed to outperform the market by a large margin. Figure 4.38 and 4.39 show how these portfolios diverged from the market downturn and reached cumulative returns in the excess of 50% through 2021.

For the given set of data, the model has undoubtedly succeeded in creating optimal portfolios and by comparing the portfolios with portfolios constructed from a random set of stocks, with confidence it can be assumed that these portfolios with their performance are not occurring randomly. The model has also been proven that its results are replicable even with a different set of data.

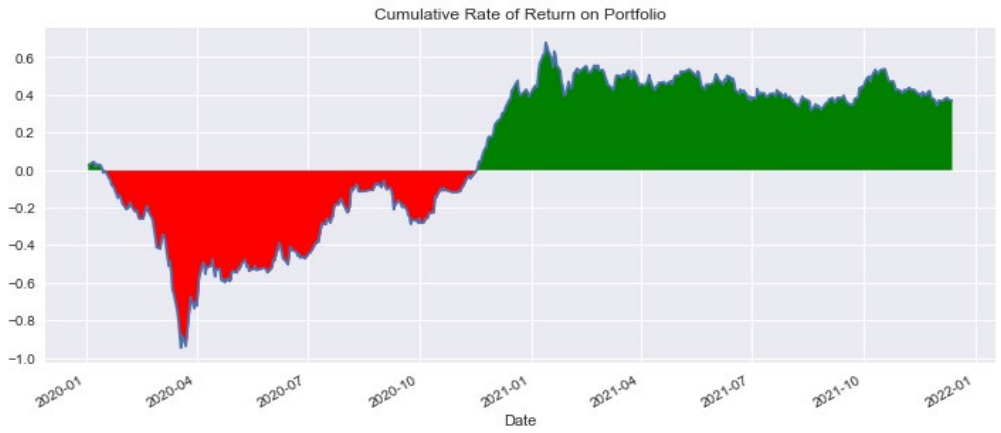

Figure 4.38 Cumulative Return of k-means Cluster 3 Portfolio with HRP Allocation

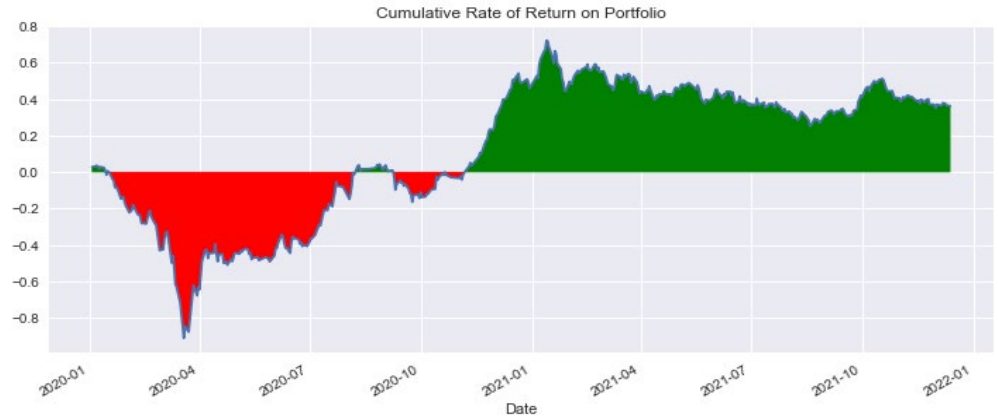

Figure 4.39 Cumulative Return of Agglomerative Cluster 0 Portfolio with HRP Allocation

Overall, the study has shown that data science techniques such as clustering, Monte Carlo simulation, features engineering and exploratory data analysis have shown to be extremely useful in portfolio management and that they can help and support investors managing their investments. The model and these techniques are not limited to stocks data, different assets can be considered as well, as long as their time-series data are available.

5. CONCLUSION

5.1. Conclusion

Data science techniques can help in portfolio management. In each aspect of portfolio management, the proposed techniques have given a significant performance in comparison to baseline, naive-weighted portfolios and outperformed the market greatly. The portfolios constructed by the model proposed in this study were optimal given that the portfolios in the validation stage have relatively high holding returns and better risk-adjusted returns compared to market and random portfolios.

For the clustering methods, both *k-means* and *hierarchical agglomerative clustering* did not produce significantly different results but referring to the clustered stocks lists in Appendix 2, *agglomerative hierarchical clustering* tend to produce clusters of optimal portfolios with more stocks in them. Therefore, if an investor or portfolio manager wants a larger portfolio, they can opt for hierarchical clustering and if smaller portfolio is preferred then *k-means* should be chosen.

As suggested by Lopez de Prado (2016, 2018), HRP allocations have been proven to outperform MVO allocations. Yet, the cause of this could very well be that expected values of mean-variance were slightly off, as stated by Michaud (1989) MVO can be very sensitive to the estimated inputs. Thus, we cannot state with full confidence that the poor performances of the MVO portfolios are essentially due to the fact that MVO is unreliable, we believe that the

implementation of the MVO algorithm or the initialisation of the inputs may have impacted the results produced. Nonetheless, from this study we can conclude that HRP is the preferred weights optimisation for the model.

5.2. Limitations

This study was implemented on a small number of stocks as 52 stocks only represent less than 10% of the whole IDX listings (+600 stocks). Computational resources were also limited such that more complex and robust machine learning models may improve these results in addition to running more training to construct better models. Portfolio holding period can also be reduced such that one could have a portfolio that gets adjusted quarterly or bi-annually. The type of assets considered in the study was also limited to stocks while there are other viable assets which have intraday data similar to the stocks considered in this study and can be used in the same way. Weights optimisation methods and clustering analysis implemented in this study were only limited to two for each case, there are numerous alternatives that can be tested as well.

5.3. Recommendations

Asset allocation based on rigorous mathematics such as HRP can be used instead of mean-variance optimisation and with easy access to tools such as Python, any investor could implement it with minimal programming ability. There are readily available portfolio management libraries on Python such as ***pyfolio*** and ***pyportfolioopt*** that can aid everyday investors in optimising their portfolio. These

Python libraries have other portfolio optimisers included in them as well such as the Black-Litterman allocation, Conditional Value-at-Risk minimisation and maximum quadratic utility. Unlike this study, these libraries are far easier to implement since the functions from these libraries can simply be imported into the program instead of explicitly calculating or simulating each process in the optimisation methods as performed in this study, where observing how each process of the methods took place was crucial. The assets studied can be expanded to other types of assets as well. As mentioned earlier, other assets such as ETFs, mutual funds and foreign stocks can be considered for in addition to stocks. The programming code from this study is also available through direct request to the author for further research and studies.

BIBLIOGRAPHY

Aroussi, R. (2021). *yfinance - Download market data from Yahoo! Finance's API*. GitHub Repository. https://github.com/ranaroussi/yfinance

De Spiegeleer, J., Madan, D.B., Reyners, S., Schoutens, W. (2018). Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance*, 18(10), 1635-1643. https://doi.org/10.1080/14697688.2018.1495335.

del Canto, A. B. (2021). *investpy - Financial Data Extraction from Investing.com with Python.* GitHub Repository. https://github.com/alvarobartt/investpy

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Oliphant, T. E. et al. (2020). Array programming with NumPy. *Nature*, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Fabozzi, F. J., & Pachamanova, D. A. (2016). *Portfolio Construction and Analytics* (Frank J. Fabozzi Series). John Wiley & Sons, New Jersey.

