Lisbon School of Economics & Management
Universidade de Lisboa

MASTER OF SCIENCE IN
FINANCE

MASTER´S FINAL WORK
PROJECT

MACHINE LEARNING APPLICATIONS IN PORTFOLIO MANAGEMENT THEORY

MARCO NERI

July – 2023

Lisbon School of Economics & Management
Universidade de Lisboa

MASTER OF SCIENCE IN
FINANCE

MASTER´S FINAL WORK
PROJECT

MACHINE LEARNING APPLICATIONS IN PORTFOLIO MANAGEMENT THEORY

MARCO NERI

SUPERVISION:
PROF. PEDRO RINO VIEIRA

July - 2023

I am grateful to my parents
for investing and believing in
me during the entire period
of my studies.

GLOSSARY

AI – Artificial Intelligence.

ARIMA – Auto-Regressive Integrated Moving Average.

CAPM – Capital Asset Pricing Model.

CVaR – Conditional Value-at-Risk.

MAE – Mean Absolute Error.

MDD – Maximum Drawdown.

ML – Machine Learning.

MPT – Modern Portfolio Theory.

S&P 500 – Standards and Poor 500.

SARIMAX – Seasonal Auto-Regressive Integrated Moving Average Exogenous.

VaR – Value-at-Risk

ABSTRACT

Portfolio management, being the practice of managing and selecting an investment strategy and allocation for a defined investor, has always aimed at maximizing return while minimizing the risk of a combination of financial securities, hence a portfolio. The financial world has been evolving since Markovitz introduced the *modern portfolio theory* (MPT) in 1952, although nowadays it is still widely addressed as the benchmark and foundation for optimization methods. Traditional techniques of portfolio allocation such as MPT were considered without flaws for many decades, however its implications and notions were utilized to create enhanced several newer theories over the years, such as *capital asset pricing theory* (CAPM), *arbitrage pricing theory* (APT) and many others.

The technologic advancement introduced computing power and Artificial Intelligence (AI) techniques into the industry, creating the possibility of handling large and complex datasets through instructed algorithms.

The scope of this analysis was to employ the oldest and most popular approach such as MPT in combination with the Monte-Carlo method, a stochastic model to simulate random portfolio, and create an investment strategy based on these assumptions. Machine Learning (ML) models were then applied to analyse their impact on the previous strategy. Specifically, a clustering algorithm was implemented to reach a high level of diversification, while an auto-regression model, such as ARIMA, aimed at predicting future stock prices. The project utilized historical data to compute the analysis and each strategy was back-tested over four years to evaluate their accuracy and performance and compared with a benchmark index, *Standards and Poor* (S&P 500) in this case.

The results of the machine learning-based techniques showed a higher performance compared to the index benchmark, indicating a well-diversified portfolio due to the clustering algorithm and an acceptable level of accuracy for the ARIMA model. The portfolio randomly constructed displayed the lowest performance out of all the strategies and the benchmark index, since the stocks selection did not provide a high degree of diversification.

KEYWORDS: MPT; diversification; Machine Learning; Monte-Carlo, ARIMA.

JEL Codes: C1, C6, C8, G1, G14, G15

RESUMO

A gestão de carteiras, sendo a prática de gerir e selecionar uma estratégia de investimento e alocação para um investidor definido, sempre teve como objetivo maximizar o retorno, minimizando o risco de uma combinação de títulos financeiros, portanto de uma carteira de investimento. Embora o mundo financeiro tenha evoluindo desde que Markovitz introduziu a Moderna Teoria da Carteira (MPT) em 1952, ainda hoje este modelo é a referência para os métodos de otimização das carteiras. As técnicas tradicionais de alocação de ativos, como o MPT, foram consideradas durante várias décadas. Não obstante, a partir da MPT, surgiram outras teorias, tal como o Capital Asset Pricing Model (CAPM) e a Arbitrage Pricing Theory (APT), entre várias outras. Por outro lado, o avanço tecnológico trouxe poder computacional e técnicas de Inteligência Artificial (IA) para a indústria, criando a possibilidade de tratar grandes e complexos conjuntos de dados através de algoritmos de IA.

No âmbito desta análise, empregou-se a abordagem mais clássica da MPT em combinação com Monte-Carlo, um modelo estocástico para simular o comportamento do valor dos títulos e da própria carteira, de modo a criar uma estratégia de investimento com pressupostos próprios. Modelos de Machine Learning (ML) foram igualmente aplicados para analisar o seu impacto na estratégia anterior. Especificamente, foi implementado um algoritmo de agrupamento para atingir um nível de diversificação elevado. Em simultâneo, recorreu-se a um modelo de auto-regressão, ARIMA, para prever os preços futuros das ações. Foram utilizados dados históricos na implementação de cada estratégia, cada uma testada ao longo de quatro anos para avaliar sua precisão e desempenho, tendo sido comparada com um índice de referência, Standards and Poor (S&P 500) neste caso. Os resultados das técnicas baseadas em machine learning mostraram um desempenho superior em relação ao benchmark, indicando um portfólio bem diversificado, devido ao algoritmo de agrupamento, e um nível de precisão aceitável para o modelo ARIMA. A carteira construída aleatoriamente apresentou o menor desempenho entre todas as estratégias e o índice de referência, pois a seleção de ações não proporcionou alto grau de diversificação.

Palavras Chave: MPT, diversificação; Machine Learning; Monte-Carlo, ARIMA.

*C*ódigos JEL: C1, C6, C8, G1, G14, G15

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

During the first decades of portfolio management evolution, its approaches and methods were considered sophisticated and widely applied by investors, with MPT being the most relevant. Nowadays, after a substantial improvement in financial market understanding and implementations, these approaches can be defined inaccurate and with multiple biases.

An individual investor can produce and follow a financial strategy which is economically incorrect, indeed financial market agents usually make economic decisions based on their opinion and beliefs. Many of these judgments contradict financial principles due to average or poor level of financial knowledge. According to Pinelis and Ruppert (2022), the technological revolution had a wide impact on the financial world, contributing to a constant growth in portfolio management techniques and accessibility to complex strategies for individual investors. They analysed how machine learning-based models and the introduction of computing power enhanced and improved the general practices of stock selection, maximization of return, diversification and minimization of risk. Machine learning, being a subfield of Artificial Intelligence (AI), introduced the possibility of constructing algorithms capable of handling large dataset with complex structures and without much supervision. These algorithms can be employed as a rational assistant and instructed to manage a defined investment strategy, under pre-established constraints and objectives and without intervention, therefore increase efficiency and reduce the negative impact of bias inducted by human behaviour (Rezaei & Elmi, 2018).

Nowadays, individuals and professional investors are effectively creating strategies based on traditional assumptions with the combination of more recently formulated quantitative models, in order to discover and exploit discrepancy in the market while reducing exposure to risk. However, the relentless development in improving the computational performance of supercomputers has led the industry to an imbalance between individual and institutional investors, with the latest having the advantage of exploiting market's anomalies with high-frequency trading. However, also individual investors have the possibility of benefitting from implementations of structured and rational investment strategies. In today's market, the majority of trading volume is

produced by algorithms and computational systems instructed to take advantage of discrepancies and anomalies which are impossible to detach for an individual investor.

The core objective is to construct a diversified portfolio to outperform the index benchmark employed, S&P 500. Therefore, this project seeks to compare traditional approaches to recently developed methods and create an active portfolio management model with a defined number of selected stocks. The scope of the analysis is to create an automated machine-learning-based tool to simulate three different portfolio strategies and models with traditional optimization approaches, specifically the modern-portfolio-theory method was considered. More precisely, the first portfolio allocation was constructed with a Monte-Carlo simulation on randomly selected stocks, the second can be defined as a hybrid model since constructed with a combination of clustering, a machine-learning-based model employed to select optimal stocks for diversification, and Monte-Carlo simulation, while the last portfolio was created with the same clustering algorithm for the components' selection and ARIMA model for portfolio prediction and optimization, representing a model of solely machine-learning approaches and implications.

The structure of the project can be divided into 5 core sections and a conclusion chapter where the results of the model will be discussed. Sections include an overview of the topics covered from the project, a preliminary section for explaining and displaying data selection and the functioning of the algorithm, the core sections for the three different portfolio allocation strategies, and lastly a chapter where final results and conclusions are displayed and discussed.

The clustering algorithm produced a higher level of diversification during the testing periods than the randomly constructed portfolio, indicating an overall outperforming portfolio. After applying the ARIMA model to the stocks selected by the clustering algorithm, the three portfolios' performance were compared with the benchmark index S&P 500, showing a higher performance and a superior capacity to recover from drawdowns for the hybrid and machine learning-based portfolio.

2. LITERATURE REVIEW

2.1 Portfolio Management Overview

The foundation of portfolio management industry is based on traditional and relatively aged models, *Modern-Portfolio Theory* written by Harry Markovitz in 1952 being the most significant, and these are still considered relevant and being used as a benchmark in many asset allocation strategies. As Markowitz stated, the allocation of capital over alternative assets should consist of an optimized trade-off between return and risk of a security, which he quantified with expected return and variance, or standard deviation, respectively. Also, he claimed that by taking into consideration security returns and their associated movements over time, it could be possible to quantitatively estimate portfolio return and risk. The concept of diversification is a key concept in this scenario. Its foundation is the assumption that the risk measured for a specific portfolio is dictated not only by the level of risk of its individual assets but also by the correlations among its components (Kolm et al., 2014).

As stated in (Elton & Gruber, 1997), the construction of a portfolio should not depend merely on expected returns and variance, since a simplification of this nature excludes other essential moments and factors affecting a stock return's distribution.

The optimization problem was assessed by Markowitz through a simple but revolutionary decision-making process. Specifically, he formulated the mean-variance optimization which proposes that an investor, given a return objective, should choose the portfolio with the lowest variance out of the infinite sample of portfolio combinations, also called *Minimum Variance Portfolio*. Another optimal combination of weights that can be constructed is called *Maximum Sharpe Ratio Portfolio*, which should be chosen from an investor that aims to achieve the highest excess return, or risk premium, over the risk-free rate per unit of risk.

Given an unlimited weighted combination of assets, the so-called *Efficient Frontier* is the set of optimal portfolios that achieve the highest expected return for a given level of risk or the lowest amount of risk for a given level of expected return. All other portfolios that do not follow this rule are called "sub-optimal" or "inefficient", since they provide a lower expected return for a given level of risk, or variance.

Investment Portfolio Management in the last decades has seen its models and applications increasing exponentially in quantity and quality due to the advent of technology and its computational speed performance, since being directly connected with the technology advancement of financial techniques. In this context, financial engineering plays a crucial role for recent investment strategy models, especially if handling large and complex dataset is required. This requirement is satisfied by machine-learning-based systems which are able to adapt their properties and parameters to the dataset and overcome multiple obstacles encountered with traditional approaches (Idowu et al., 2022).

2.2 Machine Learning

Machine learning, being a subfield of Artificial Intelligence, seeks to detect relevant relationships and patterns from observations of a dataset. Developments in machine learning have contributed to the increasing number of artificially intelligent systems used in decision-making frameworks, employee management, assistant systems for customer's preferences and trading applications (Janiesch et al., 2021).

According to Goodell (2021), financial firms and institutions, from hedge funds and retail banks to financial technology (FinTech) firms, nowadays are aggressively allocating capital on development and acquisition of artificial intelligence and machine learning resources. The introduction of machine-learning based models, along with a constant growth in computing power and storage, into financial systems has led to major improvements in data-driven applications.

Machine Learning can be divided into two branches, supervised and unsupervised learning, which are defined as the internal methods and processing of the model. The principal difference between supervised and unsupervised learning is represented by the necessity of labelled training data as input for the first. Unsupervised learning employs unlabelled or raw data, whereas supervised learning requires labelled input and output training data. The model learns the relationship between the labelled input and output data in supervised machine learning. Models are then finetuned until they can predict the outcomes of previously undiscovered data. Labelled training data, however, are often time-consuming and require computing power to be developed. Whereas unsupervised machine learning learns from unlabelled raw training data. Unsupervised models are

widely employed to detect intrinsic trends and linkages in a given dataset since they are trained to identify links and patterns inside an unlabelled dataset (Love C. Bradley, 2002).

2.2.1 Clustering Overview

Clustering, often known as cluster analysis, is an unsupervised learning method of labelling elements based on similarities. Specifically, it can be defined as the process of identifying similar components in an unlabelled set of data in order to make it comprehensible and manipulable. It identifies subgroups in a given heterogeneous datasets and aims to create individual clusters with the highest possible homogeneity between the elements.

Clustering techniques can be divided into partitional, hierarchical, density-based, model-based, grid-based and soft-computing systems with the first two being the most popular methods (Rokach & Maimon, 2005). Partitional or centroid-based clustering is a clustering approach which focuses on discovering similar groups of elements and their relative central points, called *centroids*. These strategies are recognized as one of the most basic yet successful methods of building clusters. The premise behind centroid-based clustering is that the centroid describes a cluster, and data points that are closest to these vectors are assigned to the appropriate group (Gunopulos, 2009).

Gunopulos (2009) described also the process to construct a partition-based clustering algorithm, as the initial step is to create several clusters and assign random objects to each one. The centroid of each cluster can subsequently be computed. Precisely, the centroid of each cluster is calculated as a *medoid*, which is an item whose average dissimilarity to all the objects in the cluster is smallest. The algorithm then generates new clusters based upon their closeness to the medoid, and the new medoid is determined for the newly generated clusters. This is continued until the clusters stop changing. Amongst the partition clustering methods, the so-called K-Means is largely the most utilized.

Hierarchical clustering can be agglomerative or divisive, which either agglomerate smaller clusters into larger clusters or splits larger clusters into smaller clusters. agglomerate clustering first assigns each data point into its own cluster, and gradually

merges clusters until only one remains, while for the divisive clustering the process is reversed from one unique cluster to many.

2.2.2 Arima Overview

In the context of financial forecasting using time-series, one major model developed in the last 50 years is the so-called Box-Jenkins Model or the more popular name ARIMA model. The Box-Jenkins method was formulated by George Box and Gwilym Jenkins in the first edition of their textbook *Time Series Analysis: Forecasting and Control* published in 1970.

ARIMA, which stands for autoregressive integrated moving average model, is a form of regression analysis that aims to extract the core pattern from past data or predict future values of a time-series. Just as for linear and multiple regression, ARIMA can be defined as a supervised learning algorithm. The model is generally used for short-term forecasting of 18 months or less since is able to discover seasonality and trend patterns which are usually displayed over a length of a calendar year or more, depending on the input type of the time-series and the sector of interest (Ho S. L. & Xie, 1998).

