UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF FINANCE AND BANKING

GRADUATION THESIS

CLUSTERING STOCKS TO GENERATE INVESTMENT PORTFOLIO

Lecture: **Master. Phan Huy Tam**

Student: **Huynh Thi Ha Thanh**

Code: **K194141746**

Class: **K19414C**

Ho Chi Minh City, 04/2023

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF FINANCE AND BANKING

GRADUATION THESIS

CLUSTERING STOCK TO GENERATE INVESTMENT PORFOLIO

Lecture: **Master. Phan Huy Tam**

Student: **Huynh Thi Ha Thanh**

Code: **K194141746**

Class: **K19414C**

Ho Chi Minh City, 4/2023

ASSESSMENT OF INSTRUCTOR

Assessment

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

TABLE OF CONTENTS

List of Figures

List of tables

List of Acronyms

No	Acronym	Meaning
1	2D	2 dimensions
2	D/E	Debt to Equity ratio
3	HNX	Hanoi Stock Exchange
4	HOSE	Ho Chi Minh City Stock Exchange
5	P/B	Price to Book ratio
6	P/E	Price to Earnings ratio
7	PCA	Principal Component Analysis
8	ROA	Return on Assets
9	ROE	Return on Equity

Abstract

The research aims to cluster stocks with similar financial characteristics and select stocks from each cluster to generate investment portfolio. The K-means clustering algorithm and the Elbow and Silhouette methods are used in this study for selecting the number of clusters. Financial data is collected from financial reports of listed companies on the HNX (Hanoi Stock Exchange) and HOSE (Ho Chi Minh City Stock Exchange) exchanges from 2017 to 2022. Financial ratios, including ROA, ROE, P/B, P/E, D/E, and total assets, are analyzed using the K-means algorithm to cluster stocks with similar financial characteristics into clusters. The results showed that the K-means algorithm can effectively clustered stocks and selected representative stocks to construct investment portfolio. This clustering method was effective for four years (2017-2020) but uneffective in 2021.

Keywords: *K-means, Stock Clustering, Investment Portfolio*.

1. Introduction

1.1. Reason for choosing the topic

In recent years, the Vietnamese stock market has experienced significant growth in the number of listed companies and market capitalization. Simultaneously, the number of new securities accounts has been increasing sharply. This indicates that investors are becoming more interested in the stock market and are looking to increase their income through stock investments. However, investing in stocks involves risks. Therefore, diversifying investment portfolios by building an effective portfolio is an effective way to minimize risks.

Diversifying investment portfolios means spreading investments across various stocks, different industries, and different countries to minimize risks and increase opportunities for profit. In this way, if one stock or industry is affected by a certain factor, a portion of the investment portfolio is still secured by other stocks. An important thing that investors must do is evaluate and select stocks with potential for growth and good future profitability that fit their investment goals. Investors often struggle to choose stocks because it depends on the risk-return characteristics of each type of stock. Typically, they will choose stocks with high profit returns and allocate their money to this group. However, they face the challenge of deciding which type of security to choose and how much to invest in each type (Baser, P., & Saini, J. R., 2015). Therefore, an automated method of classifying or grouping stocks would be very useful and necessary for investment decision-making and diversification of stocks for investors. Stocks will be grouped through clustering methods to maximize similarities within groups and minimize similarities between groups. This will allow a person to find out which asset combinations can create a well-diversified investment portfolio (Baser, P., & Saini, J. R., 2015).

Stock clustering using machine learning can be an effective method for grouping stocks with similar characteristics together. When analyzing a large number of stock properties, machine learning clustering methods require less time to analyze, whereas traditional analytical methods consume a lot of time when working with large data sets. Additionally, clustering with machine learning helps to limit the determination of the number of clusters needed for cluster, while traditional methods require analysts to determine how many clusters to classify, which can sometimes lead to inaccurate classification results. Therefore, it can be said that machine learning provides more accurate classification results and can handle larger and more complex data sets compared to traditional clustering methods. This method can help investors, fund managers, and financial experts evaluate and compare the performance of stocks in the same classification group. After clustering stocks, these groups can be evaluated to identify factors that affect stock prices, such as economic factors, profitability, and market trends. These results can be used to optimize investment decisions or develop corresponding trading strategies. Therefore, "Clustering stocks to generate investment portfolio" was chosen as the topic for this research. In this study, the author will cluster stocks with similar characteristics together and use stocks from different clusters to build an effective and risk-diversified portfolio. The author expects that stocks with different financial characteristics will tend to have different volatility and risk levels, and vice versa. By combining stocks in a portfolio, investors can build an effective and risk-diversified portfolio.

1.2. Research scope

The spatial scope of the research is limited to the Vietnamese market and the financial data of Vietnamese companies listed on the HOSE (Ho Chi Minh City Stock Exchange) and HNX (Hanoi Stock Exchange).

The time frame for conducting the study is from February 2023 to April 2023. The time scope of the data used in the research is from January 1, 2017 to December 31, 2022. The financial ratios data, including ROE (Return on Equity), ROA (Return on Assets), P/E (Price to Earnings ratio), P/B (Price to Book ratio), Total Assets, and D/E (Debt to Equity ratio) will be collected at the end of the financial years from 2017 to 2021. The data on stock prices and index for HNX and HOSE are collected from 2018 to 2022.

The dataset will be applied with the K-means clustering algorithm and the author will evaluate whether this method can effectively cluster stocks based on their financial ratios to create an efficient portfolio for diversification. The data set is obtained from reliable sources and is referenced from relevant documents and previous studies.

1.3. Research Objectives

The objective of this study is to assist investors in saving time and providing more accurate results when analyzing stocks based on various financial characteristics to generate

an investment portfolio. The study utilizes the K-means algorithm to cluster stocks with similar financial characteristics into a group, allowing investors to use stocks from each cluster to diversify their investment portfolio.

The main focus of this study is on clustering stocks with similar characteristics using machine learning since there are few studies on this topic in Vietnam, while there are many studies on optimizing investment portfolios.

2. Literature review

2.1. Theoretical review

2.1.1. Modern Portfolio Theory

Modern Portfolio Theory is an investment management theory introduced by Harry Markowitz in 1952. This theory mainly focuses on risk, return, variance, standard deviation, and portfolio investment Lực, V. T. (2011). In particular, the theory emphasizes the use of diversified investment portfolios to minimize risk and optimize returns. According to this theory, investors should not bet on a single investment, but instead allocate their assets to multiple investments. Asset allocation is based on factors such as investment objectives, investment time horizon, and investor risk tolerance.

According to this theory, the investment portfolio is calculated as follows:

Expected Return = $w_1 . r_1 + w_2 . r_2 + ... + w_n . r_n$

Where:

w_i is the weight of each investment in the investment portfolio.

r_i is the rate of return of each investment in the investment portfolio.

According to efficient portfolio theory, investment efficiency is not only based on returns, but also on risk considerations. Therefore, investors need to allocate capital to investments with different risk levels and not concentrate too much on a single type of asset or industry.

The modern portfolio theory also suggests that investors should choose investments that are not correlated with each other to achieve investment diversification. By doing so, if one investment incurs risk, other investments can still offset it. According to this theory, diversification is considered one of the most important methods for reducing investment risk and maximizing profit. This will help reduce the overall risk of the investment portfolio, because when one type of investment is in trouble, other types of investments can help compensate and minimize the decline in the value of the investment portfolio.

In this study, the author will apply the idea of not investing in a single asset, but allocating assets to different investments to reduce risk and maximize profit, and the idea of investments

that are not correlated with each other to achieve investment diversification and calculation of expected return. The different investments referred to here are the stocks that have been clustered after applying the K-means algorithm and have different characteristics, which are expected to be uncorrelated in terms of risk and return. Therefore, building a stock portfolio from these stocks is expected to help reduce risk.

2.1.2. Behavioral finance

Behavioral finance is an important field in finance that studies how human psychology and behavior influence investment decisions and asset prices. It emphasizes understanding and explaining irrational trends and behaviors in financial markets, uncovering hidden factors in investment decision-making, and predicting future investor behavior.

The theory of behavioral finance stems from the idea that investors do not always act rationally and do not accurately evaluate the true value of assets. Instead, they are often influenced by emotions, fear, inaccurate expectations, and other psychological factors. Researchers in this field focus on analyzing irrational decisions and how they impact market performance.