Fischer, T. & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2) 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

Hilpisch, Y. (2020). *Artificial Intelligence in Finance: A Python-Based Guide*. O'Reilly Media, Inc., Sebastopol.

Karim, F., Majumdar, S., Darabi, H. & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Network*, 116, 237-245. https://doi.org/10.1016/j.neunet.2019.04.014

León, D., Aragón, A., Sandoval, J., Hernández, G., Arévalo, A., Niño, J. (2017). Clustering algorithms for Risk-Adjusted Portfolio Construction. *Procedia Computer Science*, 108, 1334-1343. https://doi.org/10.1016/j.procs.2017.05.185

Lewinson, E. (2020). *Python for Finance Cookbook*. Packt Publishing, Birmingham.

Lintner, J. (1965). Security prices, risk, and maximal gains from diversification. *The Journal of Finance*, 20(4), 587-615.

Lopez de Prado, M. (2016). Building Diversified Portfolios that Outperform Out of Sample. *The Journal of Portfolio Management,* 42(4), 59-69.

Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons, New Jersey.

Lopez de Prado, M. (2020). *Machine Learning for Asset Managers* (Elements in Quantitative Finance). Cambridge University Press, Cambridge.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91. https://doi.org/10.2307/2975974.

Mattson M. P. (2014). Superior pattern processing is the essence of the evolved human brain. *Frontiers in Neuroscience*, 8(265). https://doi.org/10.3389/fnins.2014.00265

McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media, Inc., Sebastopol.

McKinney, W. et al. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 445, 51–56.

Michaud, R. (1989). The Markowitz Optimization Enigma: Is 'Optimized' Optimal? *Financial Analyst Journal,* 45(1), 31-42.

Mossin, J. (1966). Equilibrium in a Capital Asset Market. *Econometrica*, *34*(4), 768–783. https://doi.org/10.2307/1910098

Nanda, S.R., Mahanty, B. & Tiwari, M.K. (2010). Clustering Indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12) 8793-8798. https://doi.org/10.1016/j.eswa.2010.06.026

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12, 2825–2830.

Provost, F. & Fawcett, T. (2013). Data Science and its Relationship to Big Data and Data-Driven Decision Making. *Big Data,* 1(1) 51-59. http://doi.org/10.1089/big.2013.1508

Raffinot, T. (2017). Hierarchical Clustering-Based Asset Allocation. *The Journal of Portfolio Management*, 44(2), 89-99.

Rasekhschaffe, K. C. & Jones R. C. (2019). Machine Learning for Stock Selection. *Financial Analysts Journal*, 75(3), 70-88. https://doi.org/10.1080/0015198X.2019.1596678

Roy, A. D. (1952). Safety First and the Holding of Assets. *Econometrica*, *20*(3), 431–449. https://doi.org/10.2307/1907413

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and Statistical Modelling with Python. *9th Python in Science Conference*.

Sharpe, W. F. (1963) A Simplified Model for Portfolio Analysis. *Management Science*, 9(2), 277-293. Retrieved from: https://www.jstor.org/stable/2627407

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3), 425-442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Shermer, M. (2008). Patternicity: Finding Meaningful Patterns in Meaningless Noise. Scientific American. Retrieved on Dec 6, 2020 from https://www.scientificamerican.com/article/patternicity-finding-meaningful-patterns/

Siami-Namini, S., Tavakoli, N. & Namin, A.S. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, 2018*, 1394-1401. https://doi.org/10.1109/ICMLA.2018.00227.

Tatsat, H., Puri, S. & Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprint for Finance*. O'Reilly Media Inc., Sebastopol.

The Pandas Development Team (2020). pandas-dev/pandas: Pandas (Version latest). doi:10.5281/zenodo.3509134

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D. et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. doi:10.21105/joss.03021

Zhai, J., Cao, Y. & Liu, X. (2020). A neural network enhanced volatility component model. *Quantitative Finance*, 20(5), 783-797. https://doi.org/10.1080/14697688.2019.1711148.

APPENDIX

Appendix 1. List of Stocks Selected in 2021 from KOMPAS100

Number	Symbol	Company
0	AALI.JK	Astra Agro Lestari Tbk.
1	ADHI.JK	Adhi Karya (Persero) Tbk.
2	ADRO.JK	Adaro Energy Tbk.
3	AKRA.JK	AKR Corporindo Tbk.
4	ANTM.JK	Aneka Tambang Tbk.
5	ASII.JK	Astra International Tbk.
6	ASRI.JK	Alam Sutera Realty Tbk.
7	BBCA.JK	Bank Central Asia Tbk.
8	BBRI.JK	Bank Rakyat Indonesia (Persero) Tbk.
9	BBTN.JK	Bank Tabungan Negara (Persero) Tbk.
10	BDMN.JK	Bank Danamon Indonesia Tbk.
11	BMRI.JK	Bank Mandiri (Persero) Tbk.
12	BMTR.JK	Global Mediacom Tbk.
13	BRPT.JK	Barito Pacific Tbk.
14	BSDE.JK	Bumi Serpong Damai Tbk.
15	CPIN.JK	Charoen Pokphand Indonesia Tbk
16	CTRA.JK	Ciputra Development Tbk.
17	DOID.JK	Delta Dunia Makmur Tbk.
18	ELSA.JK	Elnusa Tbk.
19	EXCL.JK	XL Axiata Tbk.
20	GGRM.JK	Gudang Garam Tbk.
21	GJTL.JK	Gajah Tunggal Tbk.
22	ICBP.JK	Indofood CBP Sukses Makmur Tbk.
23	INCO.JK	Vale Indonesia Tbk.
24	INDF.JK	Indofood Sukses Makmur Tbk.

Number	Symbol	Company
25	INDY.JK	Indika Energy Tbk.
26	INKP.JK	Indah Kiat Pulp & Paper Tbk.
27	INTP.JK	Indocement Tunggal Prakarsa Tbk.
28	ISAT.JK	Indosat Tbk.
29	ITMG.JK	Indo Tambangraya Megah Tbk.
30	JPFA.JK	Japfa Comfeed Indonesia Tbk.
31	JSMR.JK	Jasa Marga (Persero) Tbk.
32	KAEF.JK	Kimia Farma Tbk.
33	KLBF.JK	Kalbe Farma Tbk.
34	LPKR.JK	Lippo Karawaci Tbk.
35	LSIP.JK	PP London Sumatra Indonesia Tbk.
36	MAPI.JK	Mitra Adiperkasa Tbk.
37	MEDC.JK	Medco Energi Internasional Tbk.
38	MNCN.JK	Media Nusantara Citra Tbk.
39	PGAS.JK	Perusahaan Gas Negara Tbk.
40	PNBN.JK	Bank Pan Indonesia Tbk
41	PNLF.JK	Panin Financial Tbk.
42	PTBA.JK	Bukit Asam Tbk.
43	PWON.JK	Pakuwon Jati Tbk.
44	RALS.JK	Ramayana Lestari Sentosa Tbk.
45	SMGR.JK	Semen Indonesia (Persero) Tbk.
46	SMRA.JK	Summarecon Agung Tbk.
47	TINS.JK	Timah Tbk.
48	TLKM.JK	Telkom Indonesia (Persero) Tbk.
49	UNTR.JK	United Tractors Tbk.
50	UNVR.JK	Unilever Indonesia Tbk.
51	WIKA.JK	Wijaya Karya (Persero) Tbk.