The framework is a mathematical model which uses inputs from data of a time-series to forecast a defined data range in the future. It can be used to analyse a variety of sequenced data for forecasting purposes. According to Ho and Xie (1998), the main concept of the process is based on discovering and detaching disparities between data points to determine the outcome. Recognize seasonality and trend in a time-series, and analyse its stationarity enables the model to spot patterns over time, which provide consistency and strength to the system. Seasonality, usually displayed as regular and predictable patterns that repeat over a calendar year, suggests that exogenous factors affect the financial performance of a company and could negatively impact the regression model. Therefore, an assumed presence of trend and seasonality should be separated from the time-series, otherwise many of the computations throughout the process will provide non-accurate results. Certain auto-regressive models are able to discover seasonality patterns and create predictions including it as a factor in the process, such as *SARIMAX*. However, since stock prices do not show a specific seasonality cycle a traditional ARIMA model will be employed.

3. DATA AND METHODOLOGY

Python programming language was used to create a tool for the entirety of data pre-processing, analysis, plots, and prediction stages of the model. Yahoo Finance API was utilized to extract daily time-series with prices of all S&P 500 constituents, and the index itself used as a benchmark to compare and back-test the results of the project. Tables were created with Python during the analysis, and then formatted with Microsoft Excel. The period range considered was 2013-2022 as it includes both years of increasing stock prices and financial crisis. For the scope of the analysis, for each component of S&P 500 was selected its adjusted close price, since it reflects distribution of dividends, stock splits and new stock offerings apart of the actual closing price, therefore considered a more accurate indicator of a stock value. The data was then divided into two portions, train-set to develop the algorithm and test-set to back-test the performance generated from the model, which considers the first years (2013-2018) as the first training set and the next subsequent year (2019) as the first testing set. After the analysis was completed, the train-test split period was shifted by one year forward and the model re-trained and re-tested. This process was repeated until the last available year of data (2022) was selected as testing-test, as shown in Figure 1 and 2.

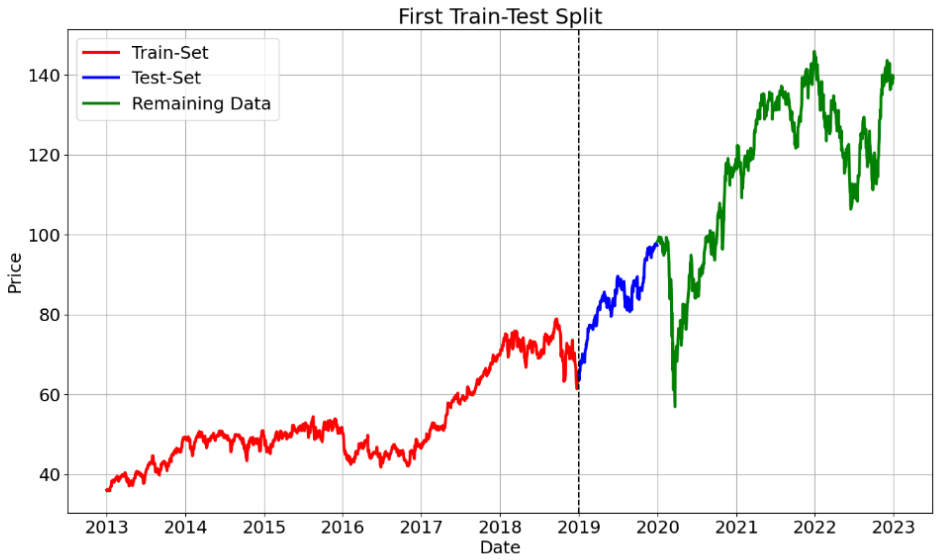

Figure 1. Train-Test Split until 2019

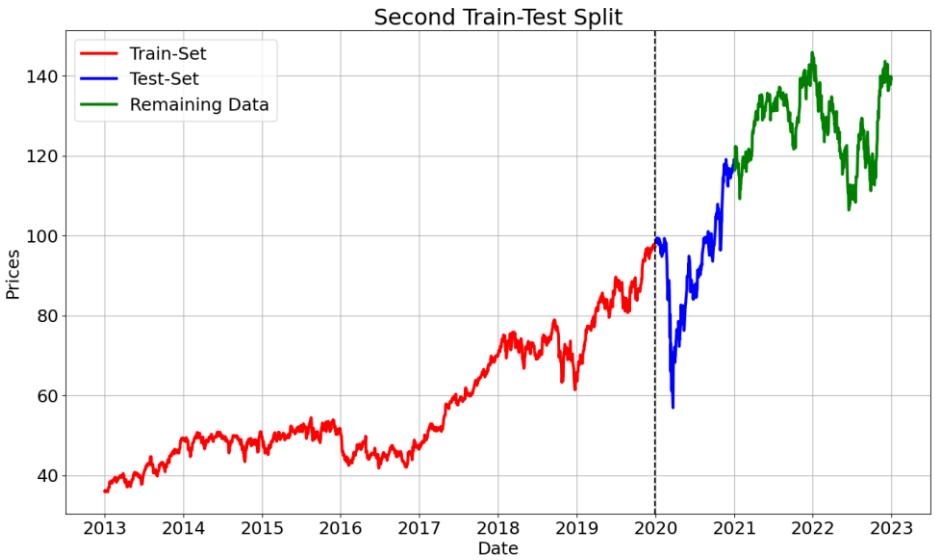

Figure 2. Train-Test Split until 2020

Given this approach, the algorithm tries to simulate a real-world investment strategy, while rebalancing and restructuring the portfolio at the end of each year after processing the new available data. The U.S. 1 year treasury bill was selected and considered the risk-free rate for the entire project, since each portfolio simulation had a duration of a single year. A screening procedure was conducted during each iteration of the model in order to reduce the number of available stocks and increase the algorithm efficiency.

The 2008 financial crisis, the Covid pandemic in 2020 and the war waged by Russia against Ukraine in 2022 demonstrated definitively that financial diversification cannot totally prevent major drawdowns (Nystrup et al., 2018). Diversification fails when it is required the most due to the increasing correlations between risky assets during periods of crisis. As a result, setting a reasonable maximum drawdown is crucial to the performance of any portfolio (Nystrup et al., 2019).

Considering the importance of capital preservation for a portfolio strategy during stock market crashes and downfalls, maximum drawdown (*MDD*) can be defined as the biggest observable drop in a stock price from a peak to a subsequent trough, as shown in the Equation (1).

$$(1) \qquad MDD = \frac{(P-L)}{P},$$

where *P* is the peak value before the largest drop in price and *L* is the lowest value before a new higher price is established. Therefore, all the stocks with lower MDD than the worst average between the considered years, in this case -43,05% in 2020 as shown in Table I, were excluded from the analysis and considered not ideal for diversification purposes. For simplicity, the algorithm was instructed to set the boundary at -40%, hence exclude stocks with a maximum drawdown lower than -40%.

TABLE I

AVERAGE MAXIMUM DRAWDOWN PER YEAR

Year	*MDD*
2019	-18,16
2020	-43,05
2021	-18,65
2022	-33,62

a) COMPUTED ON THE TOTAL COMPONENTS OF S&P 500
b) VALUES EXPRESSED IN PERCENTAGE

Moreover, stocks with a negative annualized *Sortino ratio* were also excluded since they would affect negatively the yearly total performance of a constructed portfolio. The more popular Sharpe ratio treats equally large upwards and downfall movements of a stock, since the standard deviation used for the computation gauges variances to both the upside and downside. However, investors are not indifferent to upside risk and downside risk whereas, in fact, the majority of investors are risk averse. More specifically, the ratio is considered 'penalizing positive volatility'. Differently the Sortino ratio measures the risk-adjusted performance using only the downside volatility of a stock, being measured by the standard deviation of negative or below-the-mean returns. Hence Sortino ratio was preferred to the Sharpe ratio for this step. Lastly, the volatility level of the stocks was considered. Since the early 2000s low volatility strategies have developed as a distinct and popular investment approach and the 2008 financial crisis demonstrated how this conservative technique diverges from traditional value investing, which, unlike low volatility, did not provide protection during the stock market collapse. Many recent studies indicate risk reduction levels of around 25% due to low volatility strategies (Van Vliet, 2018). Hence the stocks with a higher semi-deviation than 20% were excluded from the investment basket to reduce the total downside risk of the portfolios. The final result

reached after filtering the entire dataset of S&P 500 amounts of 95 total stocks to be considered for the investment strategy.

A machine learning based investment strategy was implemented and compared with a naïve strategy. For the scope of the project different portfolio management techniques to the given set of stocks and analyse the results and implications derived from. Three portfolio construction techniques were applied for each train-test iteration:

1. *Naïve strategy:* Monte-Carlo simulation and Mean-Variance optimization for a portfolio built with a random picking regime out of the 95 stocks. To improve diversification, one security was selected for each industry remained available after the screening process.
2. *Hybrid strategy:* Monte-Carlo simulation and mean-variance optimization for a portfolio built with a clustering algorithm, with the number of components being the same as the previous portfolio.
3. *Machine Learning strategy:* ARIMA prediction model applied to the set of selected stocks by the clustering algorithm and optimization through quadratic programming.

A major limitation of this approach can be explained by a popular sentence used in the financial world: *"Past success does not guarantee future performance"*. Hence, historical data of stock prices are a useful tool for portfolio management purposes, but they cannot be solely responsible for financial decisions due to numerous additional factors and assumptions to consider. Therefore, the created model could lead to inaccurate results which must not be interpreted as a sufficient indicator of an investment strategy.

4. CLUSTERING

According to Marvin and Bhatt (2015), a crucial objective and concern in portfolio management is how to create a combination of assets which have a certain degree of protection to downfalls of the market or to extreme events, such as the 2008 financial crisis or the start of the war in Ukraine in 2022. In order to obtain that, idiosyncratic risk exposure must be eliminated from a portfolio, which is defined by the specific risk associated with a certain asset, since market risk cannot be totally excluded from the investment equation. A technique that allows to reduce the idiosyncratic factor is diversification, which in this case consists in distributing the capital for an investment into assets with low correlation between each other, most preferably close to 0, and therefore create a portfolio that will not strictly follow a particular market trend. The application of this method will benefit the portfolio during downfalls of the market and will reduce slightly the upside during increasing market periods.

Both partitional and hierarchical clustering methods were employed on the stock's dataset, filtered with the constraints mentioned above during a preliminary phase of the project in order to select the most appropriate. Four methods were evaluated, specifically k-means was selected for the partitional approaches, while three different forms agglomerative hierarchical clustering were implemented.

The approach with the highest average Sharpe ratio between the constructed clusters, was preferred to the other methods.

4.1 K-Means

As previously mentioned, the most popular clustering method is k-means due to its simplicity in implementation and adaptability to datasets of various size.

The most used types of distance to be minimized used in K-Means Clustering are Euclidean and Minkowski distance, with the first being the most popular and employed in this analysis. The objective of the algorithm is to minimize the distance between data points and their relative cluster centroid, which is relocated at each iteration such as every observation belongs to the cluster with the nearest mean, hence the name K-Means (Renugadevi et al., 2016).

According to the Euclidean distance formula, the distance between two points in the Euclidean space with coordinates (x, y) and (a, b) is given by Equation 2:

$$(2) \qquad distance\ (x,y),(a,b) = \sqrt{(x-a)^2 + (y-b)^2},$$

Therefore, the objective is to minimize the sum of squares errors of each cluster which can be defined with Equation 3:

$$(3) \qquad argmin_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - u_i\|^2,$$

where k is the number of clusters, x are the observations, $S = S_1, S_2, \ldots, S_k$ are the sets of observations and u_i is the mean distance of the points in S_i (Marvin & Bhatt, 2015).

4.1.1 Elbow Method

As explained by Syakur, Khotimah, Rochman and Satoto (2018), to gauge the problem of selecting the optimal number of centroids while minimizing the sum of squares errors for each cluster, a reasonable and appropriate approach is employed. The so-called elbow method is a technique used in cluster analysis to determine the optimal number of clusters for a specific dataset. This method illustrates the value of the cost function generated by various k values, with k being the number of clusters. With the rise of k, the sum of squared distances from the centroids of their respective clusters to each data point decreases, as previously shown in the Equation 3, which can be called distortion of the model.

Each cluster has fewer constituent instances as its number increases, and the observations are closer to their respective centroids. However, the improvements in distortion diminish. The value of k at which the improvement in distortion decreases the most is recognized as the elbow, and this is the value where dividing the data into further clusters should cease (Humaira & Rasyidah, 2020).

For the scope of this analysis and selection between the clustering approaches, the elbow method was adopted for the screened dataset of stocks and the results then plotted, in order to provide the optimal number of clusters for the k-means approach.

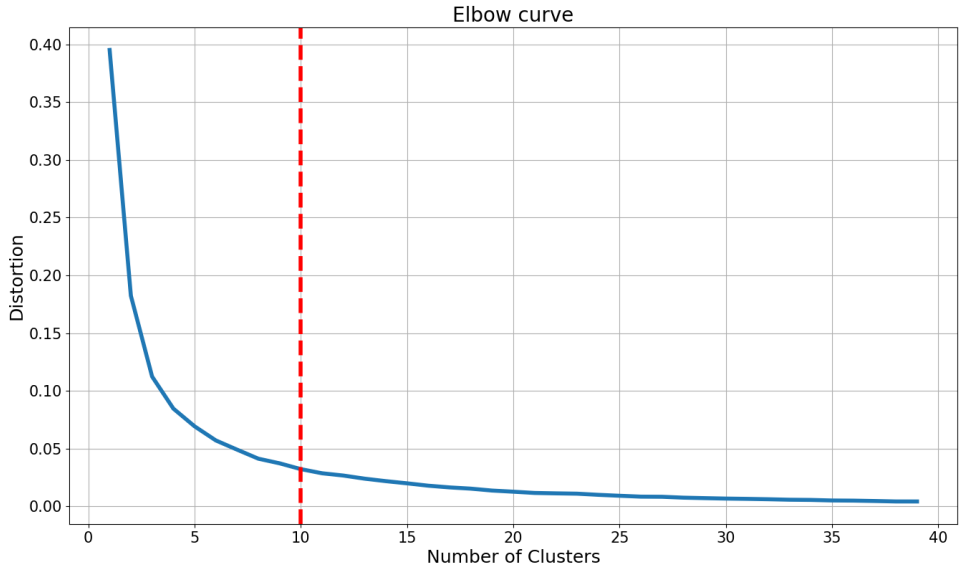

Figure 3. Elbow Method for K-Means Clustering

As shown in Figure 3, the optimal number of clusters for the specific dataset can be defined as 10, since is the value where the decrease in distortion begins to stabilize. The stock with the highest risk-adjusted return, hence Sharpe ratio, for each constructed cluster was selected, in order to create a portfolio with the same number of stocks as the one assembled randomly or in case all the industries were included in the specific train-test iteration, with a disparity of one stock.