The principles of behavioral finance are based on the following:

Existence of irrational behavior: Behavioral finance theory suggests that investors often exhibit irrational behavior, such as incorrect evaluations, cognitive limitations, conservatism, and other behaviors.

Systematic existence of irrational behavior: Behavioral biases in financial behavior are quite common among individual investors, creating a "herd effect" that causes stock prices to not reflect true values.

Limiting the possibility of arbitrage in the financial market: Contrary to the efficient market theory, behavioral finance theory argues that the costs of implementing profit-seeking strategies and the existence of investors hinder this mechanism. The adjustments mentioned in efficient market theory do not occur instantaneously in reality, but often persist for many years, indicating limitations in arbitrage opportunities.

Applying behavioral finance theory to explain individual investor behavior in the Vietnamese stock market is not unfamiliar. This behavior can be explained by the reliance on anchor-based investment psychology. In such cases, individual investors consider the world stock index, such as the Dow Jones index, as an anchor or reference point to make predictions about the Vietnamese stock index. For example, when investors observe an increase in the Dow Jones index today, they may believe that the VN-Index will also increase tomorrow or in the next few days. As a result, they make the decision to buy stocks, despite the lack of a scientific basis for the relationship between these variables moving in the same direction.

Another common behavior is investment decisions based on disclosed information. This information can come from people around them or from news sources. Thus, news serves as both a reference source for decision-making and as a basis for forecasting the trend of the VN-Index, guiding investors' investment behavior. Hiển, N.Đ (2012).

In conclusion, behavioral finance theory helps explain irrational behavior of investors and how psychological and social factors influence investment decisions. Applying this theory to study individual investor behavior in the Vietnamese stock market can enhance our understanding of behavioral patterns and explain market fluctuations in this country. This research helps to minimize the irrational behaviors of individual investors when participating in the stock market.

2.2. Extant literature review

Mansoor Momeni, Maryam Mohseni, Mansour Soofi (2015) conducted a study comparing the effectiveness of K-means and Hierarchical clustering in stock classification based on financial ratio such as EPS, P/S, ROE, ROA for three industries in 2012 and the listed stocks on the Tehran Stock Exchange. The results showed that K-means performed better than hierarchical clustering and ROA was the most important ratio in the clustering model, followed by EPS, ROE, P/S. The results from the model showed that investors could find optimal investment opportunities based on clustering results.

Iwan Fadilah, Rini Setyo Witiastuti (2018) examined the formation of an optimal investment portfolio using clustering methods. The data used were financial reports and stock prices of companies listed on LQ-45 during the period of 2012-2016. The ratios used for clustering were ROE, ROA, company size, returns, and variances of each stock. The data tested whether it was normally distributed. Author group used ROA, ROE, and company size calculated as log of asset to perform K-means clustering. The clustered stocks then formed a portfolio for diversification. The diversified clusters must have more than one stock and use the Sharpe Index, Jensen Index, and Treynor Index to evaluate the portfolio's performance. The results showed that portfolio three, consisting of ASII (3.96%), BBCA (21.39%), BBNI (5.84%), BBRI (21.50%), BMRI (30.32%), INDF (3.97%), and TLKM (13.03%) optimized returns and minimized risk.

Preeti Baser and Jatinderkumar R. Saini (2015) propose a method of portfolio management using non-hierarchical clustering. The data consists of financial ratios of 50 companies from the Nifty stock exchange and the variables such as EPS, DPS, P/E, BV and average daily returns for 5 years in the period 2012-2013. In the study, k-means, k-medoids, and fast k-means algorithms were used and the DB index was used to evaluate the effectiveness of the model. The results showed that the k-means algorithm produced more compact clusters than the other methods. Finally, the clusters generated by k-means were used

for investment and portfolio analysis using the Markowitz model. Thus, stock clustering helps investors optimize portfolio framing and have better risk-return profiles.

B. Kalyan Kumar and P. Soundararajan (2014) studied the use of clustering methods to classify stocks based on financial ratios. This study used six different financial ratios to analyze 20 stocks in the Indian stock market from 2008 to 2013. These ratios include EPS, P/E, P/B, ROE, ROA, and NPM (Net profit margin). Then, clustering methods were used to cluster the stocks into similar groups. In this study, clustering methods used included PCA (Principal Component Analysis), hierarchical clustering, and k-means. The results of the study showed that k-means is the best method for classifying stocks based on financial ratios. The study also showed that groups classified based on different financial ratios can provide useful information about the characteristics and trends of the stock market.

Bilgehan Tekin and Fatih Burak Gümusss (2017) conducted a study to select the most reasonable stocks and increase profits by minimizing human intervention. In this study, stocks were classified based on financial index ratios from companies' financial reports. Variables used in the study included the price-to-earnings ratio, market value/book value ratio, dividend yield, return on assets, return on equity, changes in sales and equity in 2015 and average profits, earnings, and risk in April 2016. The results showed that 88 stocks in the Borsa Istanbul 100 index were divided into 12 clusters. Among these stocks, the most suitable ones to form an investment portfolio were determined based on financial ratios and stock performance over one and three years.

R. Yusuf, B.D. Handaria, G.F. Hertono (2019) clustered 40 different stocks based on their financial ratio scores: Current ratio, debt to equity ratio, profit margin, return on equity, price/earnings per growth, diluted earnings per share, and price/earnings ratio using a fuzzy algorithm. After clustering the stocks, the agglomerative algorithm was applied to each cluster to determine the proportion of each stock. The author group also assumed that the profit of the asset risk was fuzzy. The study showed porfolio has a return (29.77%) and Sharpe ratio (18.71) higher than the S&P 500 index during the same period, which were 12.34% and 2.7%, respectively.

Shu Bin (2020) conducted a study using the Sharpe ratio and clustering results from the K-means algorithm based on financial ratios such as ROA and asset turnover and historical price index to generate an optimized risk portfolio during the early stage of the Covid-19 economic recession (2-3-2020 to 4-14-2020) of 232 companies. The author used the Silhouette method to determine the number of clusters. After clustering, the stocks with the highest Sharpe ratio from each cluster were selected to build the investment portfolio. The investment portfolio was built with the clustering results of poorly performing stocks based on their

historical price activity, but the portfolio built with the clustering results of companies based on financial ratios always exceeded the market average.

S.R. Nanda, B. Mahanty, and M.K. Tiwari (2010) conducted a study on stock clusters using K-means, Fuzzy, and SOM algorithms. Stocks can be selected from these cluster to build investment portfolios and help minimize risk by diversifying the investment portfolio. The variables used for clustering include Price earning (P/ E), Price to book value (P/BV), Price/cash EPS (P/CEPS), EV/EBIDTA, Market cap/sales, and short-term and long-term yields of listed companies on the Bombay Stock Exchange in 2007-2008. The analysis results show that K-means is the most efficient clustering analysis for classifying securities compared to SOM and Fuzzy for this dataset. Randomly selecting stocks from these clusters to build investment portfolios and implementing the Markowitz theory to determine the proportion of each stock can minimize investment portfolio risks and compare returns with the Sensex. The results show that the portfolio revolves around the Sensex and has higher returns than the market. This demonstrates that the application of K-means algorithms and Markowitz theory to diversify risks is effective and helps investors save time when selecting stocks.

Bakti Siregar1 and F. Anthon Pangruruk (2021) studied the K-means clustering method used to classify stocks listed on LQ45 and selected stocks with a tendency to increase in price from January 2015 to September 2021. Then, the Markowitz method was used to analyze the performance of optimal investment portfolio models with minimum variance in expected returns and risks. After knowing the performance of this optimal investment portfolio model, these methods can be applied in cloud computing or artificial intelligence. In addition, investors will develop a better understanding of the latest performance of stocks listed on LQ45 and support them in deciding which stocks to include in their investment portfolio, thus preventing wrong decisions.

Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, and Dr. Kavita Suryawansh (2018) used the K-means algorithm to cluster stocks listed on the BSE100 to create an investment portfolio. The group used the P/E and Operating Profit Margin (OPM) ratios to cluster at the beginning of 2017. They chose the Elbow method to determine the number of clusters and then used stocks closest to the center point to build the investment portfolio for the following year on a monthly basis. The result was that the stocks were divided into 4 clusters and at the end of the fiscal year in March 2018, the Sensex provided a return of 11.3%, the BSE 100 provided a return of 10.62%, while the investment portfolio returned 27.51%.