Source: Kontan.co.id

Appendix 2. Lists of Clustered Stocks and Random Selected Stocks

Table A.2.a Stocks List from k-means Clusters in Model Construction

Cluster 0	Cluster 1	Cluster 2	Cluster 3
ADHI.JK	ASII.JK	AALI.JK	CPIN.JK
ADRO.JK	BBCA.JK	ANTM.JK	INKP.JK
AKRA.JK	BBRI.JK	BMTR.JK	MAPI.JK
ASRI.JK	BMRI.JK	DOID.JK	
BBTN.JK	ICBP.JK	ELSA.JK	
BDMN.JK	INDF.JK	GJTL.JK	
BRPT.JK	KLBF.JK	INDY.JK	
BSDE.JK	PWON.JK	ISAT.JK	
CTRA.JK	TLKM.JK	KAEF.JK	
EXCL.JK	UNVR.JK	LPKR.JK	
GGRM.JK		MEDC.JK	
INCO.JK		MNCN.JK	
INTP.JK		TINS.JK	
ITMG.JK			
JPFA.JK			
JSMR.JK			
LSIP.JK			
PGAS.JK			
PNBN.JK			
PNLF.JK			
PTBA.JK			
RALS.JK			
SMGR.JK			
SMRA.JK			
UNTR.JK			
WIKA.JK			

Table A.2.b Stocks List from Agglomerative Clusters in Model Construction

Cluster 0	Cluster 1	Cluster 2	Cluster 3
KAEF.JK	ANTM.JK	AKRA.JK	BBCA.JK
LSIP.JK	BMTR.JK	ASII.JK	BBRI.JK
MNCN.JK	DOID.JK	BMRI.JK	CPIN.JK
PGAS.JK	ELSA.JK	BRPT.JK	
PNBN.JK	INDY.JK	GGRM.JK	
PNLF.JK	ISAT.JK	ICBP.JK	
PTBA.JK	LPKR.JK	INDF.JK	
RALS.JK	MEDC.JK	INKP.JK	
SMGR.JK		JSMR.JK	
SMRA.JK		KLBF.JK	
TINS.JK		MAPI.JK	
UNTR.JK		PWON.JK	
WIKA.JK		TLKM.JK	
AALI.JK		UNVR.JK	
ADHI.JK			
ADRO.JK			
ASRI.JK			
BBTN.JK			
BDMN.JK			
BSDE.JK			
CTRA.JK			
EXCL.JK			
GJTL.JK			
INCO.JK			
INTP.JK			
ITMG.JK			
JPFA.JK			

Table A.2.c Random Portfolios Stocks List for Model Construction

Random 5	Random 10	Random 15
INKP.JK	SMRA.JK	TLKM.JK
DOID.JK	EXCL.JK	UNVR.JK
MNCN.JK	PGAS.JK	UNTR.JK
BMRI.JK	ASII.JK	BSDE.JK
ISAT.JK	LSIP.JK	BBTN.JK
	KAEF.JK	BRPT.JK
	BRPT.JK	PNBN.JK
	BMTR.JK	EXCL.JK
	BBCA.JK	ADRO.JK
	KLBF.JK	INDY.JK
		MEDC.JK
		BBCA.JK
		PTBA.JK
		CPIN.JK
		BMTR.JK

Table A.2.d Stocks List from k-means Clusters in Model Validation

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
ADHI.JK	BBCA.JK	AKRA.JK	ANTM.JK	AALI.JK	ADRO.JK
ASRI.JK	BBRI.JK	ASII.JK	DOID.JK	BRPT.JK	BBTN.JK
BDMN.JK	CPIN.JK	BMRI.JK	ELSA.JK	MNCN.JK	BMTR.JK
INKP.JK	ICBP.JK	GGRM.JK	GJTL.JK	PGAS.JK	BSDE.JK
ISAT.JK	INDF.JK	JSMR.JK	INDY.JK		CTRA.JK
ITMG.JK	KLBF.JK	PWON.JK	MEDC.JK		EXCL.JK
KAEF.JK	MAPI.JK	SMRA.JK			INCO.JK
PNLF.JK		TLKM.JK			INTP.JK
RALS.JK		UNVR.JK			JPFA.JK
TINS.JK		WIKA.JK			LPKR.JK
UNTR.JK					LSIP.JK
					PNBN.JK
					PTBA.JK
					SMGR.JK

Table A.2.e Stocks List from Agglomerative Clusters in Model Validation

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
ANTM.JK	ADRO.JK	AALI.JK	BBCA.JK	ADHI.JK	ASII.JK
DOID.JK	AKRA.JK	BRPT.JK	BBRI.JK	ASRI.JK	BMRI.JK
ELSA.JK	BBTN.JK	CPIN.JK		BDMN.JK	ICBP.JK
INDY.JK	BSDE.JK	EXCL.JK		BMTR.JK	INDF.JK
INKP.JK	CTRA.JK	LPKR.JK		GJTL.JK	KLBF.JK
ISAT.JK	GGRM.JK	MAPI.JK		INCO.JK	TLKM.JK
KAEF.JK	INTP.JK	MNCN.JK		ITMG.JK	
MEDC.JK	JPFA.JK			LSIP.JK	
TINS.JK	JSMR.JK			PNBN.JK	
	PGAS.JK			PNLF.JK	
	PTBA.JK			RALS.JK	
	PWON.JK				
	SMGR.JK				
	SMRA.JK				
	UNTR.JK				
	UNVR.JK				
	WIKA.JK				

Table A.2.f Random Portfolios Stocks List for Model Validation

Random 5	Random 10	Random 15
ADHI.JK	EXCL.JK	ASII.JK
BBCA.JK	ADHI.JK	BDMN.JK
BBRI.JK	INTP.JK	INDY.JK
CTRA.JK	ADRO.JK	PNLF.JK
EXCL.JK	KLBF.JK	BBRI.JK
	INKP.JK	KAEF.JK
	ASII.JK	TINS.JK
	AKRA.JK	BBTN.JK
	LSIP.JK	ASRI.JK
	INDY.JK	BBCA.JK
		JSMR.JK
		INDF.JK
		SMRA.JK
		INKP.JK
		LSIP.JK