After the elbow method analysis, the optimal number of clusters was implemented into the algorithm, which divided the datapoints having x-axis coordinates of annualized volatility and y-axis of annualized return. The results produced 10 different clusters, due to the elbow method, as shown below in Figure 4, and an implementation of a k-means algorithm on the dataset composed of all 500 stocks was completed. The partition into groups and their relative centroids were subsequentially plotted.

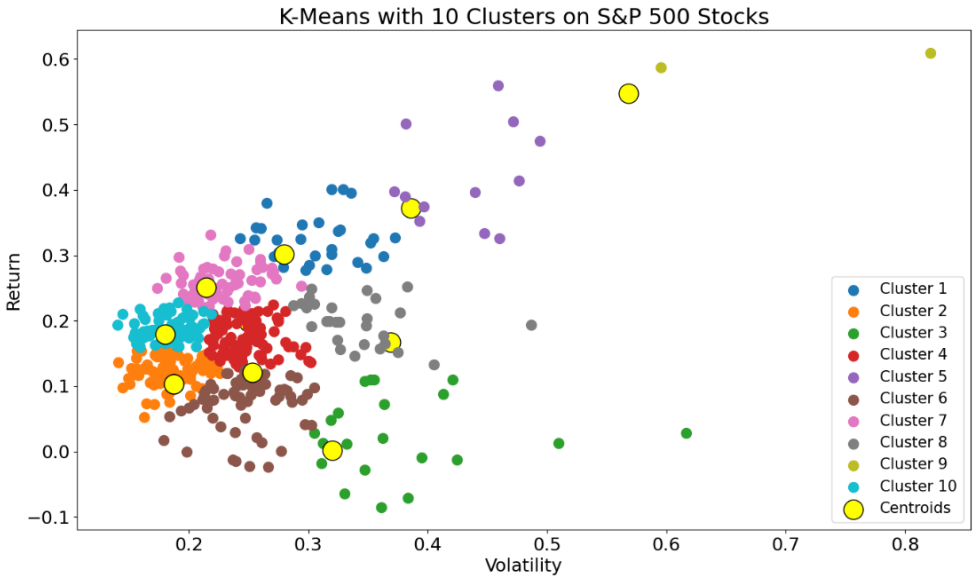

Figure 4. K-Means Plot with 10 Clusters

4.2 Hierarchical Clustering

Since the essence of the S&P 500 index is considered to be a composition of stocks instead of a single financial instrument, agglomerative hierarchical clustering was selected for the scope of the project as the process operates bottom up until a single unique cluster is formed. While for the k-means algorithm the traditional Euclidean distance was adopted in order to assign each datapoint to its cluster, in this case a different approach was employed. To pursue a portfolio with a high degree of diversification, agglomerative hierarchical clustering was based on low correlation between stocks, measured with the distance metric for correlation as shown in Equation 4 and 5, hence the least correlated securities having the highest affinity in the model (Renugadevi et al., 2016).

$$(4) \qquad \rho(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

$$(5) \qquad D_\rho = 1 - \rho,$$

Where ρ is the correlation between two stocks x and y, σ is their standard deviation, Cov is their covariance and D_ρ is the distance metric for correlation that will result in a distance matrix.

Using correlation increases the likelihood to perform better than using daily prices, since it takes into account not only similarities between prices but also similarities in trends of stock fluctuation.

The relationships between data points and clusters in hierarchical clustering can be displayed with a Dendrogram, which is a tree-like structure that explains the clustering system.

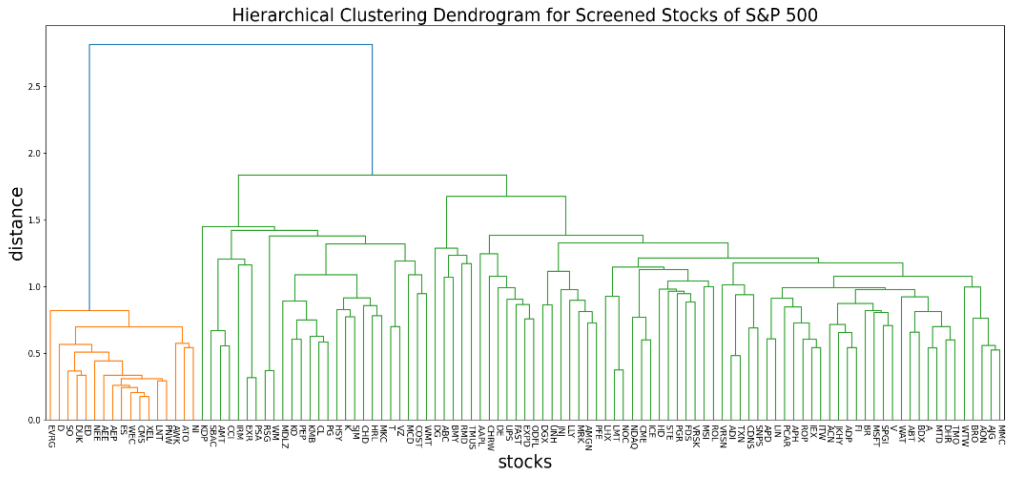

Figure 5. Hierarchical Clustering Structure

As shown in Figure 5, each cluster created in the first instance was then re-grouped into a bigger one using the same process as for the single datapoints, until a unique single cluster contained all the stocks.

Although one of the main advantages in using hierarchical clustering is the freedom provided to the model to learn from the dataset without pre-specify the clusters, for the scope of this analysis the algorithm was instructed to utilize the number of stocks selected in the portfolio with the random regime as the number of clusters. The reason can be explained by the need to create two comparable portfolios since constructed by the same number of components.

In order to compute the hierarchical agglomerative clustering, each sub-cluster must be linked and re-grouped into larger clusters. The different types of linkages methods describe the approaches to measure the distance between two given sub-clusters of data points. As explained by Li, Rezaeipanah and Tag El Din (2022), selection of the best

fitting linkage method is essential for a successful hierarchical cluster analysis. Types of linkages employed for this analysis are: single, complete and average.

Single linkage, also called nearest neighbour technique, partitions a dataset into groups such that the distance between clusters is defined as the distance between the closest pair of observations with each belonging to a different cluster, as shown in the Equation 6.

$$D(t,z) = Min\{d(i,j)\}, \quad (6)$$

where observation *i* belongs to cluster *t* and observation *j* to cluster *z*.

Complete linkage, also called farthest neighbour, is the opposite of single linkage since it considers the two most distant data points from different clusters, as shown in Equation 7 below.

$$D(t,z) = Max\{d(i,j)\}, \quad (7)$$

where the distance between clusters is defined as the distance between the farthest two observations belonging to two different groups.

Finally, in average linkage the distance between two sets of points is defined as the average of the distances between all the pair of observations belonging to both clusters, as shown in Equation 8.

$$D(t,z) = \frac{S_{tz}}{M_t \times M_z}, \quad (8)$$

where S_{tz} is the sum of all distances between pairs of data points of clusters *t* and *z*, while M_t and M_z are the number of constituents of each relative group.

4.2.1 Results

After each one of the four clustering techniques being tested and compared, agglomerative hierarchical was selected over k-means to pursue a higher degree of diversification for the portfolios, hence adopt the distance matrix of correlation as the algorithm's input.

According to Gagolewski, Bartoszuk and Cena (2016), the single linkage method exhibits sensitivity to noise and outliers, and since a dataset formed by historical data,

more precisely stock prices, presents a certain amount of statistical noise and outliers, this approach was excluded from the analysis.

Being both complete and average linkage less sensible to noise and outliers, the algorithms were implemented over the train-test periods and selected by their Sharpe ratio value as shown in Table II.

TABLE II

SHARPE RATIO COMPARISON BETWEEN COMPLETE AND AVERAGE LINKAGE

Year	*Complete*	*Average*
2019	**2,94**	2,55
2020	**0,89**	0,79
2021	1,73	**2,29**
2022	**0,22**	-0,03

Complete linkage displays a higher Sharpe ratio in three out of four train-test split periods. For the scope of the project, agglomerative hierarchical with complete linkage was therefore established as the clustering method going forward.

5. MONTE CARLO

In this phase of the project the selected stocks by the hierarchical clustering algorithm were utilized to simulate one-year future returns, which is the length of test split, through a stochastic process called Monte-Carlo.

Monte-Carlo approach can be defined as a Markov model, which is a stochastic method for randomly changing systems that possess the Markov property. The property states that, at any given time, the next state is only dependent on the current state and is independent of anything in the past (Van Ravenzwaaij et al., 2018).

According to Fama (1995; 1965), a stock market where successive, price changes in individual securities are independent is, by definition a random walk market. According to the random walk hypothesis, stock prices move randomly and are not affected by their history. As a consequence, using historical prices or fundamental analysis to forecast future trends or prices results impossible. If markets behaviour is indeed explained by a random factor, then all available information is reflected into markets prices. The random walk hypothesis also assumes that a stock price movement is independent, hence not correlated, to the price evolution of another security (Seymour Smidt, 1968).

However, stock market forecasting is defined more by its failures than by its triumphs because stock prices reflect investors' judgments and expectations based on the information available. If the movements are predicted to be positive or negative, the price changes so quickly that the agent possessing the news has little or no time to act upon it.

Monte-Carlo simulation was then implemented with modern portfolio theory and optimization by Markowitz, which aims to define and rank different portfolio combinations under few assumptions. Given *i* risky assets, optimal weights are defined as shown in Equation 9:

$$\sum_i w_i = 1, \tag{9}$$

as the sum of individual weights w_i of all portfolio's components must equal 1, hence no leverage or shorting is allowed. Moreover, the Equation 10 define the portfolio expected return *E(R)*:

$$E(R) = \sum_i w_i \mu_i = X^T \mu \tag{10}$$

where u_i is the mean return for each component and X^T represents the transpose vector of individual weights w_i. Below is shown the portfolio variance σ^2 in Equation 11:

$$(11) \qquad \sigma^2 = Var(R) = \sum_i \sum_j Cov(R_i, R_j) w_i w_j = X^T \sum X \,,$$

with R_i and R_j representing the returns for assets i and j respectively, and $\sum$ defining the covariance between the assets.

During this step, the performance of the portfolio constructed with the set of stocks selected by the clustering algorithm for each train-test split was simulated employing a Monte-Carlo algorithm over 1-year period. The number of simulations to be made was set to 500.000 during which each individual stock return and standard deviation were combined with randomly assigned weights, resulting in a dataset with 500.000 simulated portfolios with different expected returns, volatility and Sharpe ratio for the relative train period. Afterwards, the two optimal portfolios were extracted, minimum volatility and maximum Sharpe ratio, and their produced weights implemented into the test period to evaluate the expected portfolio performance.

A comparison was then made between the optimal hybrid portfolios and the same structured optimal portfolios without any constraints on components' selection. Moreover, the stocks were selected randomly from the screened dataset and a diversification degree level was attempt to be given by extracting one security from each industry amongst the 11 available. A Monte-Carlo simulation with the same previously set parameters was then performed on the created portfolio.

Each simulated strategy, utilizing clustered and randomly selected stocks respectively, was then implemented into the model while plotting efficient frontier and cumulative returns year by year, displaying the performance comparison between a machine learning based investment strategy with a random selected portfolio. The Monte-Carlo results for year 2021 were shown in Figure 6, which displays the efficient frontier, optimal portfolios and individual components for the clustering strategy. Moreover, Table III provides a comparison between the hybrid portfolio and the randomly created one, while analysing the core feature statistics. An equally weighted portfolio was included into the analysis to consider an allocation strategy with a certain degree of indifference for the weights' distribution.

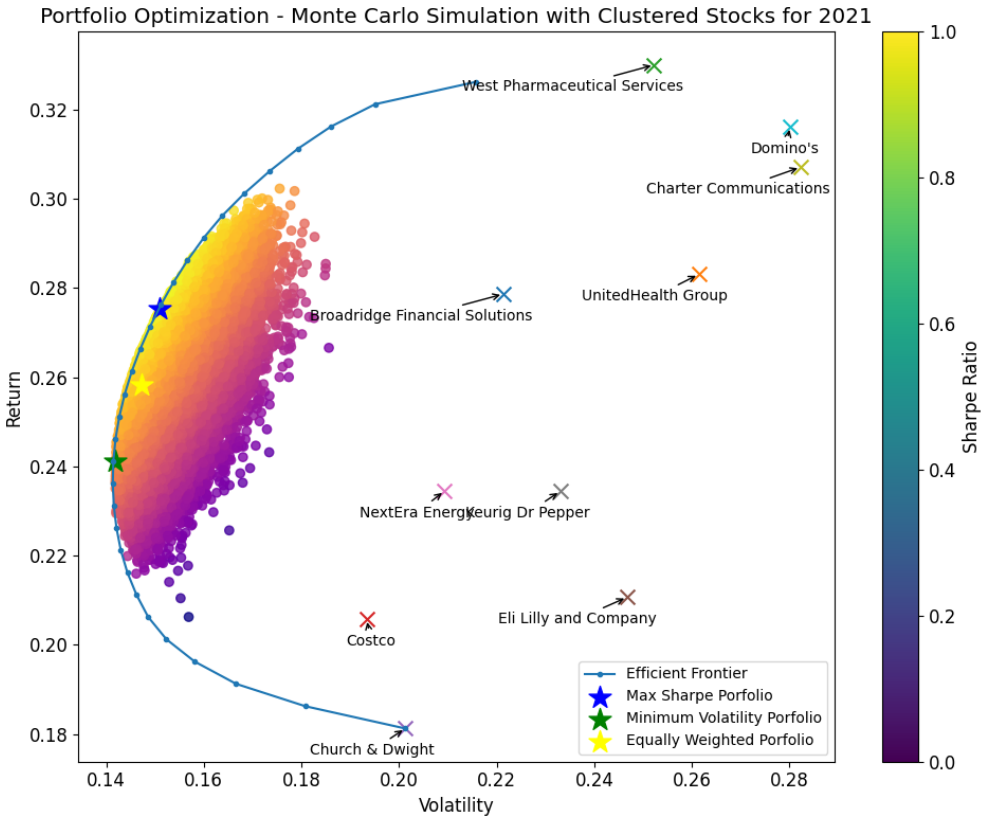

Figure 6. Efficient Frontier of Hybrid Portfolio in 2021

TABLE III

PORTFOLIO STATISTICS COMPARISON FOR 2021

	Portfolio	*Max Sharpe*	*Min Vol*	*Equal*
Return	Cluster	31,35	30,81	31,72
	Random	19,58	6,82	15,88
Volatility	Cluster	13,26	12,31	12,21
	Random	13,28	11,29	10,81
VaR (5%)	Cluster	1,31	1,23	1,21
	Random	1,08	1,13	1,07
CVaR (5%)	Cluster	1,83	1,65	1,64
	Random	1,73	1,53	1,47
Sharpe	Cluster	2,34	2,47	2,57
	Random	1,45	0,57	1,43
MDD	Cluster	-10,78	-10,29	-9,33
	Random	-11,94	-10,33	-8,43

a) VALUES EXPRESSED IN PERCENTAGE

In the specific case of 2021, the comparison exhibits a clear performance advantage by the clustering regime, since all three portfolios yield a higher annualized return than the randomly selected ones, while providing a similar annualized volatility. Furthermore, the level of risk-adjusted return, or Sharpe ratio, for the clustering strategy is higher for each kind of optimal portfolio. The two strategies display a similar and relatively low degree of downside risk, since the highest maximum drawdown difference amounts to 1,16% while the total range between both strategies wides from -8,43% to -11,94% which is significantly lower than the average maximum drawdown for S&P 500 during the same year, specifically 18,65% as previously showed in Table I.