Karina Marvin (2015) used the total revenue to ssset ratio and ROA by quarter of publicly traded companies on the NYSE and NASDAQ from 7/2000 to 7/2015 and the K-means algorithm to cluster. The study period was divided into three stages: before, during, and

after the economic recession. The author clustered stocks and decided on the number of clusters based on the lowest SSE achieved by financial ratio. Then, for each stock the author selected the one with the highest Sharp ratio to construct an investment portfolio. The clustered investment portfolio results had higher volatility than the S&P 500 as expected. However, the values were still close, and the investment portfolio volatility remained low, usually around 2-3%, except during the 2008 financial crisis.

From the literature review of previous studies, the author decides to use the following variables: ROE, ROA, P/E, P/B, D/E and company size to cluster based on the K-means algorithm and select a reasonable cluster value based on two methods SSE - Elbow and Silhouette. When clustering, the author will expect stocks in the same cluster that have similar financial characteristics and tend to have different levels of volatility compared to stocks in other clusters. From there, the author can select stocks from different clusters to diversify the investment portfolio. Most previous studies only conducted research on financial ratio within one year, which may not fully reflect the situation and may not be suitable for the next years. In addition, some studies only conducted research based on two financial ratio. Therefore to address these issues, the author will collect data over five years with multiple ratios for analysis.

3. The research methods

3.1. Overview methods used in the study

3.1.1. K-means clustering algorithm

The K-means clustering algorithm is one of the simplest and most popular data clustering algorithms in machine learning and data mining. This algorithm is used to classify data points into clusters based on their features.

To cluster data using K-Means Clustering, we first choose k as the number of clusters to divide and randomly select k from the m initial data points as cluster centers $\mu_1, \mu_2, \ldots, \mu_k$. Then, for data point $x^{(i)}$, we assign it to cluster $c^{(i)}$ which is the cluster whose center is closest to it.

$$c^{(i)} = argmin_k \parallel x^{(i)} - \mu_k \parallel^2$$

Once all data points have been assigned to clusters, the next step is to recalculate the positions of the cluster centers by taking the average coordinates of the data points in that cluster.

$$\mu_k = \frac{1}{n}(x^{(k_1)} + x^{(k_2)} + \ldots x^{(k_n)})$$

where $k_1, k_2, \ldots, k_k$ are the indices of the data points belonging to cluster k.

The steps above are repeated until the cluster center positions remain unchanged after a certain iteration. The k-means algorithm works as follows:

Randomly initialize k initial data points to represent k initial clusters.

Repeat the labeling and centroid calculation process until there is no change in labeling.

Assign each data point to the nearest cluster (according to Euclidean distance) with the centroid.

Recalculate the centroid for each cluster by taking the average of the data points in that cluster.

Repeat steps 3 and 4 until there is no change in labeling.

In this study, the author will use the point closest to the cluster center to construct a stock portfolio. Therefore, the author uses the Euclidean distance formula to determine this point.

$$d(x,m) = \sqrt{(x_1 - m_1)^2 + (x_2 - m_1)^2 + \ldots + (x_n - m_n)^2}$$

Where:

$x_1, x_2, ..., x_n$ are the values of the attributes of point x

$m_1, m_1, ..., m_n$ are the corresponding values of the attributes of cluster center m

The smaller the distance d(x,m) between point x and cluster center m, the closer the point is to the cluster center. Therefore, the point closest to the cluster center will have the smallest value.

3.1.2. Evaluating the effectiveness of clustering algorithm

3.1.2.1. Sum of Square Errors (SSE) method

The Sum of Square Errors (SSE) method - Elbow is used to determine the optimal number of clusters by choosing the number of clusters in such a way that the reduction in the total distance between data points and cluster centers is the highest.

These distances are measured by the squared Euclidean distance between the data points and their corresponding cluster centers. The optimal number of clusters is determined using the Sum of Square Errors (SSE) evaluation method, also known as the Elbow method, by selecting the number of clusters at which the reduction in the total distance between data points and their cluster centers becomes less significant with increasing cluster numbers. After computing the SSE for each k value, the author visualize and identify the elbow point on the SSE curve, which indicates the optimal k value for clustering (Rokach & Maimon, 2005 cited in Duda et al., 2001).

SSE, also known as distortion, is calculated using the following formula:

$$SSE_n = \sum \text{dist}(p, c)^2$$

Where:

p is a data point in cluster c

c is a cluster

dist (p, c) is the Euclidean distance between data point p and cluster center c

k is the number of clusters.

3.1.2.2. Silhouette method

The Silhouette method for selecting the number of clusters is used to evaluate the quality of clustering by computing the Silhouette value for each data point and for the entire dataset. The Silhouette value is calculated by comparing the distance between the data point and the points in its current cluster with the distance between the data point and the points in the nearest cluster it does not belong to (Hoss Belyadi, Alireza Haghighat, 2021).

The Silhouette value ranges from -1 to 1, with a value closer to 1 indicating better clustering and a value closer to -1 indicating poorer clustering. The dataset is divided into k clusters, where the optimal value of k corresponds to the highest Silhouette value.

The Silhouette value for a data point in a cluster is calculated using the formula:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Where:

S_i is the Silhouette value of data point i

a_i is the average distance between data point i and the points in the same cluster as i

b_i is the average distance between data point i and the points in the nearest cluster to i (excluding cluster i).

Both the Silhouette and Elbow methods are useful for determining the optimal number of clusters for K-means clustering. The combination of these two methods can be achieved by using the SSE graph and Silhouette value to evaluate the optimal number of clusters. When selecting the number of clusters, author choose the number of clusters corresponding to the elbow point on the SSE graph and ensure that the average Silhouette value of those clusters is high. This will help to improve the accuracy of clustering.

3.1.3. Data preprocessing methods

3.1.3.1. Min-max normalization

Min-max normalization is a data normalization method in data science and machine learning. When data is not in the appropriate format for training machine learning models, we need to normalize the data to bring it to the same range of values. Normalization helps the

model to converge faster, reduce computation time, and increase the accuracy of the model. The Min-Max normalization method simply normalizes the data to the range [0,1] using the following formula:

Where:

$$\text{X\_norm} = \frac{(\text{X} - \text{X\_min})}{(\text{X\_max} - \text{X\_min})}$$

X is the original value of the variable

X_min is the smallest value of the variable in the data

X_max is the largest value of the variable in the data

X_norm is the normalized value

The result obtained after normalization will be in the range [0,1] and maintains the proportion between the initial values. This can help balance the contribution of features in calculating the distance between data points, avoiding the phenomenon of features with large values dominating the clustering process.

This process can help balance the contribution of features in calculating the Euclidean distance between data points, making the K-means algorithm more efficient and yielding better clustering results.

3.1.3.2. PCA (Principal Component Analysis) technique

PCA (Principal Component Analysis) is a common dimensionality reduction technique in machine learning and other fields. PCA helps to reduce the dimensionality of the data while still retaining the most important information. This technique helps to reduce computational costs, speed up training, and reduce overfitting.

The idea of PCA is to find the principal components of the data so that the data points are evenly distributed on these axes. These principal axes are arranged in decreasing order of importance and form a new coordinate system. We can reduce the dimensionality of the data by retaining only the most important principal components.

To find the principal components, we need to calculate the covariance matrix of the data, then solve the orthogonal equation to find the eigenvectors of this matrix. These eigenvectors form the principal components. The eigenvalues of the covariance matrix indicate the importance of the principal components. After obtaining the principal components and eigenvalues, we can calculate the new data points on this new coordinate system by multiplying the original data matrix with the normalized orthogonal matrix of the eigenvectors.

In this research, the author will use this method to help observe how stocks are represented in 2D (2 dimensions).

3.2. Measurement of variables used in the study

There are 6 variables used in this research: ROE, ROA, D/E, P/B, P/E, SIZE.

ROE (Return on Equity) is a financial ratio that measures a company's ability to generate profit from shareholder equity. ROE is calculated by dividing after-tax profit by shareholder equity of the company. The formula for calculating ROE is performed as follows:

$$\text{ROE} = \frac{\text{Net income}}{\text{Equity}}$$

ROE (Return on Equity) is a financial ratio that measures a company's ability to generate profit from its shareholder's equity. ROE is calculated by dividing net income after taxes by shareholder's equity of the company. ROE is an important ratio in evaluating a company's financial performance, as it shows the ability of the company to generate profit for its shareholders. A high ROE typically indicates that a company is able to generate good profit from its shareholder's equity, while a low ROE may indicate risk in investing in that company. However, ROE can also be affected by other factors such as bond yields, debt-to-equity ratio, and asset structure, so it needs to be combined with other ratios to comprehensively evaluate a company's financial performance.