Appendix 3. Portfolio Weights in Model Construction

Table A.3.a MVO Random Portfolios Weights

Random 5	Weights	Random 10	Weights	Random 15	Weights
INKP.JK	81.50%	SMRA.JK	3.30%	TLKM.JK	1.20%
DOID.JK	1.60%	EXCL.JK	3.00%	UNVR.JK	14.20%
MNCN.JK	0.10%	PGAS.JK	1.80%	UNTR.JK	11.00%
BMRI.JK	16.30%	ASII.JK	15.80%	BSDE.JK	1.30%
ISAT.JK	0.60%	LSIP.JK	0.20%	BBTN.JK	5.30%
		KAEF.JK	1.80%	BRPT.JK	15.60%
		BRPT.JK	26.30%	PNBN.JK	0.80%
		BMTR.JK	0.60%	EXCL.JK	0.10%
		BBCA.JK	29.20%	ADRO.JK	1.30%
		KLBF.JK	18.00%	INDY.JK	3.20%
				MEDC.JK	0.20%
				BBCA.JK	13.60%
				PTBA.JK	13.80%
				CPIN.JK	18.00%
				BMTR.JK	0.20%

Table A.3.b MVO k-means Portfolios Weights

K-means Cluster 0	Weights	K-means Cluster 1	Weights	K-means Cluster 2	Weights	K-means Cluster 3	Weights
ADHI.JK	2.70%	ASII.JK	0.90%	ANTM.JK	7.40%	AALI.JK	9.80%
ADRO.JK	1.90%	BBCA.JK	29.80%	BMTR.JK	4.10%	BRPT.JK	27.60%
AKRA.JK	2.30%	BBRI.JK	19.50%	DOID.JK	43.20%	EXCL.JK	0.70%
ASRI.JK	10.50%	BMRI.JK	3.90%	ELSA.JK	38.70%	GJTL.JK	6.00%
BBTN.JK	1.70%	ICBP.JK	28.10%	INDY.JK	0.60%	INCO.JK	2.80%
BDMN.JK	4.40%	INDF.JK	11.90%	ISAT.JK	1.80%	KAEF.JK	0.60%
BSDE.JK	1.00%	JSMR.JK	0.20%	LPKR.JK	0.70%	LSIP.JK	0.20%
CPIN.JK	16.40%	KLBF.JK	4.90%	MEDC.JK	3.30%	MNCN.JK	0.80%
CTRA.JK	3.30%	TLKM.JK	0.10%			PGAS.JK	2.40%
GGRM.JK	12.00%	UNVR.JK	0.80%			PNBN.JK	2.20%
INKP.JK	11.40%					PNLF.JK	23.20%
INTP.JK	5.60%					RALS.JK	18.70%
ITMG.JK	2.20%					TINS.JK	5.00%
JPFA.JK	1.80%						
MAPI.JK	2.20%						
PTBA.JK	7.70%						
PWON.JK	2.70%						
SMGR.JK	5.80%						
SMRA.JK	0.10%						
UNTR.JK	0.30%						
WIKA.JK	4.10%						

Table A.3.c MVO Agglomerative Portfolios Weights

Agglom. Cluster 0	Weights	Agglom.Cluster 1	Weights	Agglom. Cluster 2	Weights	Agglom. Cluster 3	Weights
AALI.JK	1.40%	ANTM.JK	10.60%	AKRA.JK	3.30%	BBCA.JK	12.70%
ADHI.JK	3.00%	BMTR.JK	1.50%	ASII.JK	6.20%	BBRI.JK	7.60%
ADRO.JK	7.30%	DOID.JK	35.90%	BMRI.JK	0.60%	CPIN.JK	79.80%
ASRI.JK	2.00%	ELSA.JK	41.50%	BRPT.JK	11.00%		
BBTN.JK	0.50%	INDY.JK	7.00%	GGRM.JK	8.60%		
BDMN.JK	9.40%	ISAT.JK	0.60%	ICBP.JK	13.20%		
CTRA.JK	7.00%	LPKR.JK	0.30%	INDF.JK	22.60%		
EXCL.JK	1.40%	MEDC.JK	2.60%	INKP.JK	21.00%		
GJTL.JK	3.00%			JSMR.JK	1.00%		
INCO.JK	3.00%			KLBF.JK	4.70%		
INTP.JK	5.20%			MAPI.JK	6.60%		
ITMG.JK	6.00%			PWON.JK	0.70%		
JPFA.JK	9.20%			TLKM.JK	0.00%		
KAEF.JK	1.60%			UNVR.JK	0.50%		
LSIP.JK	1.10%						
MNCN.JK	1.00%						
PGAS.JK	1.00%						
PNBN.JK	0.20%						
PNLF.JK	7.90%						
PTBA.JK	8.70%						
RALS.JK	4.10%						
SMGR.JK	2.80%						
SMRA.JK	3.40%						
TINS.JK	2.10%						
UNTR.JK	2.20%						
WIKA.JK	5.60%						

Table A.3.d HRP Random Portfolios Weights

Random 5	Weights	Random 10	Weights	Random 15	Weights
ISAT.JK	35.45%	KAEF.JK	14.20%	PNBN.JK	11.64%
MNCN.JK	23.14%	LSIP.JK	11.47%	BMTR.JK	7.01%
INKP.JK	12.00%	BMTR.JK	8.54%	BRPT.JK	7.48%
DOID.JK	8.44%	PGAS.JK	6.34%	PTBA.JK	5.91%
BMRI.JK	20.98%	BRPT.JK	7.85%	CPIN.JK	5.47%
		EXCL.JK	5.57%	UNTR.JK	6.60%
		BBCA.JK	23.31%	ADRO.JK	5.37%
		SMRA.JK	5.58%	INDY.JK	3.08%
		ASII.JK	9.57%	MEDC.JK	3.25%
		KLBF.JK	7.57%	EXCL.JK	4.30%
				BBCA.JK	18.02%
				TLKM.JK	6.80%
				UNVR.JK	7.56%
				BSDE.JK	4.12%
				BBTN.JK	3.39%

Table A.3.e HRP k-means Portfolios Weights

k-means Cluster 0	Weights	k-means Cluster 1	Weights	k-means Cluster 2	Weights	k-means Cluster 3	Weights
INKP.JK	4.51%	KLBF.JK	9.12%	BMTR.JK	12.01%	MNCN.JK	9.74%
MAPI.JK	8.26%	JSMR.JK	12.35%	ANTM.JK	14.67%	KAEF.JK	10.21%
PTBA.JK	6.51%	ICBP.JK	16.84%	ELSA.JK	8.44%	BRPT.JK	7.26%
BDMN.JK	8.98%	ASII.JK	8.96%	MEDC.JK	6.52%	GJTL.JK	9.14%
CPIN.JK	4.83%	BBCA.JK	15.73%	DOID.JK	6.83%	EXCL.JK	4.28%
JPFA.JK	4.69%	TLKM.JK	8.87%	INDY.JK	7.19%	PGAS.JK	4.76%
GGRM.JK	10.79%	BBRI.JK	7.93%	ISAT.JK	23.58%	AALI.JK	9.91%
PWON.JK	5.36%	BMRI.JK	7.20%	LPKR.JK	20.75%	LSIP.JK	6.39%
ADRO.JK	3.88%	INDF.JK	6.25%			INCO.JK	4.11%
ITMG.JK	3.55%	UNVR.JK	6.75%			TINS.JK	9.95%
SMGR.JK	4.74%					RALS.JK	6.98%
INTP.JK	3.51%					PNBN.JK	8.93%
UNTR.JK	5.70%					PNLF.JK	8.33%
AKRA.JK	4.38%						
ADHI.JK	4.17%						
WIKA.JK	3.13%						
ASRI.JK	3.88%						
BBTN.JK	2.55%						
SMRA.JK	2.55%						
BSDE.JK	2.41%						
CTRA.JK	1.63%						