According to Balbas, A., Balbas, B., and Balbas, R. (2017), common measures in risk management frameworks can be represented by the Value-At-Risk (VaR) and Conditional Value-At-Risk (CVaR). As shown in Figure 7, the first represents the maximum expected loss with a specified confidence level, defining the exclusion of extreme losses, which was set to 5% in this case. Between the different version of VaR such as historical, parametric Gaussian, non-parametric Gaussian and Cornish-Fisher approach, the latter was employed since it does not assume normality or any specific distribution for returns. The complementary CVaR measures the average extreme loss of the returns excluded by confidence level of the VaR, in this case 5%. Both VaR and CVaR maximum values for each strategy results lower than 2%, which altogether with the level of MDD can be perceived as an acceptable degree of downside risk.

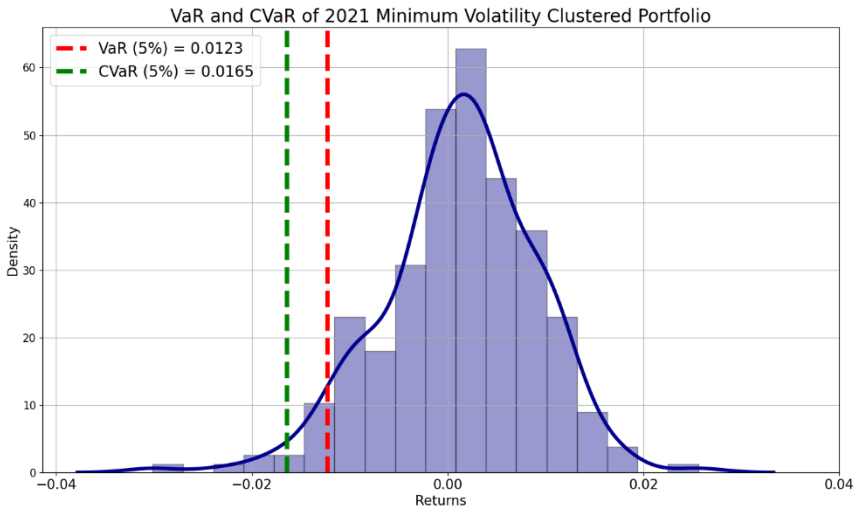

Figure 7. VaR and CVaR of 2021 Minimum Volatility Clustered Portfolio

6. ARIMA

In this phase of the project a prediction of 12 months for each train-test window was employed on the selected stocks by the agglomerative hierarchical clustering algorithm with complete linkage. The two most popular ARIMA techniques are defined as univariate and multivariate. To predict future values, univariate solely considers the time series' past values. In addition to the series of values, multivariate additionally employs endogenous variables to produce the forecast (George E. P. Box et al., 2016). For the scope of this analysis, a univariate approach was employed.

ARIMA stands for Auto-Regressive Integrated Moving Average and is different from an ARMA model which does not require the integration component, more precisely differencing, to analyse the time-series.

As explained by Ho and Xie (1998), ARIMA models are constructed by three components which are described as follows:

Auto-Regressive (AR) refers to a model that describes a changing variable that regresses on its own lagged, or prior, values. If there is autocorrelation between lags, the time-series is called to have memory since each value movement is also caused by its previous, therefore time has an important role in defining the pattern.

Integrated (I) represents the differencing of raw observations to allow the time series to become stationary. Differencing is a crucial step in the process, which specifically requires subtracting the current value of the series from the previous one, or from a lagged value, resulting in data observations replaced by the difference between their value and the previous. This technique can be applied one or multiple steps prior in time depending on the level of seasonality and trend present in the time-series. Moreover, whenever a dataset is considered non-stationary, differencing can be useful to treat its statistical properties as constant over time, perhaps in conjunction with nonlinear transformations such as logging or deflating.

Moving Average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations, hence indicates that the forecast error is a linear combination of past respective errors. This process can depict long-term trends through the computation of a certain statistical

parameter, mean or variance amongst them, over time using a rolling technique that allows the model to track and detach the pattern.

The Framework works with the assumption that the time-series forecast can be estimated using an ARMA model if the dataset is stationary or an ARIMA model if it is non-stationary (Van Greunen et al., 2014). Stationarity in a time-series is a property which defines the relationship between its statistical parameters and time. If these parameters, mean and variance amongst them, vary over time following a seasonal pattern or a trend, the time-series can be defined as non-stationary (Dixit & Jain, 2021). In a particular case where the mean distribution of the values is identically varying around zero, therefore also the variance distribution is constant, and the observations are independent with each other, the dataset can be called white-noise time-series since it is not influenced by the time. A graphical analysis of a rolling mean with a window set to 60 days was implemented for displaying whether the principal statistical parameters were changing over time, hence assume non-stationarity of the time-series. The following analysis were computed on a sample stock, specifically Verizon Communications, to display the steps and principles of the process.

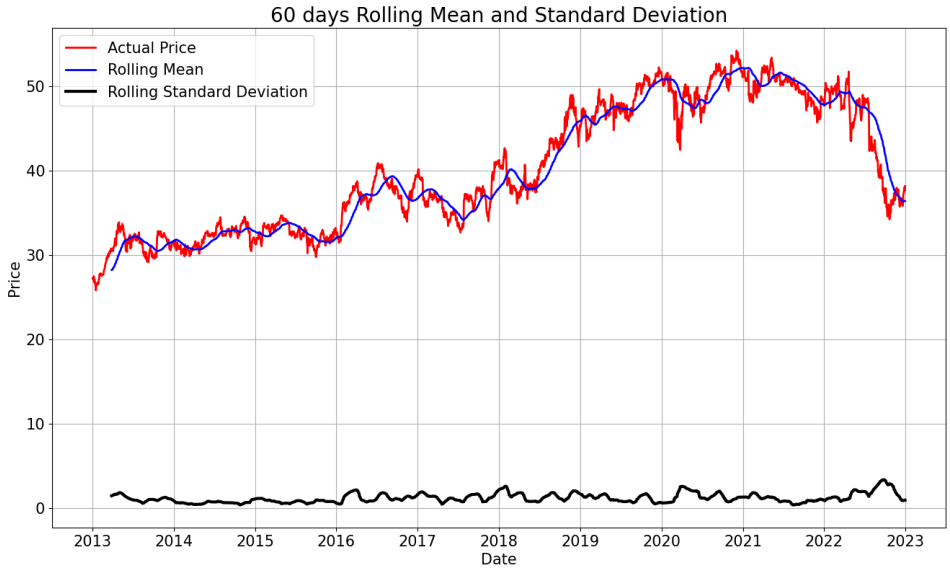

Figure 8. 60-days Rolling Mean and Standard Deviation of Verizon

As shown in Figure 8, the 60 days rolling mean appears to be time dependent and therefore not constant.

Although the conducted analysis showed a mean affected by variations of time and the time-series might be considered non-stationary, ad additional statistical analysis was conducted to confirm the assumption. A requirement of the model is that its autocorrelations, or correlations with its own prior values, remain constant throughout time, more precisely close to 0, apart from the first observation lag-0 which is correlated with itself.

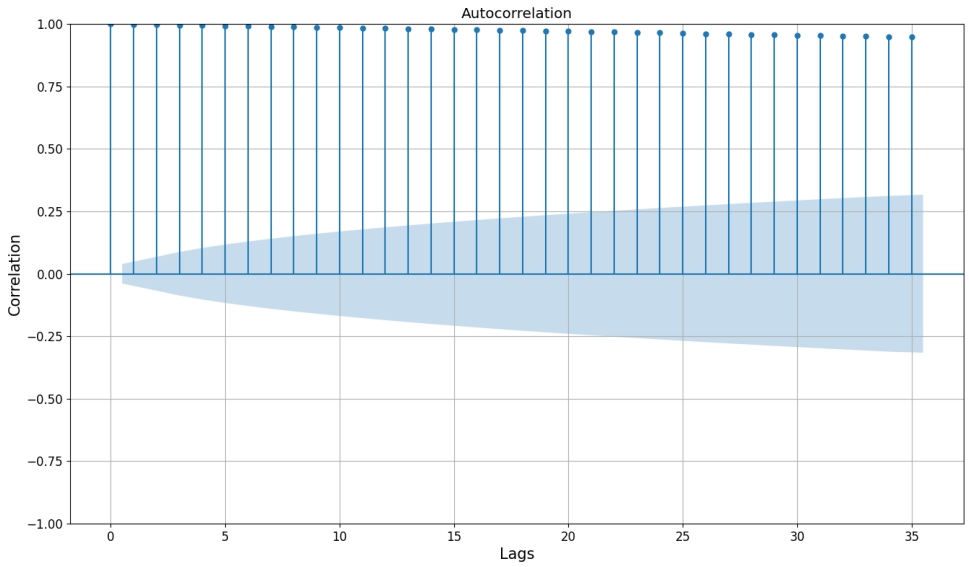

Figure 9. Autocorrelation of Verizon Price Time-Series

As shown in Figure 9, each observation is correlated with its prior lags. Specifically, the autocorrelation of a time series Y at lag-1 is the coefficient of correlation between Y_t and Y_{t-1}, which can be assumed to be also the correlation between Y_{t-1} and Y_{t-2}. Moreover, if Y_t is correlated with Y_{t-1}, and Y_{t-1} is equally correlated with Y_{t-2}, then Y_t and Y_{t-2} should also have a certain degree of correlation. Indeed, the amount of correlation expected at lag-2 is precisely the square of the lag-1 correlation. Hence, the time-series can be defined as non-stationary (De Gooijer Jan G., 1980).

However, for the scope of the analysis stock returns were used as inputs for the model, resulting in a time-series without the requirement of differencing since its values fluctuates around 0, hence the degree of autocorrelation is relatively low as shown in Figure 10.

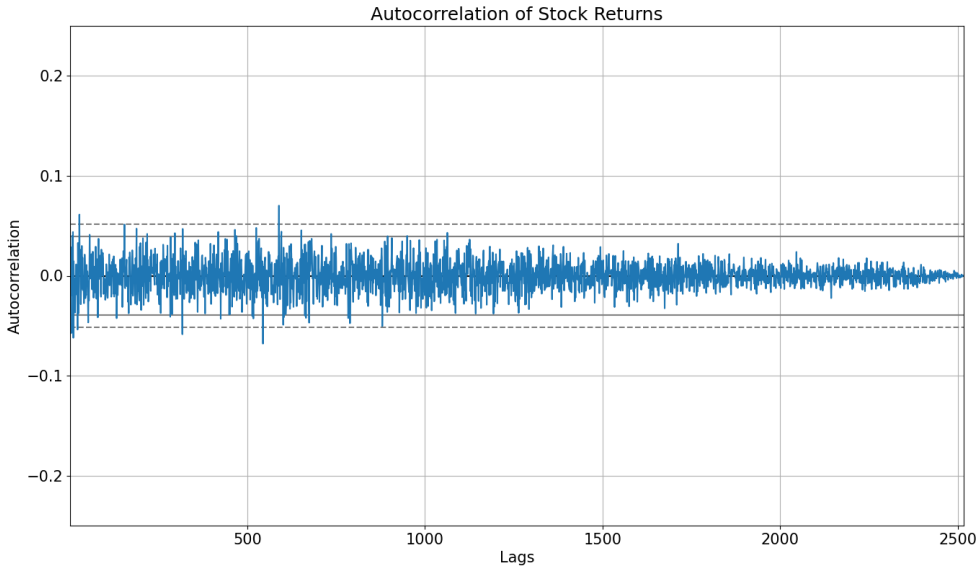

Figure 10. Autocorrelation of Verizon Returns Time-Series

According to De Gooijer (1980), each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be with *p*, *d*, and *q*, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

p: the number of lag observations in the model, also known as the lag order, which represents the AR term.

d: the number of times the raw observations are differenced; also known as the degree of differencing, which represents the I term.

q: the size of the moving average window, also known as the order of the moving average, which represents the MA term.

During each model's iteration, a built-in function of the *pmdarima* Python's package was employed to assess the correct parameters' values for each stock. The specific function, called *auto_arima*, identified *p*, *d*, and *q* parameters while increasing the model's efficiency and reducing the time for computations, which would have been higher if calculated manually for each stock and iteration.

The summary results of ARIMA after applying the selected parameters for Verizon Communications are shown in Equation 12:

$$(12) \qquad Y_t = -0.8840Y_{t-1} - 0.0552Y_{t-2} + 0.8532\varepsilon_{t-1},$$

where Y_t is the predicted value, Y_{t-1} and Y_{t-2} are the first and second order AR term and ε_{t-1} is the first order MA term, which defines the ARIMA equation implemented.