ROA (Return on Assets) is a financial ratio that is calculated by dividing net income after taxes by total assets of a company or organization. This ratio measures the ability of a company to generate profit from the assets it uses. It is often used to evaluate the operational performance of a company, especially in the banking and finance industry. The higher the ROA, the more efficient a company is in using its assets to generate profit. However, ROA can also be affected by other factors such as asset structure and cost of capital, so it needs to be combined with other ratios to comprehensively evaluate a company's financial performance. The formula for calculating ROA is performed as follows:

$$\text{ROA} = \frac{\text{Net income}}{\text{Total assets}}$$

P/E (Price-to-Earnings ratio) indicates the market valuation of each unit of profit generated by a company. Typically, P/E is evaluated in conjunction with other factors such as the company's growth, investment plans, and competition in the industry to make accurate investment decisions. P/E can be calculated in various ways, but in this study it is calculated using the following formula:

$$\text{P/E} = \frac{\text{Market capitalize}}{\text{Net incom after tax}}$$

The P/B (Price-to-Book ratio) is a ratio used to compare the price of a stock to its book value. This ratio is calculated by dividing the current closing price of the stock by the book value at the most recent quarter of that stock. P/B is calculated using the following formula:

$$\mathrm{P/B} = \frac{\text{Price}}{\text{Book value per share}}$$

D/E (Debt-to-Equity ratio) is the percentage of the company's capital raised from borrowing activities compared to the equity capital. This is the ratio of debt to equity, used to assess a company's financial leverage. P/E is calculated using the following formula:

$$\mathrm{D/E} = \frac{\text{Debt}}{\text{Equity}}$$

SIZE is calculated using the following formula:

$$\text{SIZE} = \ln(\text{Total assets})$$

In this study, the researcher will use these measurement methods to measure the variables in the study.

3.3. Data

3.3.1. Data description

The data in this study includes the financial ratios of companies listed on the two stock exchanges HNX and HOSE for 5 years from 2017 to 2021. In addition, the author also collected the closing prices of these companies and the two exchanges HNX and HOSE from 2018 to 2022 to evaluate the effectiveness of portfolio construction.

The reason for selecting stocks listed on the HOSE and HNX exchanges is because these exchanges have a high reputation in the Vietnamese stock market. Stocks listed on these two exchanges often have high liquidity and financial information is disclosed clearly.

Due to the increasing number of listed companies every year, the study sample will change annually and the results after removing empty values in the dataset are as follows:

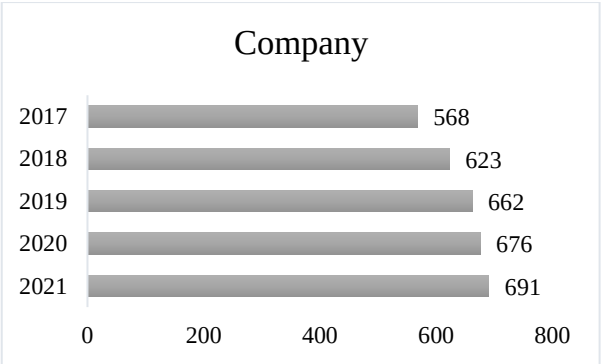

Figure 1. Number of companies included in the study from 2017 to 2021.

3.3.2. Data collection method and data source

Data was collected from financial reports of companies and stock data based on reliable sources - Thomson Reuters and cross-checked with other sources such as Vietstock and CafeF.

3.3.3. Data processing

In data processing, the author first removed missing values from the dataset and merged the tables together. Then, the variables ROE, ROA, P/E, P/B, D/E, and SIZE were measured as in section 3.2. After removing the missing values, the author divided the dataset into different years for analysis.

After performing descriptive statistics, the author realized that the dataset contained outliers and thus removed them to ensure the effectiveness of the K-means clustering algorithm. According to Agnieszka Nowak-Brzezińska and Igor Gaibei (2022), a dataset containing outliers will negatively affect the quality of the clusters created. By removing the outliers, the resulting clusters will be of higher quality and therefore more effective in exploring the dataset. Another benefit of detecting outliers is that it reduces the time required for clustering (as there are no difficulties in forming the clusters).

Before applying the K-means clustering algorithm, the author found that most of the attributes were not normally distributed, so the author normalized them using the Min-max normalization method. According to Deepali Virmani, Shweta Taneja, Geetika Malhotra (2015), research shows that when data is normally distributed, clustering algorithms work more effectively. The variables normalized by the Min-max normalization method will be calculated using the following formula:

$$\text{X\_norm} = \frac{(\text{X} - \text{X\_min})}{(\text{X\_max} - \text{X\_min})}$$

At this point, the financial index dataset was ready to apply the K-means algorithm for clustering.

3.4. Method

After the dataset has been cleaned, outliers have been removed and normalized, the author performs the K-means clustering method to cluster the stocks. To determine the number of clusters (k), the author will rely on 2 methods: SSE (Elbow) and Silhouette, as suggested by some references such as Shu Bin (2020), Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, Dr. Kavita Suryawansh (2018), and Karina Marvin (2015).

Once the stocks have been clustered into groups with similar financial characteristics, the author expects that the stocks within each group will have different risks and returns, and there will be no correlation between them. This can help to generate an investment portfolio. To generate a portfolio, the author will select stocks that are closest to the center of the cluster. According to Nguyen Cong Long, Nawaporn (2014), stocks that are close to the center of the cluster will reflect and have all the characteristics of that cluster, and Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, Dr. Kavita Suryawansh (2018) also agree with this.

To evaluate whether these stocks are truly effective when combined to create an investment portfolio, the author will evaluate the portfolio's profitability after one year, according Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, and Dr. Kavita Suryawansh (2018). In addition, the author will compare its profitability with the market's profitability on a monthly basis. This method has been used in the studies of S.R. Nanda, B. Mahanty, M.K. Tiwari (2010), and Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, Dr. Kavita Suryawansh (2018).

The author used calculation of expect return which meantioned in 2.1 to calculate the investment porfolio. r_i will be calculated as follows:

$$r_i = \frac{P_t - P_{t-1}}{P_{t-1}}$$

where is the price of the current month, and is the price of the previous month.

In this research, the author is assuming that the weights of stocks in the investment portfolio are equal. This method has been used in the studies of Vivek Kedia, Zubayr Khalid, Dr. Saptarsi Goswami, Dr. Neha Sharma, and Dr. Kavita Suryawansh (2018). The method proposed here is to construct a portfolio to evaluate whether the selected stocks can be combined to create an efficient portfolio. If the portfolio generates profits in the future and outperforms the market then these stocks can be combined together to create porfolio.

4. Research results

In this section, the author will discuss the descriptive statistical results of the dataset and the clustering results using the K-means algorithm. From there, the author evaluates how the

investment portfolios will perform when combined. The results will be analyzed for each year in the period from 2017 to 2021.

4.1. Descriptive statistic

Table 1: Descriptive statistics

index	ROE	P/B	SIZE	ROA	D/E	P/E
count	2994	2994	2994	2994	2994	2994
mean	0.11	1.09	20.79	0.05	0.69	32.4
std	0.1	0.73	1.64	0.05	0.82	118.1
min	-0.34	0.07	16.53	-0.13	0	-702.18
25%	0.04	0.6	19.68	0.02	0.06	6.48
50%	0.09	0.92	20.73	0.04	0.38	10.37
75%	0.16	1.4	21.73	0.08	1.03	18.85
max	0.56	4.39	26.22	0.25	4.75	2086.1

Table 1 provides descriptive statistics of the variables included in the K-means model, including ROE, ROA, P/B, SIZE, D/E, and P/E. The dataset used in the study consisted of 2994 stocks divided into five different years. ROE had values ranging from -0.34 to 0.56, with an average of approximately 0.11. P/B ranged from 0.07 to 4.39 with an average of 1.09. SIZE had values ranging from 20.79 to 26.22 and a mean value of 20.79. The lowest and highest values of ROA were -0.13 and 0.25, respectively, with a mean value of 0.05. D/E had values ranging from 0 to 4.75 and a mean value of 0.69. Finally, P/E had values ranging from -702 to 2086, with an average of 32.4.