Table A.3.f HRP Agglomerative Portfolios Weights

Agglom. Cluster 0	Weights	Agglom. Cluster 1	Weights	Agglom. Cluster 2	Weights	Agglom. Cluster 3	Weights
BDMN.JK	8.94%	BMTR.JK	12.01%	BRPT.JK	7.58%	CPIN.JK	21.70%
MNCN.JK	5.83%	ANTM.JK	14.67%	MAPI.JK	8.05%	BBCA.JK	52.20%
KAEF.JK	7.22%	ELSA.JK	8.44%	INKP.JK	4.39%	BBRI.JK	26.11%
AALI.JK	6.06%	MEDC.JK	6.52%	AKRA.JK	6.22%		
LSIP.JK	4.17%	DOID.JK	6.83%	PWON.JK	4.86%		
GJTL.JK	4.47%	INDY.JK	7.19%	GGRM.JK	9.30%		
PGAS.JK	3.18%	ISAT.JK	23.58%	KLBF.JK	6.84%		
EXCL.JK	2.69%	LPKR.JK	20.75%	JSMR.JK	11.11%		
PTBA.JK	3.53%			ICBP.JK	10.83%		
JPFA.JK	2.44%			INDF.JK	7.44%		
INCO.JK	2.21%			ASII.JK	6.18%		
TINS.JK	3.80%			UNVR.JK	6.74%		
ADHI.JK	3.42%			BMRI.JK	4.75%		
WIKA.JK	2.56%			TLKM.JK	5.70%		
BBTN.JK	2.09%						
ASRI.JK	3.17%						
SMRA.JK	1.35%						
BSDE.JK	1.96%						
CTRA.JK	1.52%						
SMGR.JK	2.23%						
INTP.JK	3.50%						
UNTR.JK	3.72%						
ADRO.JK	3.03%						
ITMG.JK	3.73%						
RALS.JK	3.89%						
PNBN.JK	4.81%						
PNLF.JK	4.49%						

Appendix 4. Portfolio Weights in Model Validation

Table A.4.a MVO Random Portfolios Weights in Model Validation

Random 5	Weights	Random 10	Weights	Random 15	Weights
ADHI.JK	2.40%	EXCL.JK	0.90%	PWON.JK	1.10%
BBCA.JK	83.30%	ADHI.JK	6.50%	WIKA.JK	3.10%
BBRI.JK	11.30%	INTP.JK	1.50%	BBTN.JK	0.30%
CTRA.JK	1.10%	ADRO.JK	29.50%	BSDE.JK	3.10%
EXCL.JK	2.00%	KLBF.JK	30.20%	UNTR.JK	2.30%
		INKP.JK	0.30%	PNLF.JK	7.50%
		ASII.JK	14.00%	INDF.JK	5.80%
		AKRA.JK	0.40%	MEDC.JK	1.20%
		LSIP.JK	15.70%	ADRO.JK	15.90%
		INDY.JK	1.10%	EXCL.JK	18.40%
				ISAT.JK	0.60%
				PGAS.JK	18.50%
				INCO.JK	18.20%
				AKRA.JK	2.80%
				ELSA.JK	1.30%

Table A.4.b MVO Optimal Clusters Portfolios Weights in Model Validation

k-means Cluster 3	Weights	Agglom. Cluster 0	Weights
ANTM.JK	11.60%	ANTM.JK	20.40%
DOID.JK	1.00%	DOID.JK	2.00%
ELSA.JK	5.10%	ELSA.JK	32.10%
GJTL.JK	0.20%	INDY.JK	3.40%
INDY.JK	3.90%	INKP.JK	4.90%
MEDC.JK	78.20%	ISAT.JK	0.50%
		KAEF.JK	0.90%
		MEDC.JK	34.50%
		TINS.JK	1.40%

Table A.4.c HRP Random Portfolios Weights in Model Validation

Random 5	Weights	Random 10	Weights	Random 15	Weights
EXCL.JK	11.17%	LSIP.JK	11.17%	PNLF.JK	10.54%
BBCA.JK	54.17%	AKRA.JK	15.10%	AKRA.JK	11.35%
BBRI.JK	20.27%	EXCL.JK	10.22%	EXCL.JK	8.47%
ADHI.JK	9.08%	INTP.JK	8.55%	ISAT.JK	4.38%
CTRA.JK	5.30%	KLBF.JK	15.07%	UNTR.JK	8.01%
		INKP.JK	4.72%	ADRO.JK	4.41%
		ADHI.JK	11.52%	MEDC.JK	4.64%
		ASII.JK	15.71%	ELSA.JK	6.33%
		ADRO.JK	4.20%	INCO.JK	4.29%
		INDY.JK	3.75%	PWON.JK	4.75%
				BSDE.JK	5.35%
				WIKA.JK	4.23%
				BBTN.JK	5.55%
				INDF.JK	12.12%
				PGAS.JK	5.57%

Table A.4.d HRP Optimal Clusters Portfolios Weights in Model Validation

k-means Cluster 3	Weights	Agglom. Cluster 0	Weights
GJTL.JK	30.45%	KAEF.JK	13.33%
ANTM.JK	15.61%	ISAT.JK	13.83%
DOID.JK	9.72%	INKP.JK	8.92%
INDY.JK	14.42%	ANTM.JK	15.70%
ELSA.JK	17.47%	TINS.JK	11.15%
MEDC.JK	12.32%	DOID.JK	8.14%
		INDY.JK	9.44%
		ELSA.JK	11.43%
		MEDC.JK	8.06%