The produced model was then applied to the clustered portfolio of the relative train-test period, and results were compared with the actual returns evolution while an accuracy metrics was chosen to measure the model. In this case mean absolute error was selected, which can be defined relatively low in every iteration, as shown in Table IV:

TABLE IV

MEAN ABSOLUTE ERROR OF ARIMA PREDICTION

2019		*2020*		*2021*		*2022*	
Ticker	***MAE***	***Ticker***	***MAE***	***Ticker***	***MAE***	***Ticker***	***MAE***
UNH	0,06	*MSCI*	0,06	*BR*	0,04	*KDP*	0,05
ATO	0,02	*MMC*	0,07	*UNH*	0,06	*MSFT*	0,08
BR	0,03	*CTAS*	0,11	*WST*	0,05	*LLY*	0,05
IEX	0,05	*DPZ*	0,06	*COST*	0,05	*EXR*	0,09
PGR	0,05	*SYY*	0,12	*CHD*	0,05	*CHTR*	0,13
HD	0,04	*HD*	0,07	*LLY*	0,09	*ODFL*	0,10
FI	0,04	*UNH*	0,05	*NEE*	0,05	*WST*	0,10
CME	0,05	*STZ*	0,09	*KDP*	0,05	*COST*	0,09
BSX	0,05	*MKTX*	0,08	*CHTR*	0,05	*CHD*	0,05
KDP	0,05	*NEE*	0,06	*DPZ*	0,06	*DPZ*	0,10
STZ	0,08	*KDP*	0,06				

Therefore, assumed the model had an acceptable level of forecasting accuracy, predictions with length of a year were made over the train-test splits, which are four, resulting in the same number of forecasting windows. As previously mentioned, the ARIMA framework was applied to the actual returns of selected stocks from clustering process, which provided returns over one year period. The weights were then computed for the two optimal portfolios, minimum volatility and maximum Sharpe ratio through quadratic optimization using the Python package *SciPy*. Specifically, shorting and

leverage were not allowed while the total weight of the portfolio was set to 100% with each single component weight value between 0 and 100%. The objective of the function was to minimize the portfolio volatility or to minimize the inverse of Sharpe ratio, hence maximize it, using annualized returns, volatility and Sharpe ratio computed on each prediction year. The function was initialized with equal weights for each stock providing the algorithm a set of values to modify during every optimization iteration. After the optimal weights solution was reached, this was used to compute and simulate the portfolios with actual returns of the same forecasted year, in order to back test the approach.

7. RESULTS AND DISCUSSION

After the algorithm was completed and executed for the three different portfolio allocation strategies, the portfolio's performance and statistics were compared to the selected benchmark index which in this case was S&P 500.

One drawback of the two portfolios created with machine learning techniques can be defined with a high level of transaction costs required by the strategies, since they imply a different combination of stocks for each year in order to reach high diversification, hence an active portfolio management approach with a short-term reallocation period.

The following comparisons in Table V and VI, were made with the four-year aggregated results from train and test splits and S&P 500 statistics for the same period.

TABLE V

MAXIMUM SHARPE OPTIMIZATION

	Return	*Volatility*	*VaR*	*CVaR*	*Sharpe*	*Sortino*	*MDD*
S&P 500	12,57	19,44	8,72	10,22	0,56	0,94	-24,77
Monte-Carlo	10,80	20,74	9,54	11,92	0,44	0,70	-29,40
Hybrid	14,30	20,15	8,78	12,30	**0,63**	0,86	-30,34
ARIMA	16,38	21,09	8,96	12,50	**0,70**	0,93	-25,08

TABLE VI

MINIMUM VOLATILITY OPTIMIZATION

	Return	*Volatility*	*VaR*	*CVaR*	*Sharpe*	*Sortino*	*MDD*
S&P 500	12,57	19,44	8,72	10,22	0,56	0,94	-24,77
Monte-Carlo	5,60	**16,27**	7,61	9,92	0,24	0,38	-22,54
Hybrid	14,37	**18,94**	8,28	10,67	0,67	0,99	-22,16
ARIMA	16,25	20,43	8,87	11,71	0,72	0,98	-24,79

The naïve portfolio construction based on random selected stocks and simulated with a Monte-Carlo algorithm for the Sharpe ratio's maximization showed a lower annualized return and higher volatility than the benchmark, therefore holding the index for the same investment period rather than the portfolio should be preferable. The portfolio optimization based on Markowitz theory, hence the global minimum variance portfolio,

achieved its objective and resulted in a lower annualized volatility, at 16.27%, than the benchmark and the other portfolios, nevertheless the level of annualized return was lower than all other approaches resulting in a significantly less competitive risk-adjusted return. Moreover, its Sortino ratio showed inferior values than the index for both optimization problems and therefore a lower excess return using downside volatility. Consequently, this approach can be considered a solid starting point, although it lacks depth in certain aspects, for a more complex and appropriate allocation strategy, which was the case for the other two techniques employed in this project.

The hybrid approach between traditional and machine-learning based optimization techniques, achieved its objective of diversification over a randomly selected portfolio, since a higher Sharpe and Sortino ratio can be interpreted as an indicator of greater risk-adjusted performance during a four-year span with multiples financial markets' contractions. Additionally, the model overperformed the index for both optimization techniques, since the maximization of Sharpe ratio and minimization of volatility were successfully reached with improvements over the S&P 500. Sortino and Sharpe ratios of minimum volatility portfolio resulted higher than both the index and the alternative optimization problem, which can lead to assume the maximization of Sharpe ratio technique as less efficient. To confirm the assumption through a comparison between the optimization techniques, levels of VaR, CVar and MDD resulted lower for the minimum volatility portfolio, hence an enhanced method.

Lastly, the ARIMA model outperformed all other techniques and at the same time showed the highest volatility in both optimization problems. Being the results of the two portfolios nearly identical, with the minimum volatility holding an edge, the maximization of Sharpe ratio can be deemed as a less efficient optimization approach. Moreover, the machine-learning based model, constructed with a clustering selection process and ARIMA prediction, achieved its objective of improving diversification and predicting returns as displayed by Figure and 11 and 12, where monthly cumulative returns of the optimal portfolios for the train-test period are displayed and compared with the benchmark.

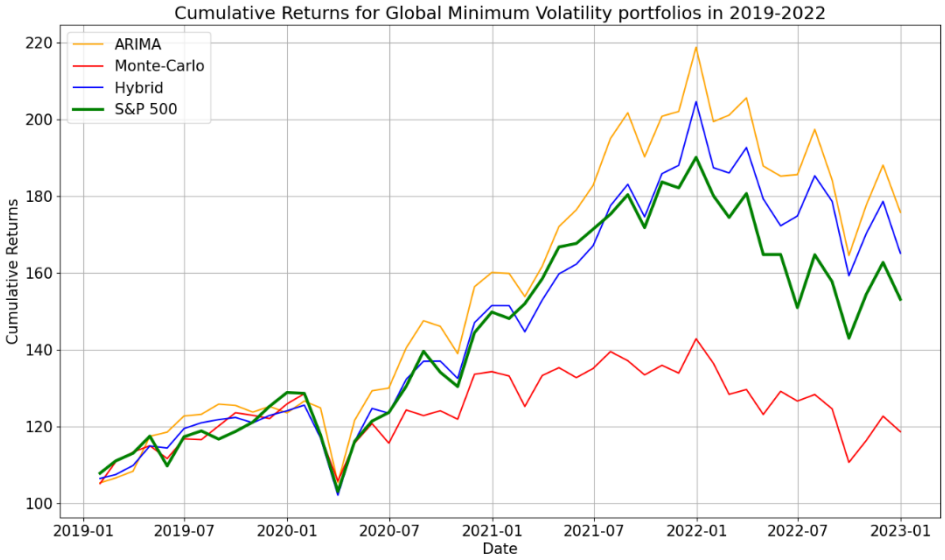

Figure 11. Monthly Cumulative Returns with Minimum Volatility Optimization

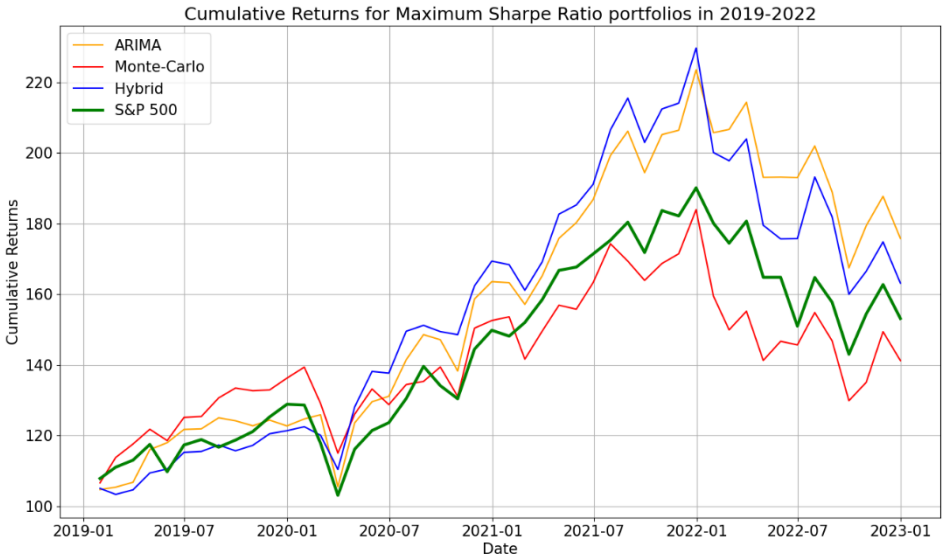

Figure 12. Monthly Cumulative Returns with Maximum Sharpe Optimization

As shown in the figures, the portfolios randomly constructed and optimized with a Monte-Carlo algorithm provided a significant lower level of cumulative return than the considered benchmark, therefore investing in the S&P 500 would assure a higher performance equal to 153% of cumulative return from 2019 to 2022. Contrarily, the

hybrid and machine-learning based approach produced portfolios able to outperform the market index over the four-year period. The hybrid approach recorded 163% and 165% for maximum Sharpe and minimum volatility portfolios respectively, while the ARIMA model displayed a barely identical cumulative return with a value of 176% in each optimal portfolio. As a final consideration of the model, the clustering approach resulted in a successful process of diversification improvement (Marvin & Bhatt, 2015) and consequentially performance, while the addition of ARIMA produced the highest cumulative return and volatility out of the three methods and could be conceived as a confirmation step and complementary analysis. Thus, the stock selection process could be assessed with a hierarchical clustering algorithm, with linkage methods and parameters subjects to modification, and an ARIMA model for a prediction based on historical data instead of random simulations in Monte-Carlo. The selection of which techniques to utilize can depend on different conditions such as level of risk aversion, time horizon or the inclusion of an insurance strategy amongst them.

8. CONCLUSIONS

Although the models and techniques showed a positive impact on performance and diversification of the portfolio, other traditional techniques could be employed to improve the algorithm, such as Capital Asset Pricing Model (CAPM) and FAMA French model which would analyse stock returns as a function of defined factors, such as size and value of firms, and excess return over the market while allowing capital allocation on the risk-free rate (Kianpoor & Dehghani, 2016). This would strengthen the algorithm and allow to comprehend the causality between stocks and market returns. Moreover, developing an investment strategy based on historical data as the input might not be accurate since it cannot be considered an the solely indicator of a security future performance. Analysing stock performance on past prices can be called "technical analysis", while assessing the company's structure and financial results can be defined as "fundamental analysis" and should have the same significance for an investment model as the more analytical approach. The financial statements and companies' reports could be analysed and used as a verification or opposition for the results, and the introduction of an economic condition analysis for each stock would provide a more accurate and stable model.

Additionally, different machine learning models could be applied to enhance the algorithm's learning efficiency and accuracy. According to Chen, Guo, Huang and Jin (2023), neural networks are the most popular methods for predicting time-series, amongst them the Long Short-Term Memory (LSTM) results the most accurate and appropriate, although its development presents multiple difficulties and the necessity of substantial computing power due to the complexity of structure and underlying concepts.

REFERENCES

Balbás, A., Balbás, B., & Balbás, R. (2017). VaR as the CVaR sensitivity: Applications in risk optimization. *Journal of Computational and Applied Mathematics*, *309*, 175–185.

Chen, K., Guo, Z., Huang, X., & Jin, Y. (2023). LSTM for Return Prediction and Portfolio Optimization in America Stock Market. *Proceedings of the International Conference on Financial Innovation, FinTech and Information Technology, FFIT 2022, October 28-30, 2022, Shenzhen, China*.

De Gooijer Jan G. (1980). Exact moments of the sample autocorrelations from series generated by general arima processes of order (p, d, q), d=0 or 1. *Journal of Econometrics*, *14*(3), 365–379.

Dixit, A., & Jain, S. (2021). Effect of stationarity on traditional machine learning models: Time series analysis. *ACM International Conference Proceeding Series*, 303–308.

Elton, E. J., & Gruber, M. J. (1997). Modern portfolio theory, 1950 to date. *Journal of Banking & Finance*, 1743–1759.

Fama F. Eugene. (1995). Random Walks in Stock-Market Prices. *Financial Analysts Journal*, *51*.

Fama F. Eugene. (1965). The Behaviour of Stock-Market Prices. *Journal of Business*, *38*(1), 34–105.

Gagolewski, M., Bartoszuk, M., & Cena, A. (2016). Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Information Sciences*, *363*, 8–23.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, & Greta M. Ljung. (2016). *Time Series Analysis: Forecasting and Control*.

Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, *32*.

Gunopulos, D. (2009). Clustering Overview and Applications. In *Encyclopedia of Database Systems*.

Ho S. L., & Xie, M. (1998). The Use of ARIMA Models for Reliability Forecasting and Analysis. In *Computers & Industrial Engineering* (Vol. 35, Issue 2, pp. 213–216).

Humaira, H., & Rasyidah, R. (2020, March 3). *Determining The Appropriate Cluster Number Using Elbow Method for K-Means Algorithm*.

Idowu, S., Strüber, D., & Berger, T. (2022). Asset Management in Machine Learning: State of research and State of practice. *ACM Computing Surveys*, *55*(7).

Janiesch, C., Zschech, P., & Heinrich, K. (2021). *Machine learning and deep learning*.

Kianpoor, M. M., & Dehghani, A. (2016). The Analysis on Fama and French Asset-Pricing Model to Select Stocks in Tehran Security and Exchange Organization (TSEO). *Procedia Economics and Finance*, *36*, 283–290.

Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, *234*(2), 356–371.

Li, T., Rezaeipanah, A., & Tag El Din, E. S. M. (2022). An Ensemble Agglomerative Hierarchical Clustering Algorithm Based on Clusters Clustering Technique and the Novel Similarity Measurement. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 3828–3842.

Love C. Bradley. (2002). Comparing Supervised and Unsupervised Category Learning. *Psychonomic Bulletin & Review*, *9*, 829–835.

Marvin, K., & Bhatt, S. (2015). *Creating Diversified Portfolios Using Cluster Analysis*.