4.2. Cluster analysis results

Year 2017: Based on Figure 2, k = 3 is a reasonable choice for the 2017 dataset. When determining k using Elbow and Silhouette methods, both methods support the choice of k = 3. In the Elbow method, when k exceeds 3 and reaches 4, the average distortion value drops sharply.

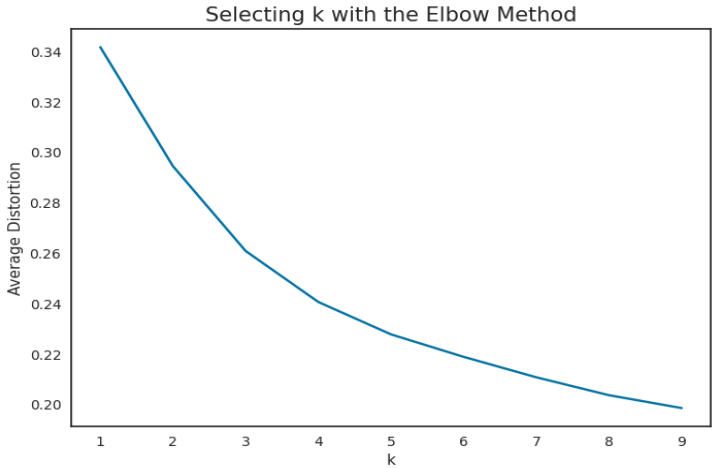

Figure 2. Average Distortion in 2017

Based on Figure 3, the average Silhouette value reaches its maximum at 3. Therefore, k is equal to 3 is the best choice.

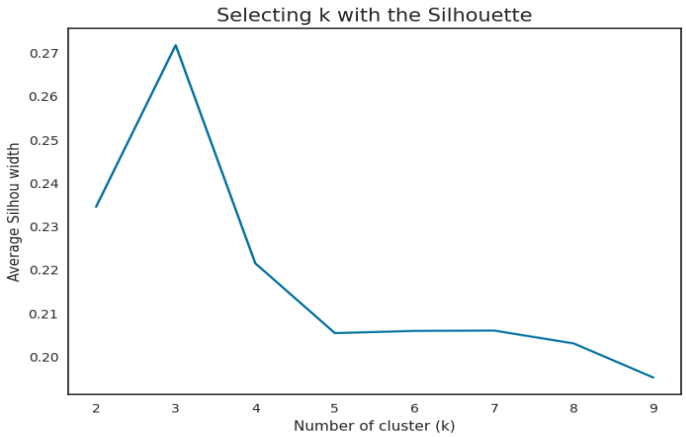

Figure 3. Average Silhouette width in 2017

Figure 4, the cluster's width exceed a relatively large Silhouette value of 0.29, but the width are not evenly distributed. This will lead to a significant difference in the number of stocks in each cluster. We also see that in clusters 1 and 2, some points are misclassified.

Figure 4. Silhouette width in 2017

The author will perform PCA on the data to observe the clusters more clearly in 2 dimensions. Based on Figure 5, it is clear that when k is equal to 3 the clusters are not too mixed but still not completely separated, and there are still some points that are not clearly classified.

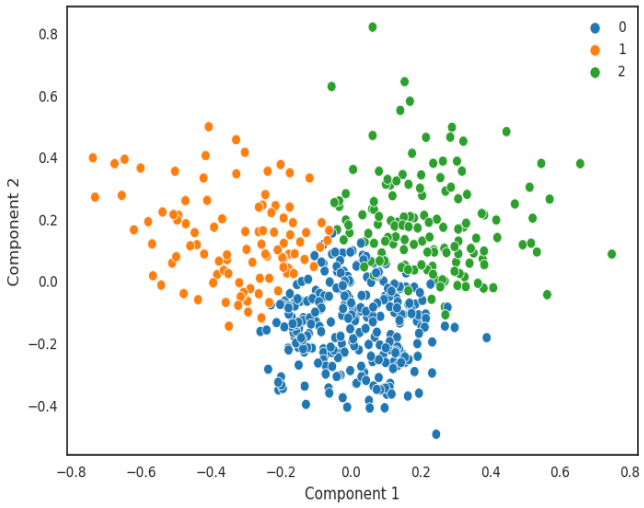

Figure 5. 2D PCA plot of stock clusters in 2017.

Table 2: Average values of ratio for each cluster in 2017

Cluster	ROE	P/B	SIZE	ROA	D/E	P/E	Count
0	0.09	0.79	20.08	0.05	0.3	20.69	271
1	0.12	0.97	21.59	0.03	1.55	11.59	140
2	0.22	2.08	20.56	0.12	0.27	11.52	100

When k is equal to 3, there are 271 companies in cluster 0, 100 companies in cluster 1, and 140 companies in cluster 2. The average values of ROE and ROA in cluster 2 are the highest, while cluster 0 has the highest P/E value and cluster 1 has the highest average values of SIZE and D/E.

Year 2018: Figure 6 shows that k can be either 3 or 4 as the elbow values are at these positions. However, to determine the exact number of clusters, the author will use the Silhouette method and the results in Figure 7 support a value of k = 3. Because k is equal to 3, average Silhouette is biggest (except k = 2).

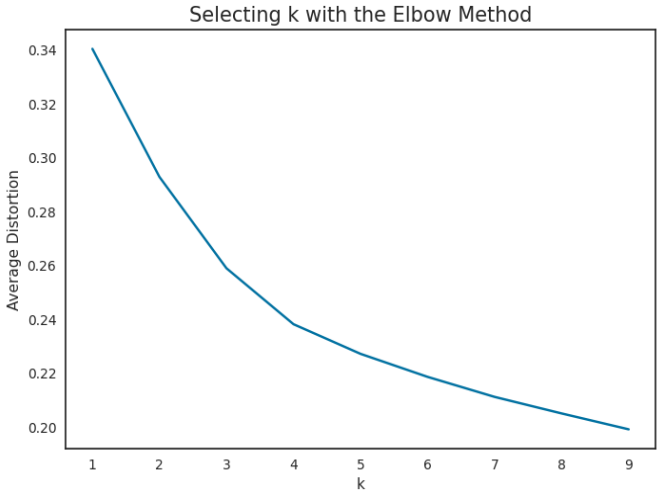

Figure 6. Average Distortion in 2018

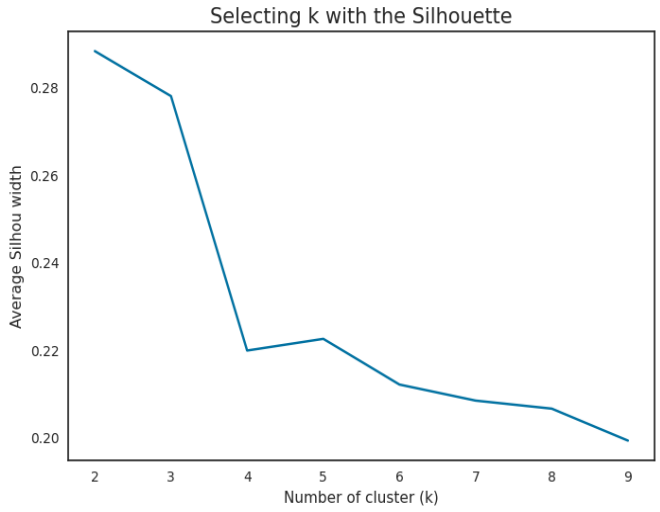

Figure 7. Average Silhouette width in 2018

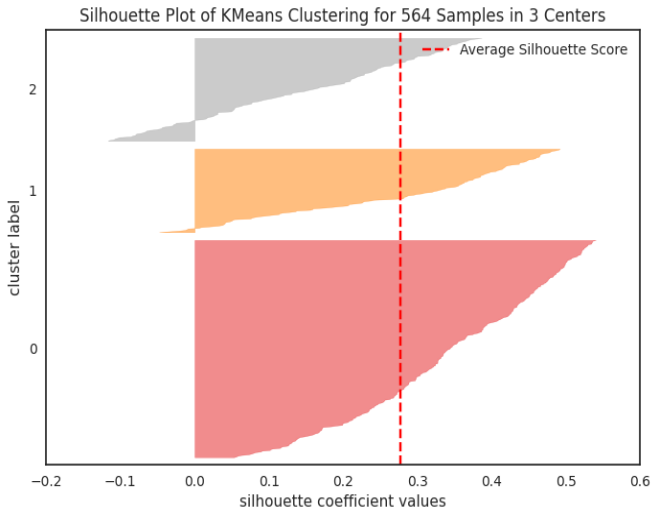

Figure 8. Silhouette width in 2018

When observing Figure 8, it is clear that the clusters's width always exceed the Silhouette average value, but some points are misclustered.