Appendix 5. Efficient Frontiers of Model Validation MVO Portfolios

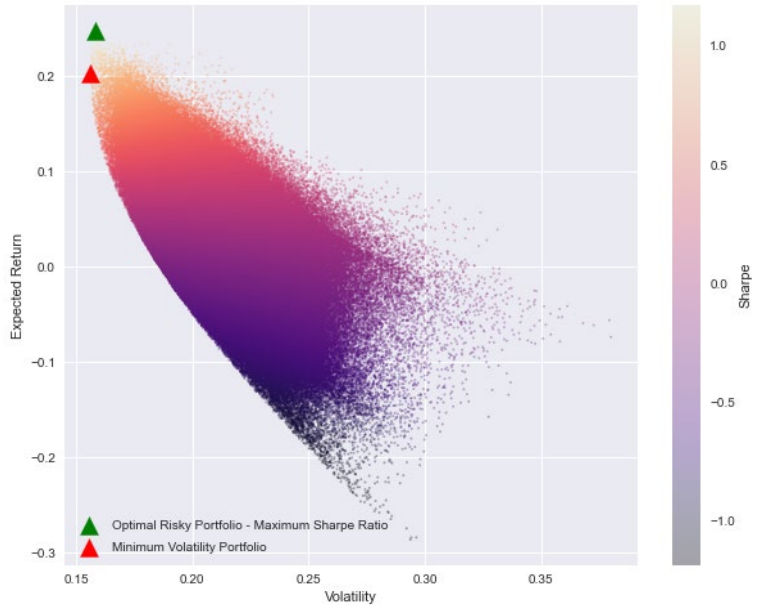

Figure A.5.a Random 5 Portfolio Efficient Frontier

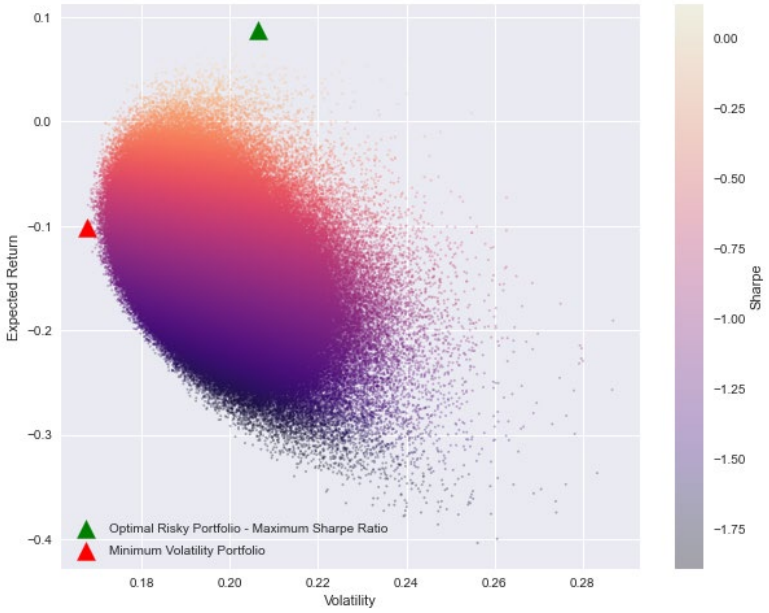

Figure A.5.b Random 10 Portfolio Efficient Frontier

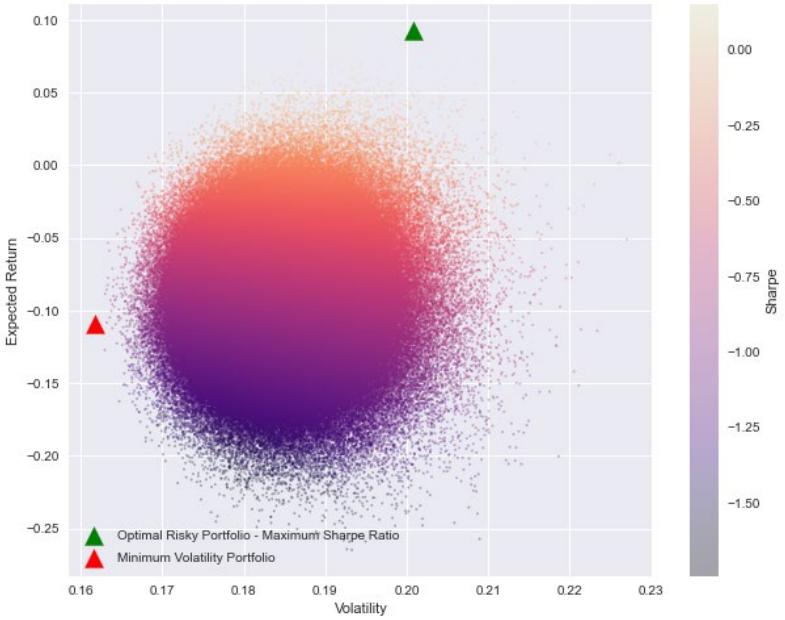

Figure A.5.c Random 15 Portfolio Efficient Frontier

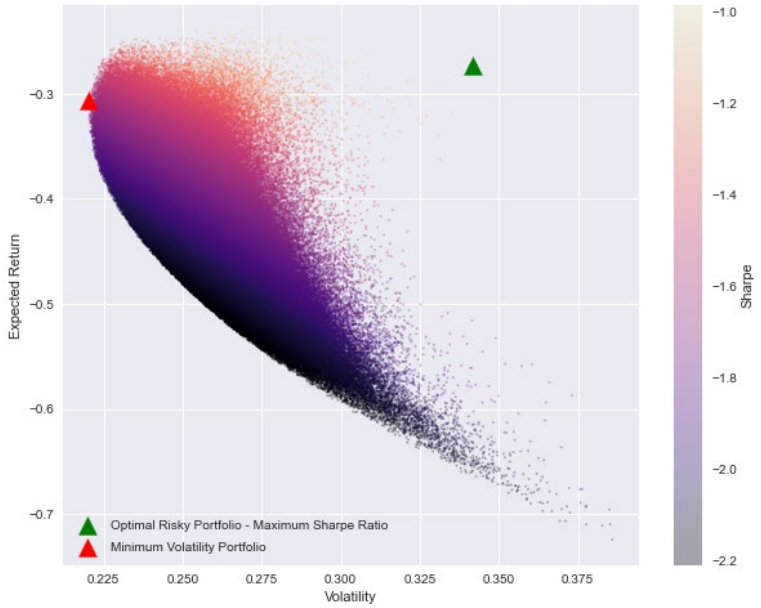

Figure A.5.d k-means Cluster 3 Portfolio Efficient Frontier