Nystrup, P., Boyd, S., Lindström, E., & Madsen, H. (2019). Multi-period portfolio selection with drawdown control. *Annals of Operations Research*, *282*(1–2), 245–271.

Nystrup, P., Hansen, B. W., Olejasz Larsen, H., Madsen, H., & Lindström, E. (2018). *Dynamic allocation or Diversification: A Regime-Based Approach to Multiple Assets*.

Pinelis, M., & Ruppert, D. (2022). Machine learning portfolio allocation. *Journal of Finance and Data Science*, *8*, 35–54.

Renugadevi, T., Ezhilarasie, R., Sujatha, M., & Umamakeswari, A. (2016). Stock Market Prediction using Hierarchical Agglomerative and K-means Clustering Algorithm. *Indian Journal of Science and Technology*, *9*(48).

Rezaei, N., & Elmi, Z. (2018). *Behavioural Finance Models and Behavioural Biases in Stock Price Forecasting*.

Rokach, L., & Maimon, O. (2005). Clustering Methods. In *Data Mining and Knowledge Discovery Handbook* (pp. 321–352). Springer.

Seymour Smidt. (1968). A New Look at the Random Walk Hypothesis. *The Journal of Financial and Quantitative Analysis, Special Issue: Random Walk Hypothesis*, *3*(3), 235–261.

Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018). Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster. *IOP Conference Series: Materials Science and Engineering*, *336*(1).

Van Greunen, J., Heymans, A., Van Heerden, C., & Van Vuuren, G. (2014). The Prominence of Stationarity in Time Series Forecasting. *Journal for Studies in Economics and Econometrics*, *38*(1), 1–16.

Van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin and Review*, *25*(1), 143–154.

Van Vliet, P. (2018). *Low Volatility Needs Little Trading*. *44*.

APPENDICES

1. PYTHON CODE

The entire project was constructed and implemented with Python using an environment from the open-source platform Anaconda, called JupyterLab, since it allows interactive development of Data Science, Machine Learning and statistical models. The algorithm was divided into six sections, more specifically five Jupyter Notebooks and one Python toolkit, with different scopes to ease the manipulation and understanding of the code. Each function was included into the toolkit and invoked when necessary, to reduce the lines of code in the core notebooks. The packages used for the entire project are showed in Table A1 below.

TABLE A1

PYTHON PACKAGES EMPLOYED BY THE ALGORITHM

Package	**Scope in the Project**
Portfolio_optimization	Toolkit with more than 40 created functions for different scopes
Pandas	Data manipulation package for tabular data. Data in the form of DataFrames.
Numpy	Scientific computing package for array processing
Yfinance	API to download stocks from yahoo.finance
SkLearn	Package for Machine Learning techniques including Clustering
Matplotlib	Package for data visualization
Seaborn	Additional data visualization features for Matplotlib package
SciPy	Package for quadratic optimization
Random	Package for generating random values
Statsmodels	Package for statistical models and ARIMA computation
Pmdarima	Package for ARIMA parameters computation with auto_arima

In the first notebook, called “Data Preparation and Pre-Processing”, all stocks from S&P 500 were downloaded through the Yahoo Finance API and their adjusted close prices along with name and the belonging industry were extracted and converted into two different datasets. Moreover, another dataset was created from the 1-year US treasury bill which was downloaded from the Federal Reserve Economic Data (FRED) website. Additionally, a dataset with the adjusted close price of the benchmark index was created, hence S&P 500. These datasets were then cleaned to exclude any missing data and organized into an optimal format.

The second notebook is called “Cluster Analysis”, where firstly the screening procedure was adopted on the stocks dataset to reach a total of 95 out of 500. Afterwards, each clustering technique was implemented and tested in each year of the train-test periods to achieve diversification, more specifically k-means and hierarchical agglomerative clustering with single, average or complete linkages methods. After excluding non-optimal techniques, the highest performing, from a Sharpe ratio standpoint, was then selected and implemented into the stocks’ dataset.

The third notebook, called “Modern Portfolio Theory and Monte-Carlo”, presents the essential section of the algorithm. In this section the stocks selected by the clustering algorithm and a set of randomly selected were converted into returns and used as input for the principal function of the project. A Monte-Carlo simulation of the two set of stocks was computed for each train-test period to back-test the model, hence the algorithm iterated four times since the testing years were from 2018 to 2022. After the MPT optimizations with maximum Sharpe ratio, minimum volatility and equally weighted portfolios were completed for each scenario, the efficient frontier and cumulative returns comparison with S&P 500 of the relative period were plotted. The returns, weights and statistics of each period were then stored for both stock selection methods.

In the fourth notebook, called “ARIMA”, preliminary statistical analyses were computed on the set of stocks selected by the clustering algorithm. Stationarity and auto-correlation of time-series were verified to implement an optimal model. A function for ARIMA prediction was then created, which specifically analysed the *p, d, q* parameters through an *auto_arima* built-in function and computed a 1-year forecast for each set of clustered stocks during the relative train-test split period. These predictions were then

utilized as input for an optimization method to maximize the Sharpe ratio and minimize the volatility. The algorithm, through quadratic programming functions from the SciPy package, computed the optimal weights for each scenario which were subsequently employed to simulate returns for each testing year. Returns, weights and statistics of each period were then stored.

The last notebook, called “Results and Comparison”, imported the three portfolio strategies data and computed a comparison between them and the benchmark index. Cumulative returns of the portfolios were plotted while the statistics were computed for each testing year. A final comparison was then made with the total 4-year cumulative returns and statistics between the three techniques and S&P 500 to analyse the total performance of the model.