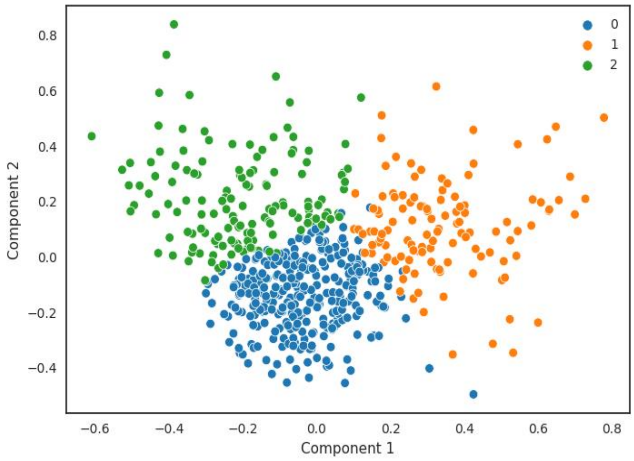

Figure 9. PCA plot of stock clusters in 2018

After performing 2D PCA, Figure 9 shows that three distinct clusters are separated from each other, but some stocks are still mixed together.

In 2018, the number of stocks increased to 564 with cluster 0 consisting of 303 stocks, cluster 1 consisting of 117 stocks, and cluster 2 consisting of 144 stocks. Cluster 0 had the highest concentration of stocks. Cluster 2 had the highest average values of ROE, P/B, and ROA, while cluster 1 had the highest average values of SIZE, P/E, and D/E. Cluster 0 had the lowest average values of ROE, ROA, P/B, Size, and D/E.

Table 3: Average values of variable for each cluster in 2018.

Cluster	ROE	P/B	SIZE	ROA	D/E	P/E	Count
0	0.07	0.85	20.19	0.04	0.32	15.44	303
1	0.08	1.05	21.49	0.02	1.92	16.08	117
2	0.22	1.99	21.27	0.12	0.4	9.53	144

Year 2019: In 2019, the author chose to divide the stocks into 3 clusters based on the Elbow method, as the average Distortion decreased significantly at k = 3 (according to Figure 10).

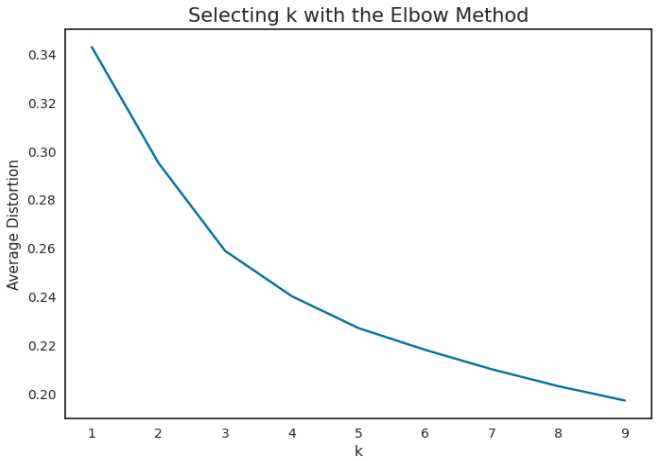

Figure 10. Average Distorition width in 2019.

At k = 3, the Silhouette value was the highest (although lower than at k = 2), according to Figure 11. But in this study, the author aims to cluster stock into more than 2 clusters.

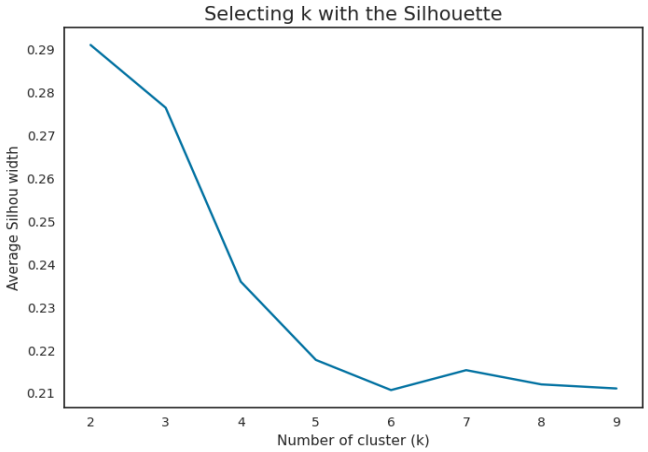

Figure 11. Average Silhouette in 2019

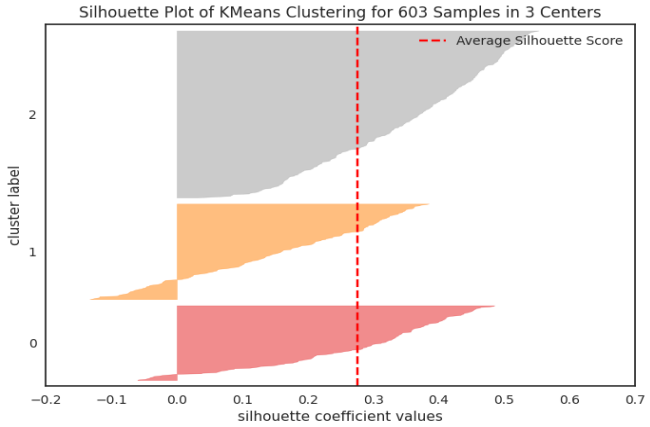

Figure 12. Silhouette width in 2019

Figure 12 shows that when the stocks were divided into 3 clusters, the clusters were not evenly sized, but each cluster had a Silhouette value above average.

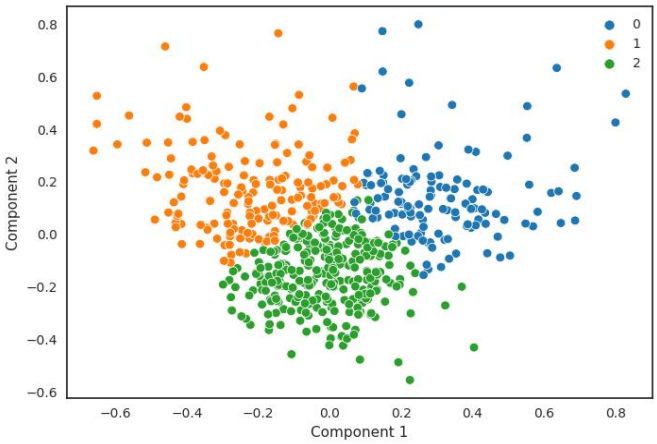

Figure 13. 2D PCA plot of stock clusters in 2019

The clusters in the PCA plot were clearly distinct, with only a few points overlapping with other clusters.

In 2019, there were a total of 603 stocks used for clustering. Cluster 0 had 134 stocks, cluster 1 had 171 stocks, and cluster 2 had 298 stocks. Table 4 showed that cluster 0 had the highest average values of SIZE and D/E. Next, cluster 1 had the highest average values of P/B, ROE, and ROA, while cluster 2 had the highest average value of P/E.