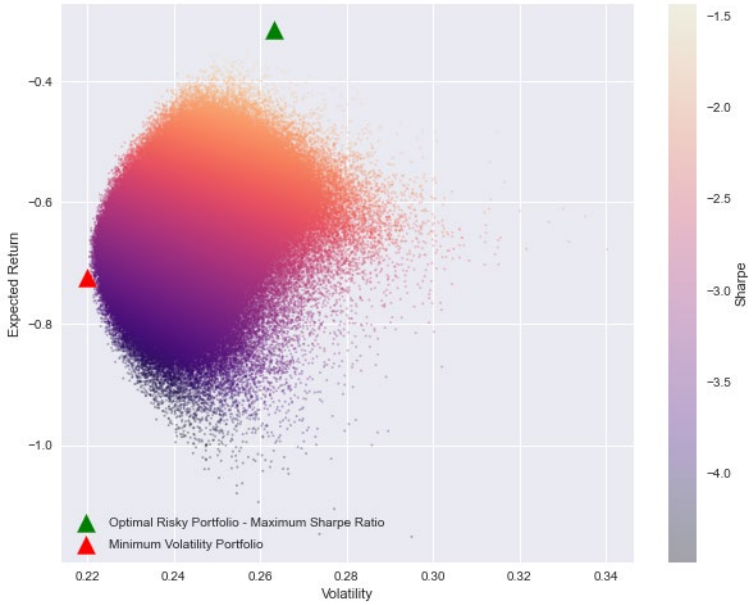

Figure A.5.e Agglomerative Cluster 0 Portfolio Efficient Frontier

Appendix 6. Return Distributions and Density of Optimal Portfolios

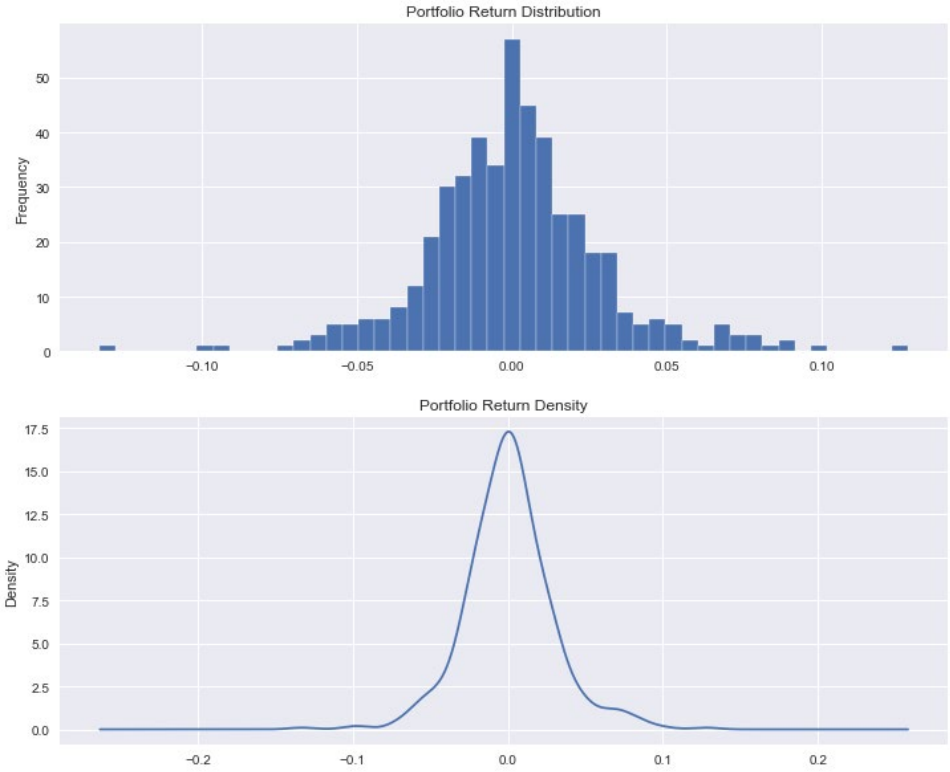

Figure A.6.a k-means Cluster 3 with HRP Weights Portfolio Returns Distribution and Density

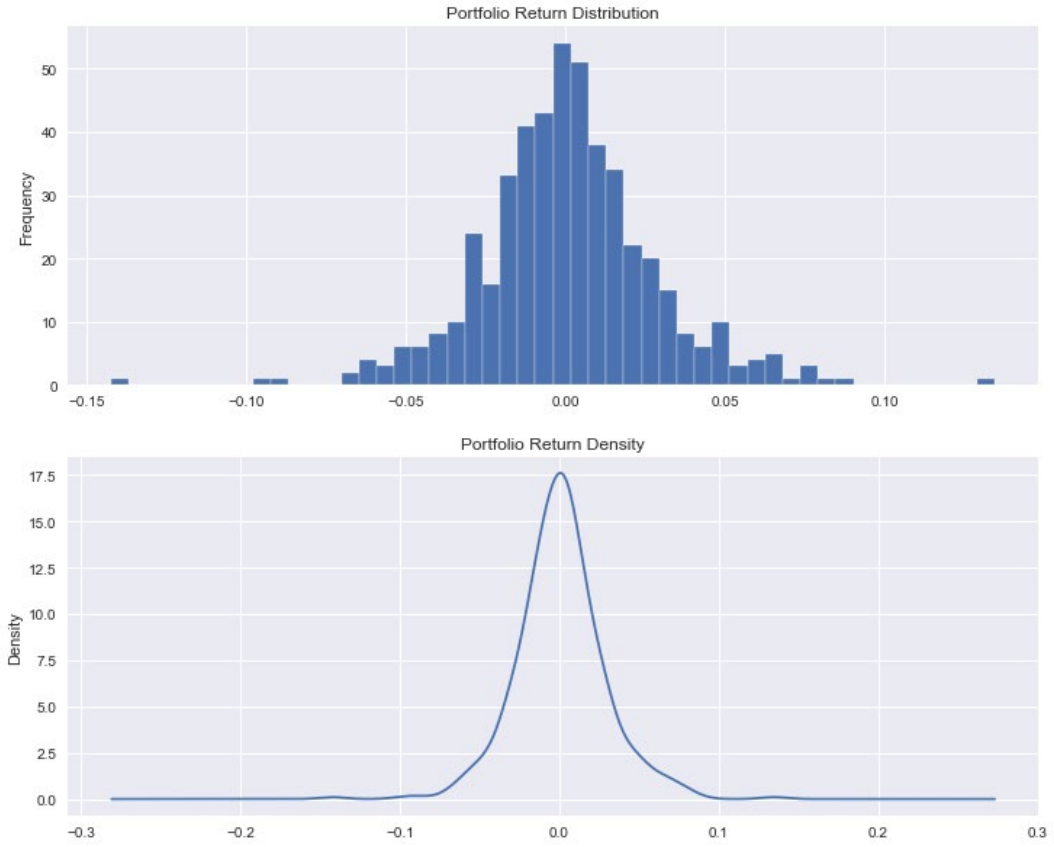

Figure A.6.b Agglomerative Cluster 0 with HRP Weights Portfolio Returns Distribution and Density