2. EFFICIENT FRONTIERS

2.1 Clustered Stocks

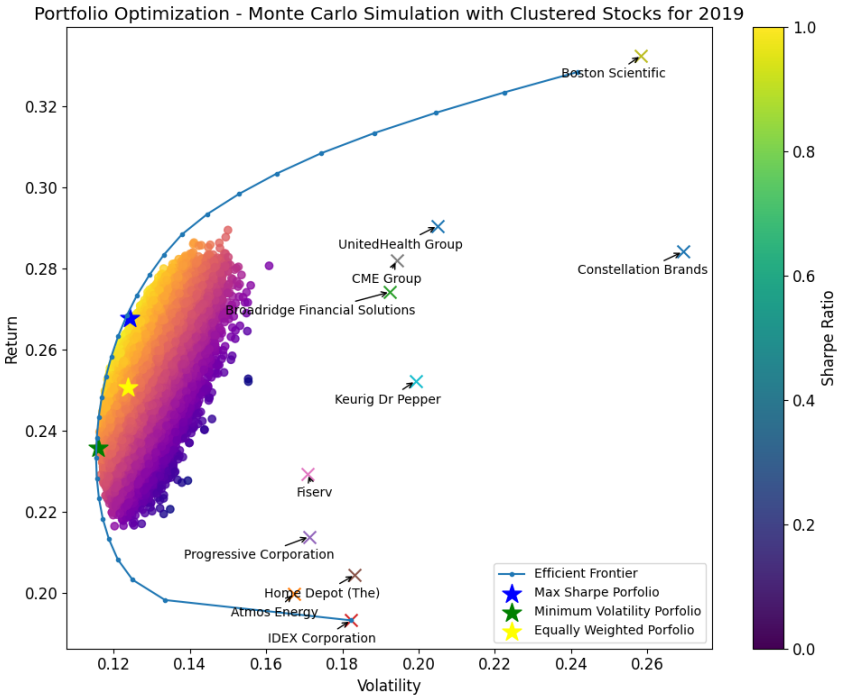

Figure A1. Efficient Frontier of 2019 Clustered Stocks

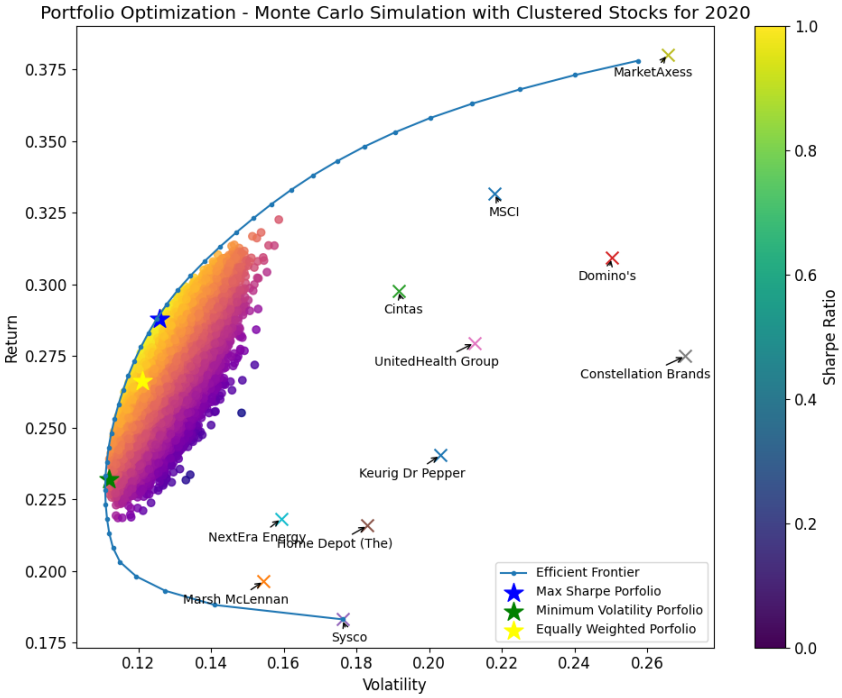

Figure A2. Efficient Frontier of 2020 Clustered Stocks

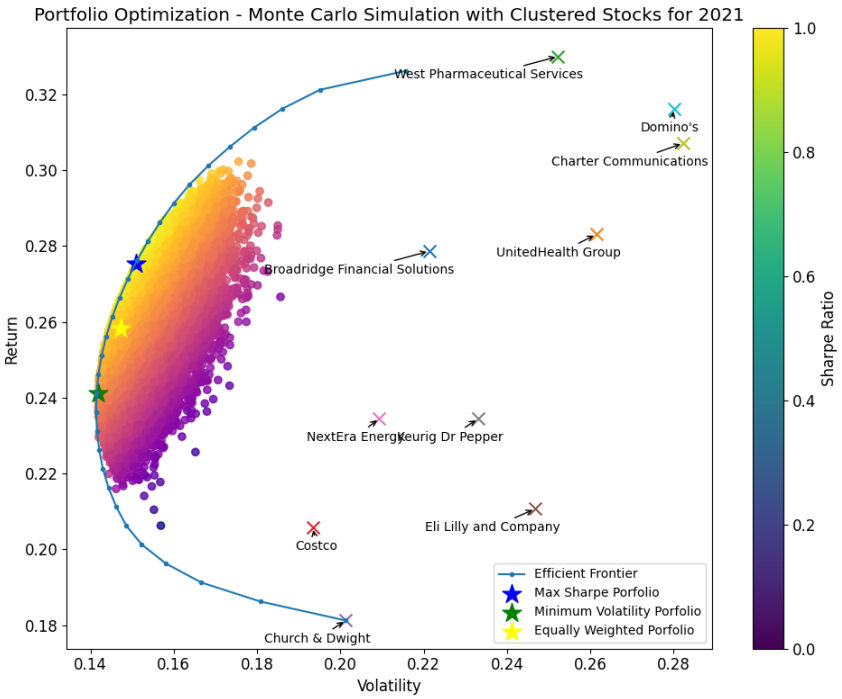

Figure A3. Efficient Frontier of 2021 Clustered Stocks

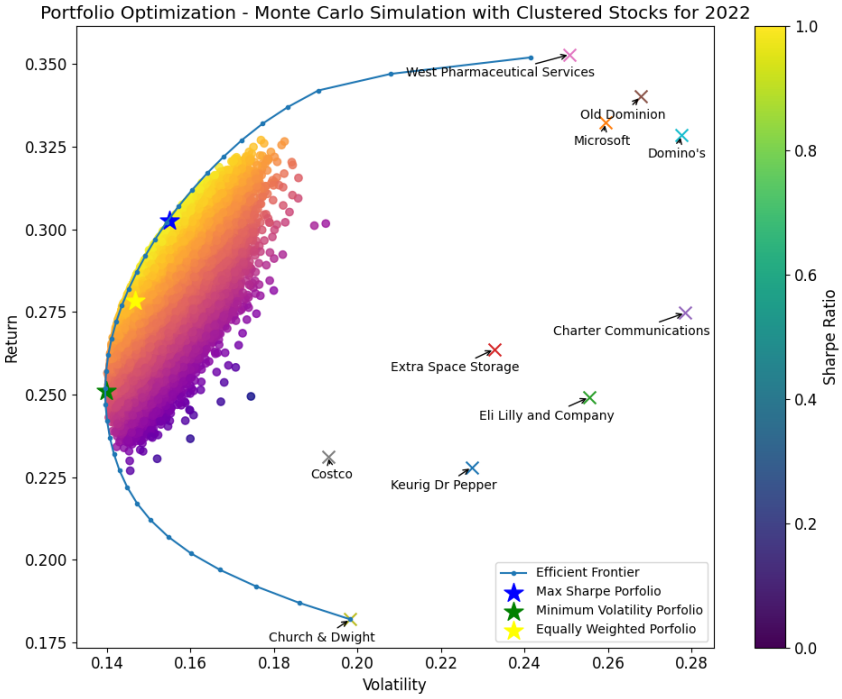

Figure A4. Efficient Frontier of 2022 Clustered Stocks

2.2 Random Stocks

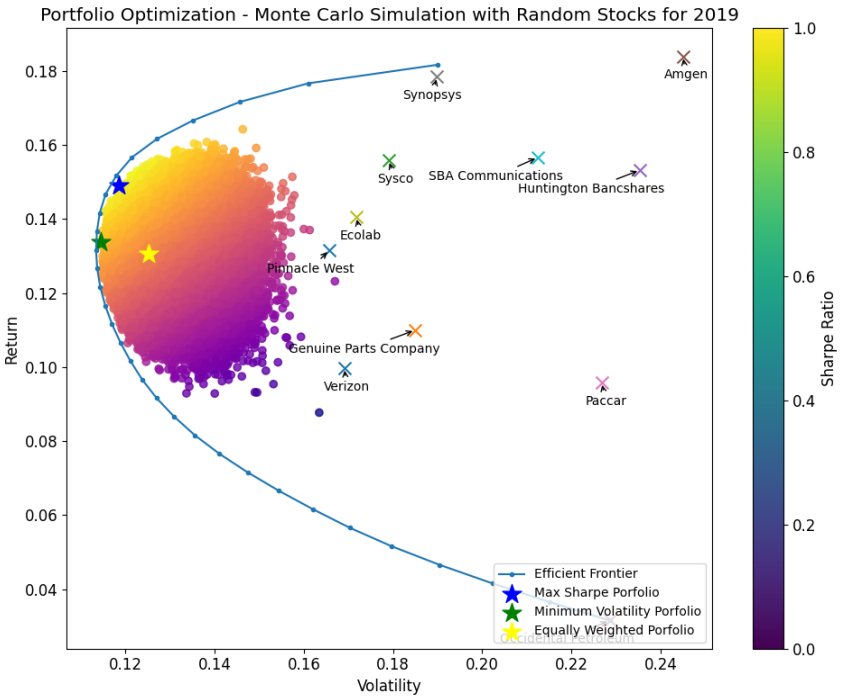

Figure A5. Efficient Frontier of 2019 Random Stocks

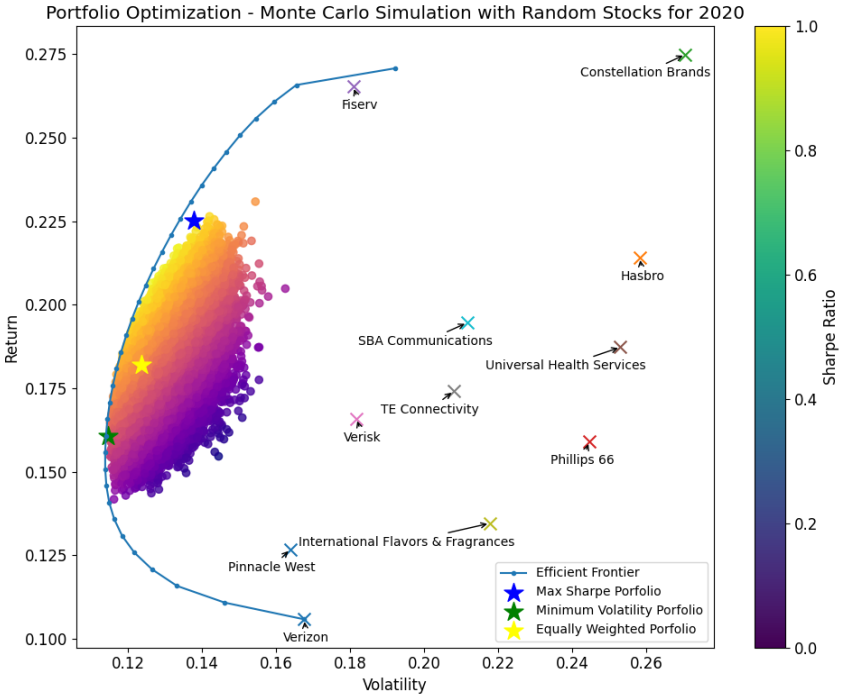

Figure A6. Efficient Frontier of 2020 Random Stocks

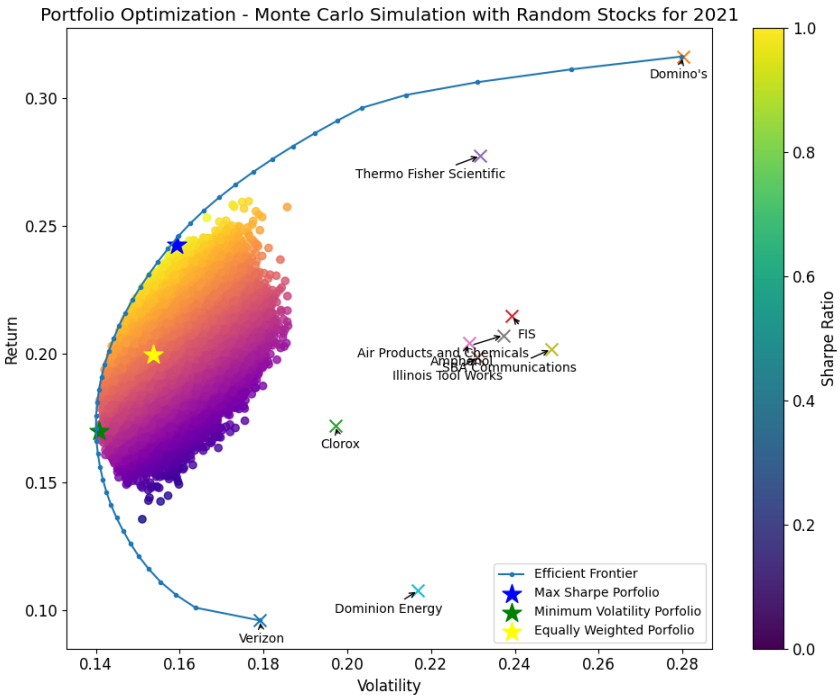

Figure A7. Efficient Frontier of 2021 Random Stocks

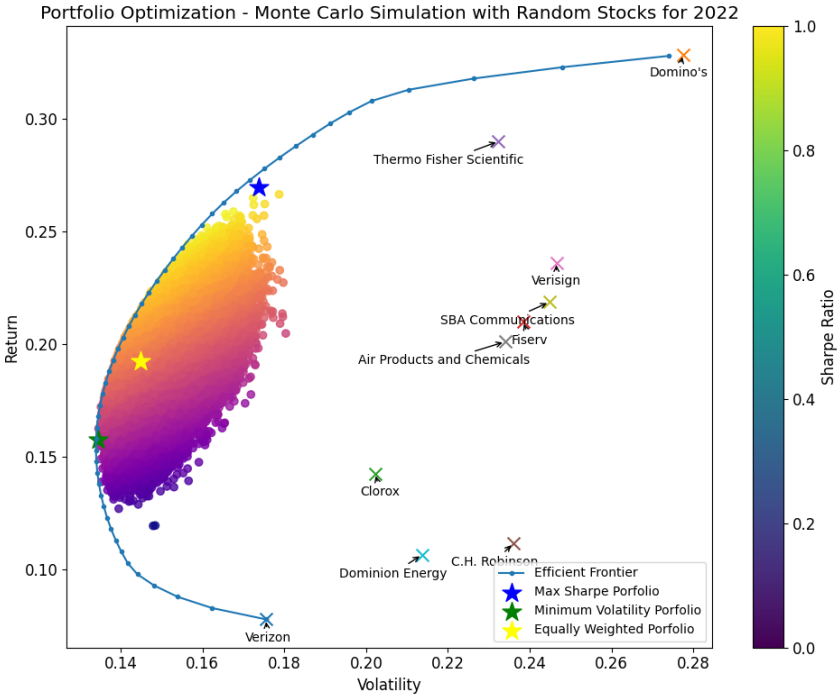

Figure A8. Efficient Frontier of 2022 Random Stocks

3. CUMULATIVE RETURNS

3.1 Clustered Stocks

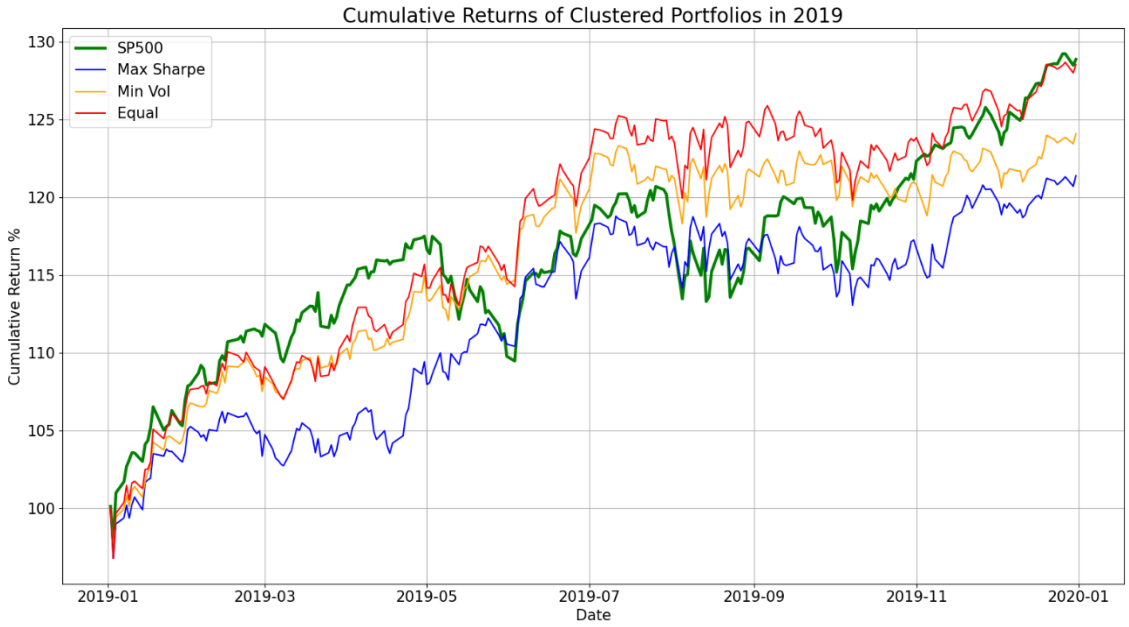

Figure A9. Daily Cumulative Returns of 2019 Clustered Portfolios

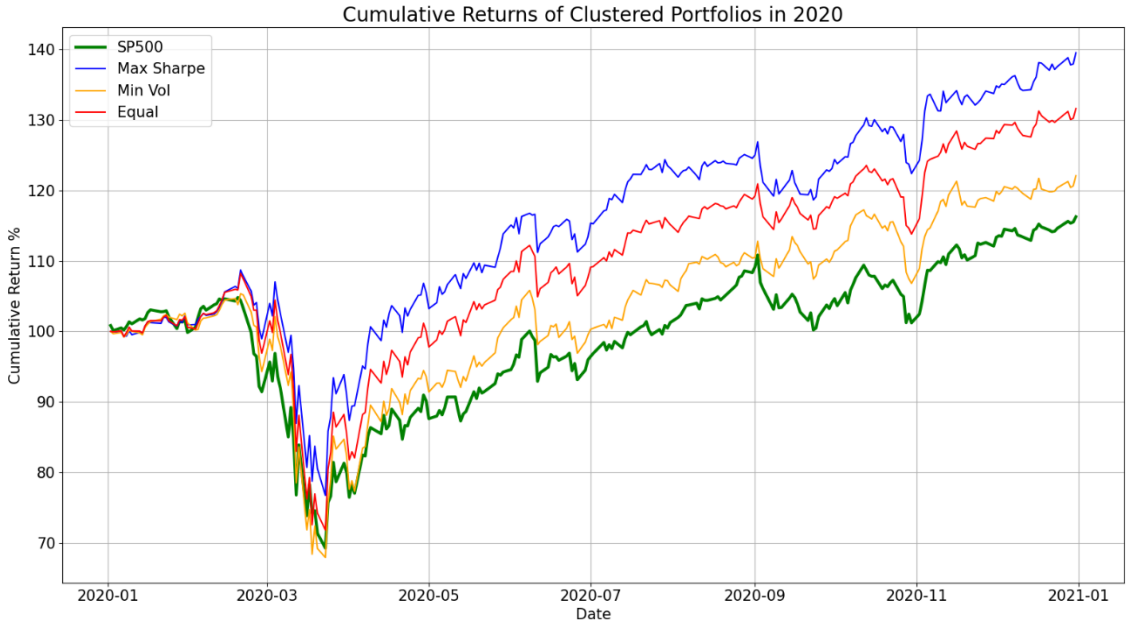

Figure A10. Daily Cumulative Returns of 2020 Clustered Portfolios