Table 4: Average values of variable for each cluster in 2019

Cluster	ROE	P/B	SIZE	ROA	D/E	P/E	Count
0	0.1	0.95	21.61	0.02	1.89	19.01	134
1	0.18	1.69	20.99	0.1	0.35	10.82	171
2	0.06	0.71	20.28	0.03	0.32	43.57	298

Year 2020

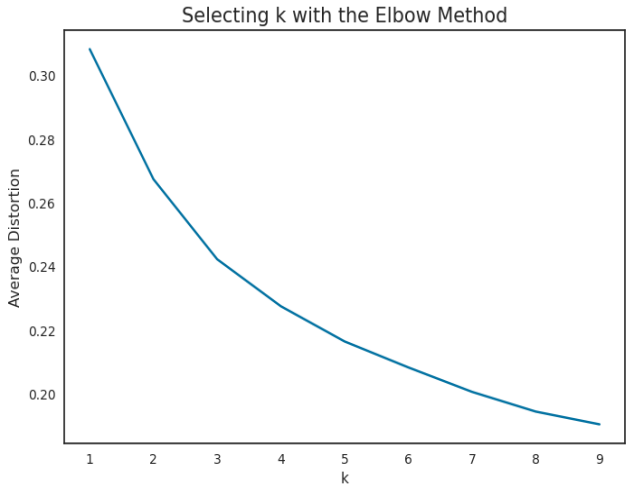

Figure 14. Average Distortion in 2020

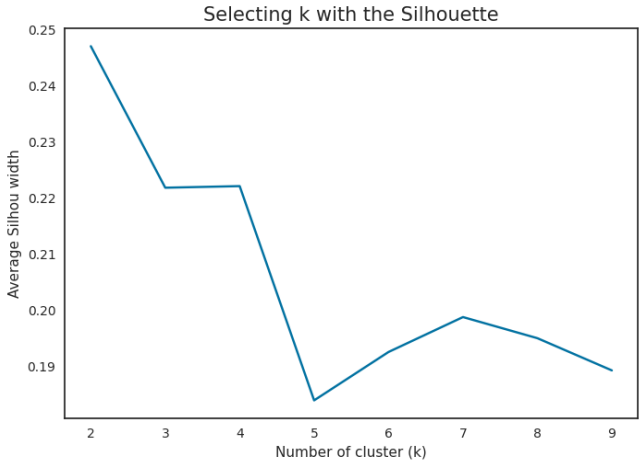

Figure 15. Average Silhouette in 2020

For the data in 2020, it was difficult to determine the value of k using the Elbow method based on Figure 14. However, based on Figure 15 using the Silhouette method, it was easy to

see that k = 4 had the maximum average Silhouette value (except for k = 2). Therefore, the author decided to divide the stocks into 4 clusters.

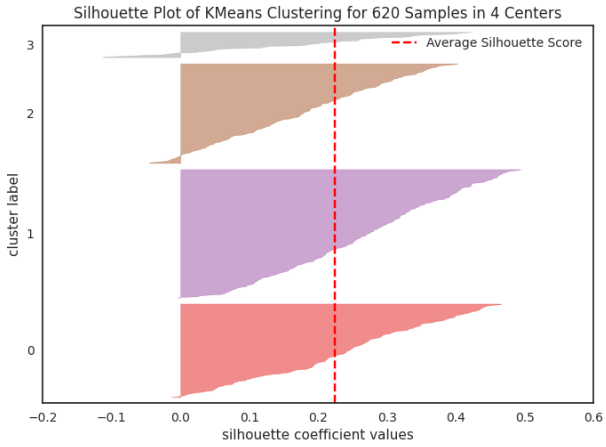

Figure 16. Silhouette width in 2020

When divided into 4 clusters, the clusters were fairly even, with only cluster 3 being much narrower than the other clusters. The clusters exceeded the average Silhouette value by a large margin based on Figure 16.

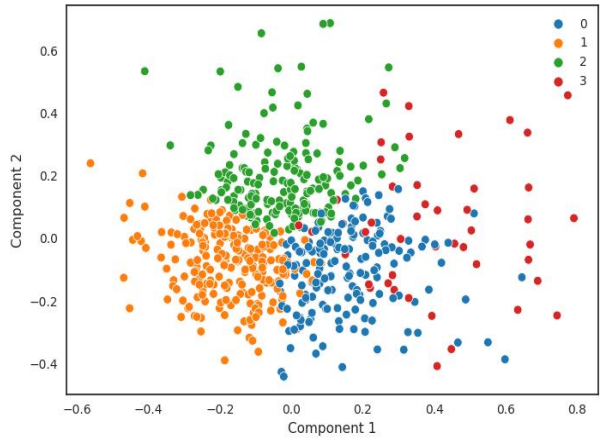

Figure 17. 2D PCA plot of stock clusters in 2020

Observing the PCA chart of the stocks in 2020 that were divided into 4 clusters, it appears that clusters 3 and 0 overlap with each other.

Table 5: Average values of variable for each cluster in 2020

Cluster	ROE	P/B	SIZE	ROA	D/E	P/E	Count
0	0.17	1.14	20.23	0.1	0.31	8.59	167
1	0.02	0.64	19.95	0.01	0.49	42.85	229
2	0.1	0.91	22.44	0.03	1.4	19.42	178
3	0.16	2.83	21.39	0.08	0.42	23.77	46

The results from table 5 show that after the stocks were divided into 4 clusters, there were 167 stocks in cluster 0, 229 stocks in cluster 1, 178 stocks in cluster 2, and 46 stocks in cluster 3. Cluster 1 had the most stocks and cluster 3 had the least. Cluster 2 had the highest average values for SIZE and D/E. Cluster 1 had the highest average values for ROE and ROA, and cluster 3 had the highest average value for P/B.

Year 2021

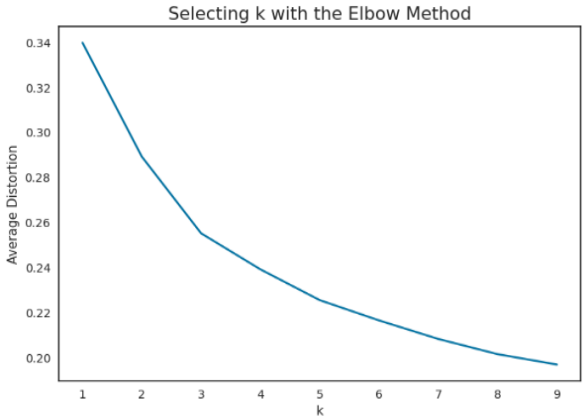

Figure 18. Average Distorition in 2021

Based on Figure 18, the author concludes that the Elbow method cannot provide a clear decision on whether k = 3 or k = 4 is the optimal number of clusters. When combined with the Silhouette method, the results from Figure 19 show that k = 4 is better because the average Silhouette value is higher than for k = 3. Therefore, the author decides to choose k = 4.

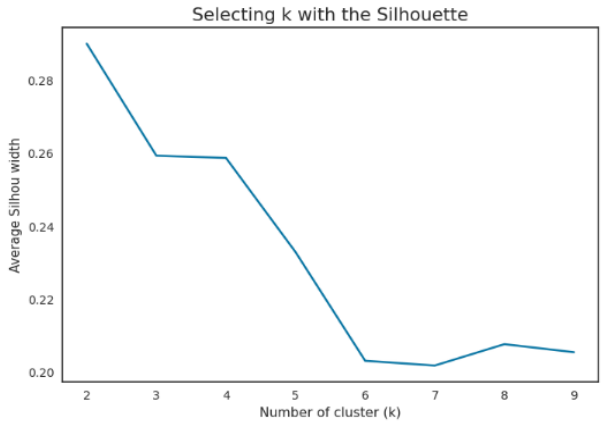

Figure 19. Average Silhouette width in 2021

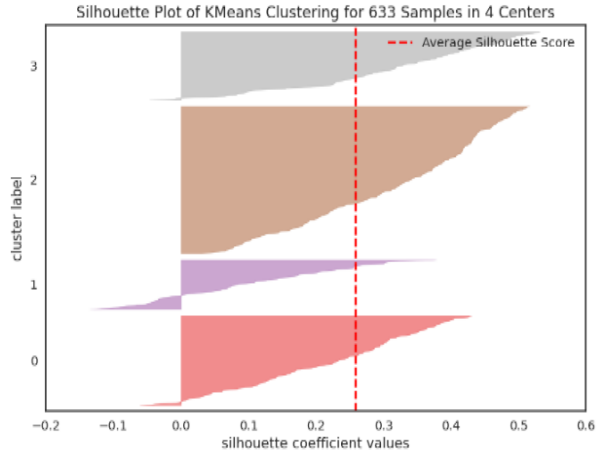

Figure 20. 2D PCA of the stock clusters in 2021

Upon observing the cluster's width in Figure 20, the author note that the clusters's width exceeded the average level, but the number of points in each cluster is uneven and some points were misclassified.