Appendix 7. Optimal Portfolios Figures

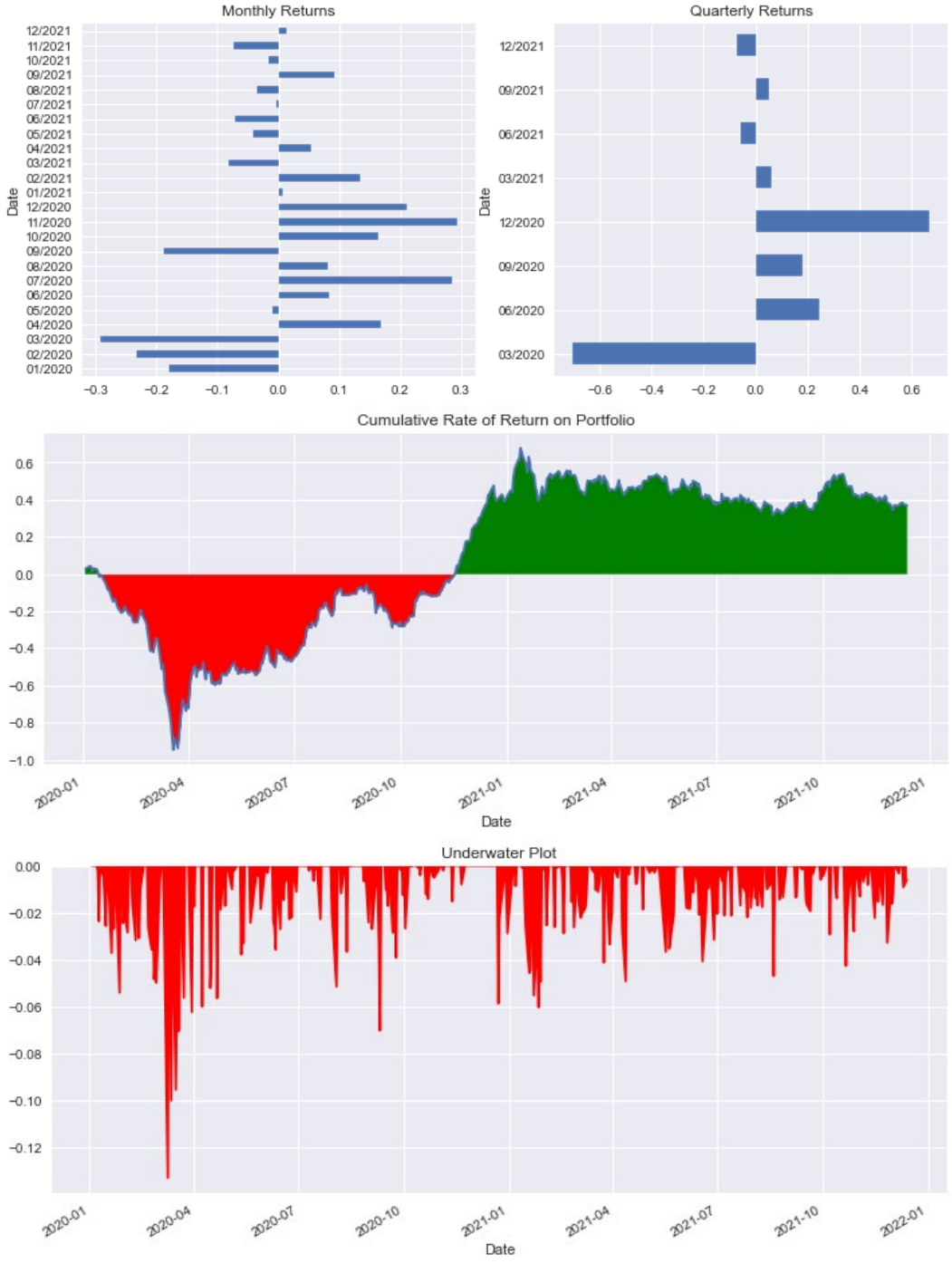

Figure A.7.a k-means Cluster 3 with HRP Weights Portfolio Monthly and Quarterly Returns (top) Cumulative Returns (middle) Underwater/Drawdown Plot (bottom)

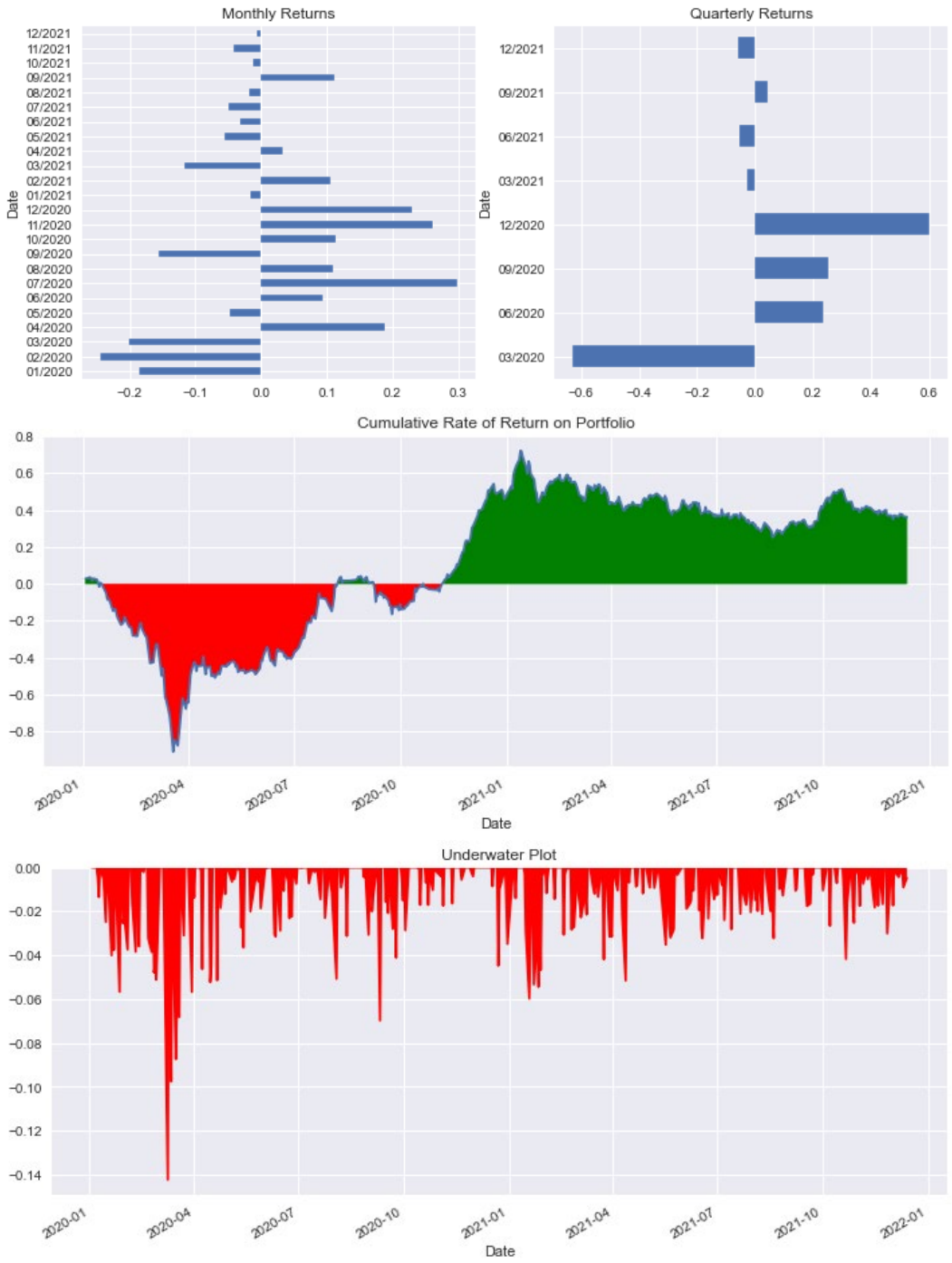

Figure A.7.b Agglomerative Cluster 0 with HRP Weights Portfolio Monthly and Quarterly Returns (top) Cumulative Returns (middle) Underwater/Drawdown Plot (bottom)