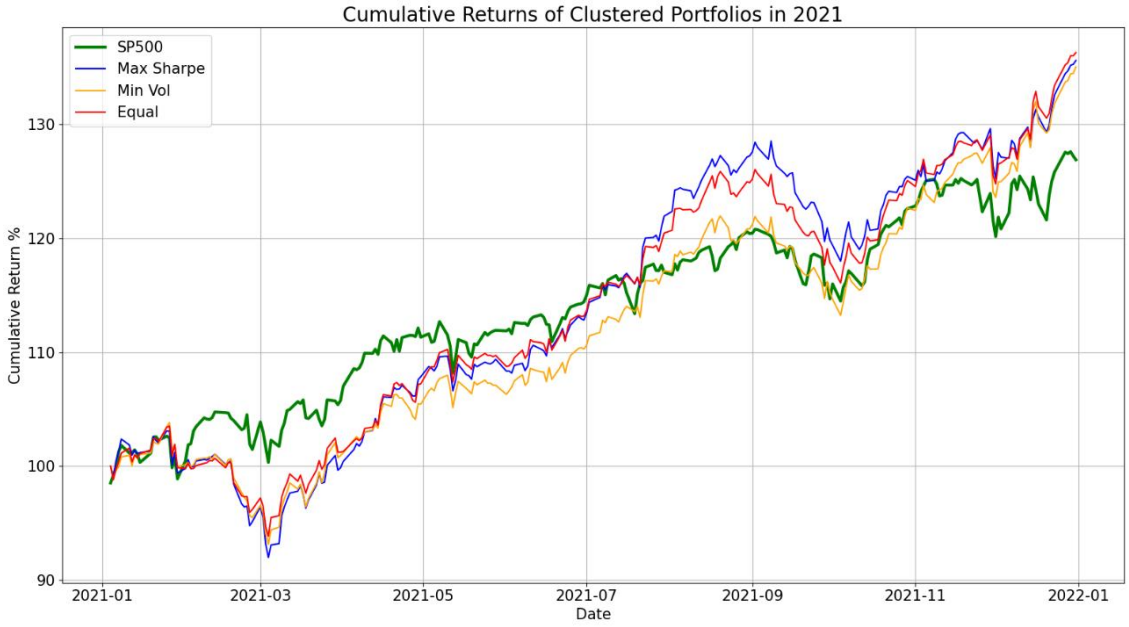

Figure A11. Daily Cumulative Returns of 2021 Clustered Portfolios

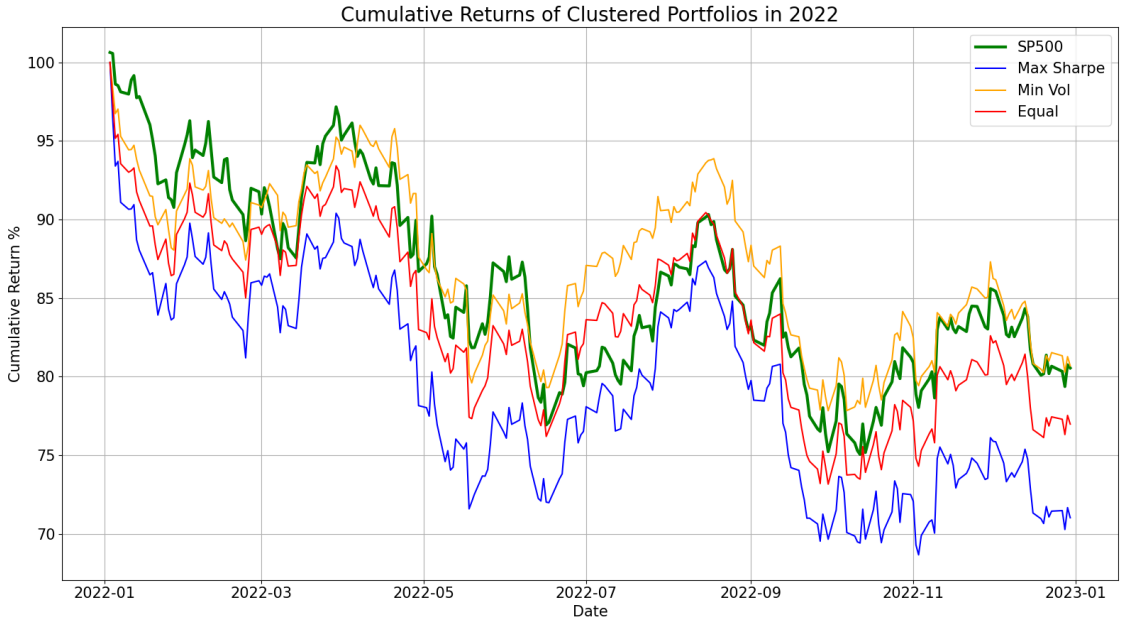

Figure A12. Daily Cumulative Returns of 2022 Clustered Portfolios

3.2 Random Stocks

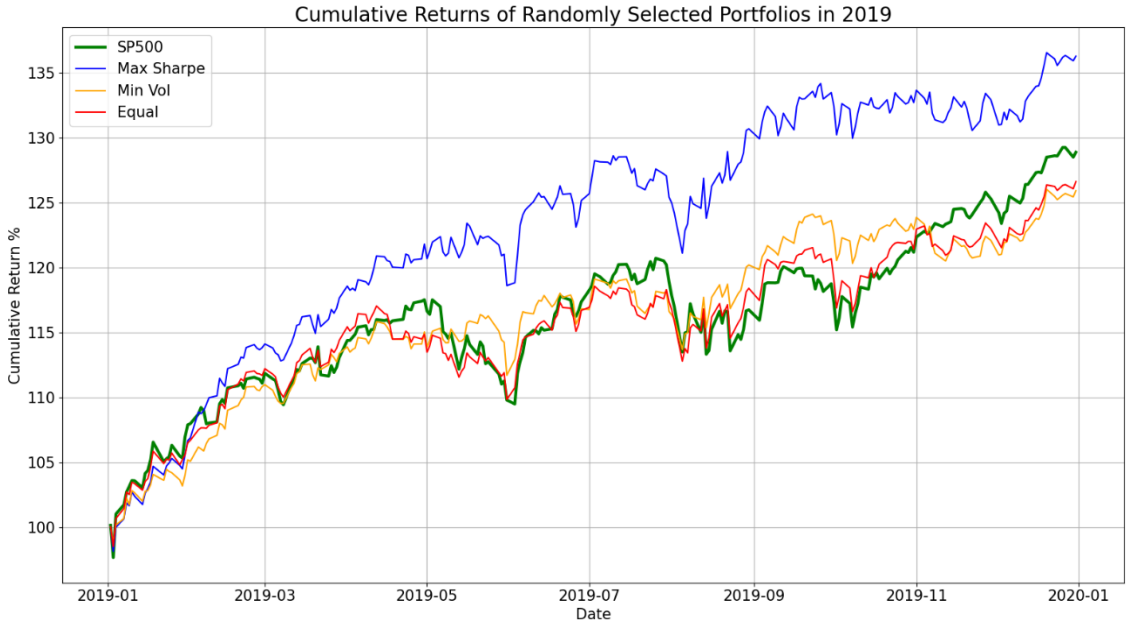

Figure A13. Daily Cumulative Returns of 2019 Random Portfolios

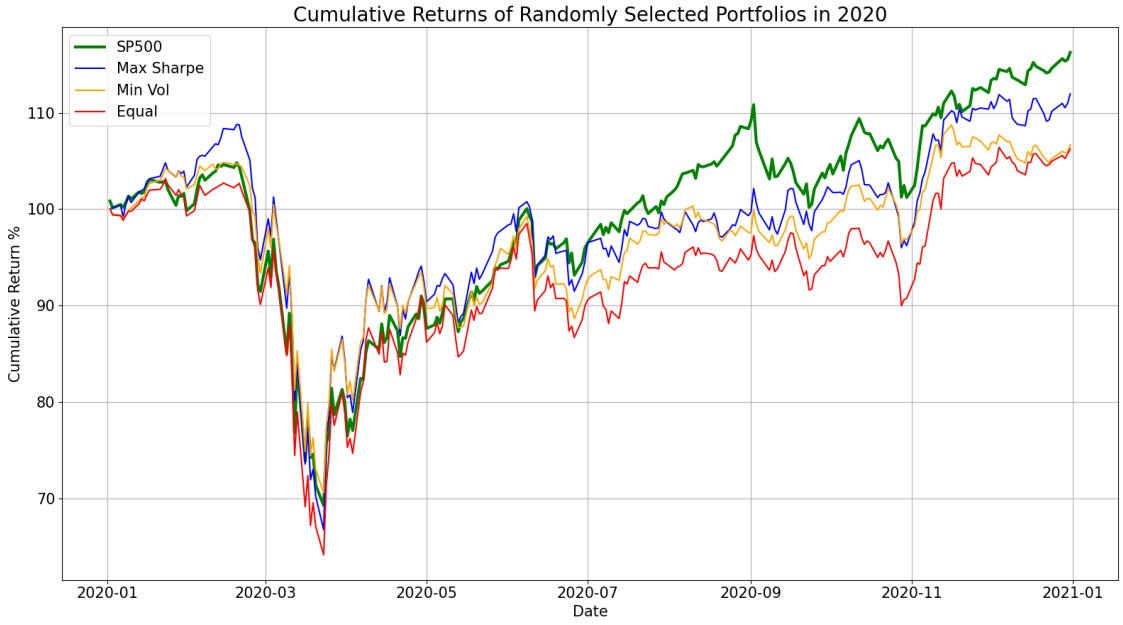

Figure A14. Daily Cumulative Returns of 2020 Random Portfolios

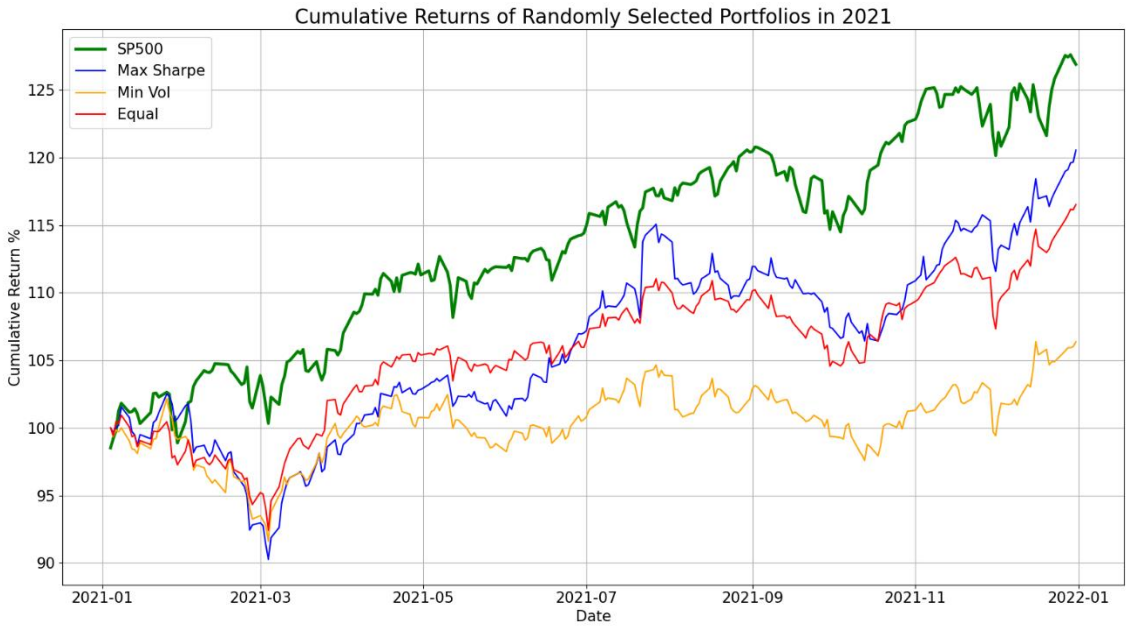

Figure A15. Daily Cumulative Returns of 2021 Random Portfolios

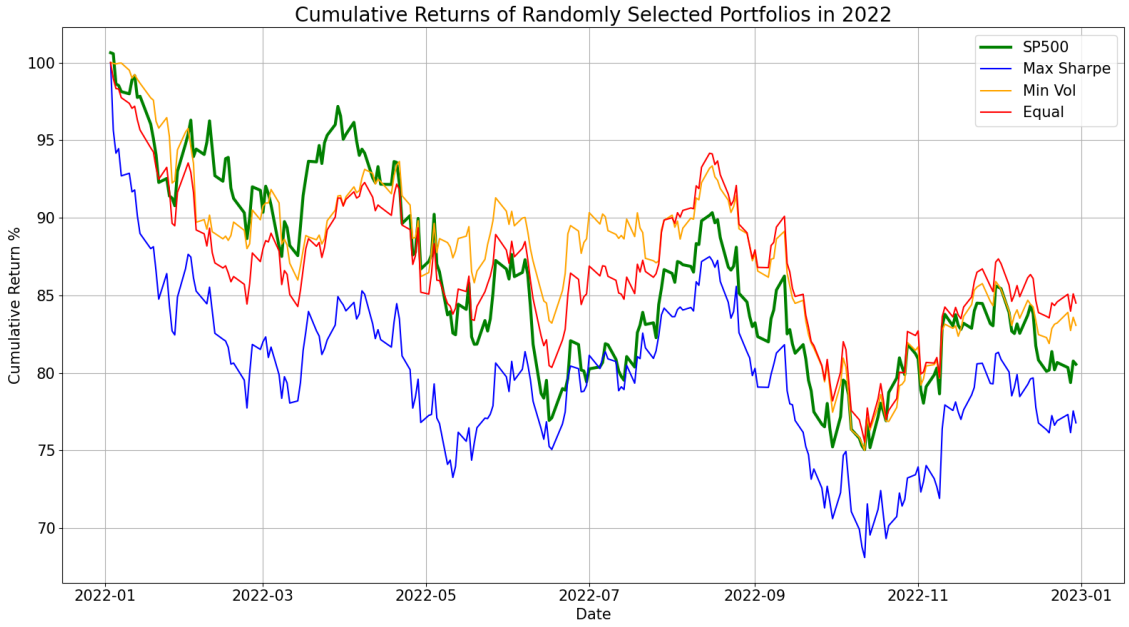

Figure A16. Daily Cumulative Returns of 2022 Clustered Portfolios

4. ARIMA DIAGNOSTICS

Diagnostic plot of ARIMA model with *p*, *d*, *q* parameters selected by *auto_arima* built-in function from *pmdarima* for stock returns of Verizon Communications. Selected parameters:

1. Auto-Regression (AR): $p = 2$
2. Integrated (I): $d = 0$
3. Moving Average (MA): $q = 1$

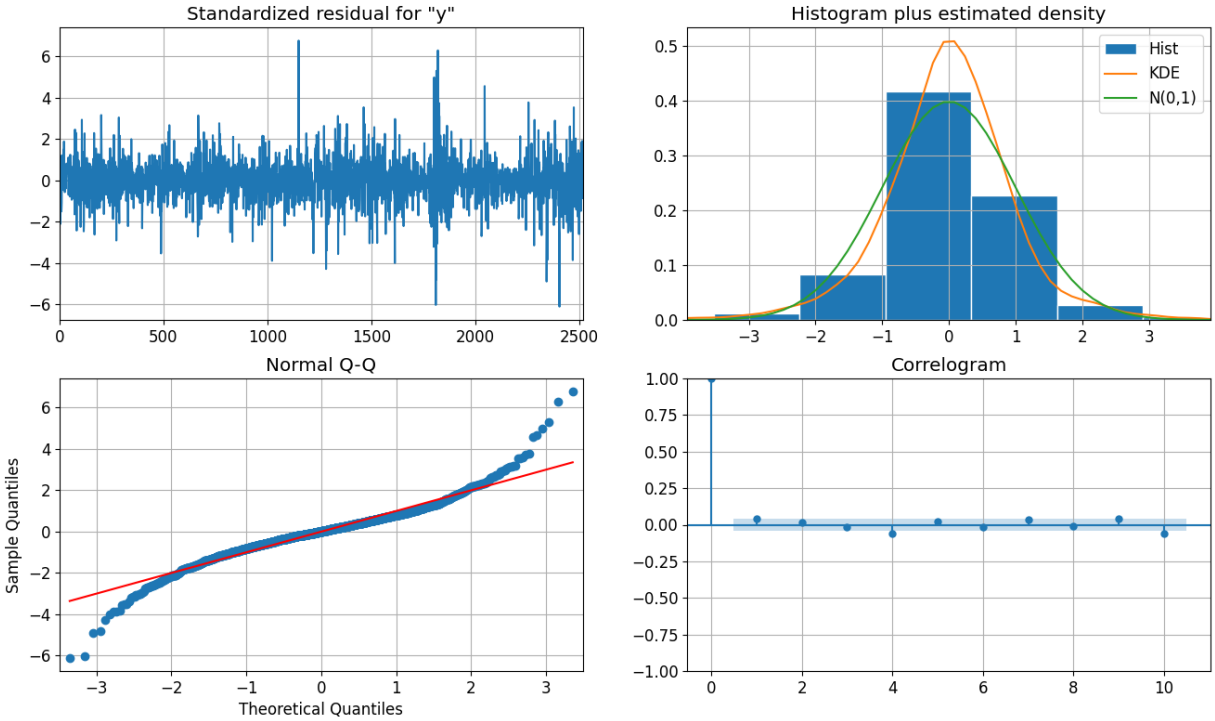

Figure A17. Diagnostics of ARIMA model for Verizon Communications