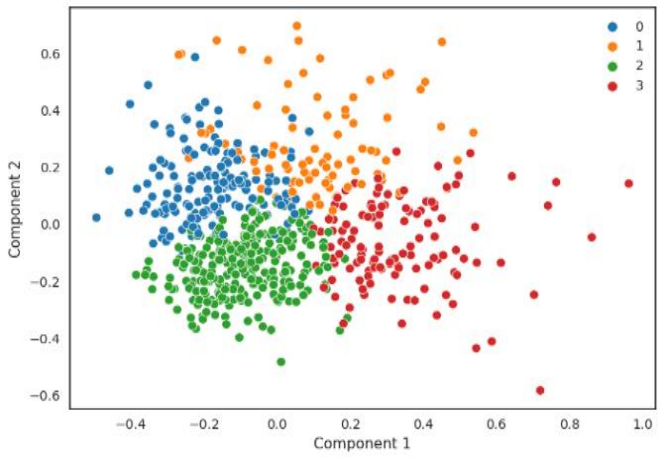

Figure 21. 2D PCA plot of stock clusters in 2021

This result is consistent with the results provided in Figure 21. Some points in the clusters are not clearly defined and there are four clusters.

Table 6: Average values of ratio for each cluster in 2021.

Cluster	ROE	P/B	SIZE	ROA	D/E	P/E	Count
0	0.21	1.31	20.61	0.12	0.21	10.55	160
1	0.13	2.31	22.77	0.06	0.68	59.19	89
2	0.05	0.73	20.12	0.03	0.31	62.9	262
3	0.1	0.89	21.76	0.02	1.88	23.02	122

Based on the results in Table 6, 633 stocks were divided into 4 clusters. Cluster 0 had 160 stocks, cluster 1 had 89 stocks, cluster 2 had 262 stocks, and cluster 3 had 122 stocks. The points in each cluster were not evenly distributed. When analyzing the average values of each cluster, the author found that cluster 0 had the highest average values of ROE and ROA, cluster 1 had the highest average values of P/B and SIZE, cluster 2 had the highest average value of P/E, and cluster 3 had the highest average value of D/E.

Overall evaluation for the period from 2017 to 2021

With the input data and analysis method used, the clusters are usually divided into 3-4 and have many different characteristics. When clusters have high average values of ROE, they tend to have high average values of ROA and vice versa. It can be seen that the clusters have very different financial characteristics.

When k is divided into 3, the clusters are more distinct than when k is 4. When k is 4, the clusters become less distinct, as seen in 2020 and 2021. The average Silhouette values over the years were always at around 0.27-0.3 - this is an acceptable result. This result surpassed some previous studies such as Bin, Shu (2020). However, it needs to be improved in the near future and the author will discuss this further in section 5.2.

4.3. Investment portfolio results

As mentioned in section 3 on the method of constructing an investment portfolio, the author will choose stocks closest to the center to build the investment portfolio for investment because these stocks fully reflect the characteristics of those clusters, are not disturbed and have a low likelihood of deviation. The stock portfolio is formed according to the formula in section 3.4.

In 2017, the three stocks closest to the cluster center were TTB, VC2, and INN for 2018.

The author assumed that the weights for each stock are equal, and then examine the total return of the portfolio compared to return of market and the return after one year is how much per month.

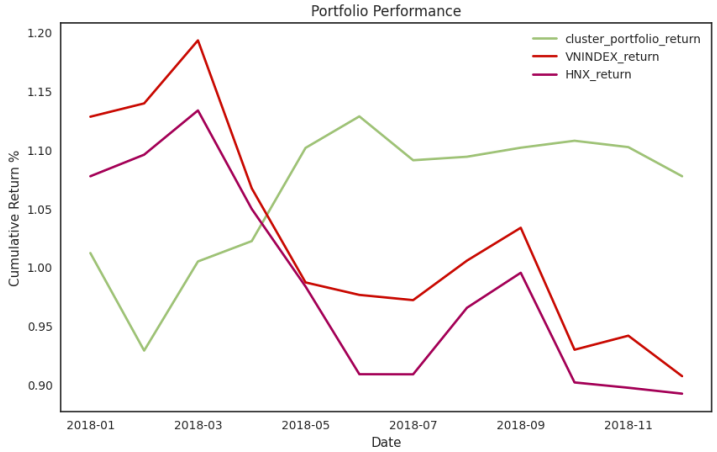

Figure 22. Efficiency of the clustered investment portfolio in 2018.

The results show that the total profit of the portfolio in 2018 has beaten the market and increased by about 7%. However, the portfolio had a negative return in the first quarter of 2018 and resumed growth in the following quarters. Meanwhile, return of market (HNX & VNINDEX) decreased sharply compared to March 2018. In general, the portfolio is efficient and can be used to build an optimized portfolio.

In 2018, the stocks used to build the investment portfolio were PCE, VTV, and VHL for 2019. These stocks are currently listed on the HNX exchange.

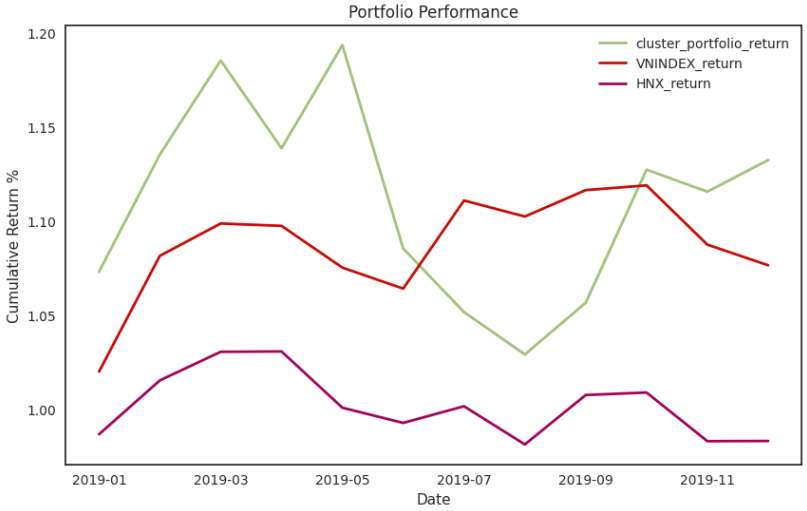

Figure 23. Efficiency of the clustered investment portfolio in 2019

Based on Figure 23, the investment portfolio fluctuates in the same direction as the return market (HNX). However, it has a higher efficiency and reached a level of nearly 15% at the end of the year compared to the decline of the HNX exchange and higher than the HOSE exchange (7%).

In 2019, the three stocks closest to the cluster center were TLD, SJD, and TVS. The result in 2020 was also high efficiency for this investment portfolio according to Figure 24.

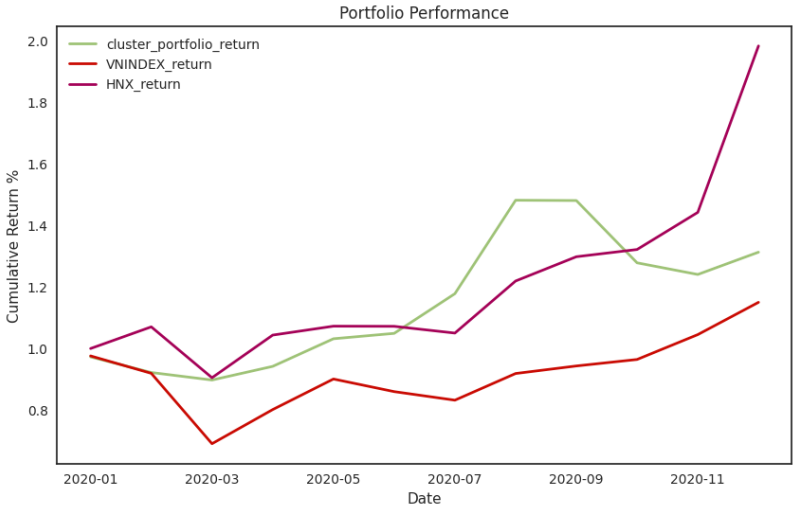

Figure 24. Efficiency of investment portfolios clustered in 2020

The return on investment of the portfolio after one year was nearly 30%, higher than return of market (VNINDEX). These stocks are listed on the HOSE exchange and come from various industries and can be considered for building an efficient portfolio in the future.

In 2020, the selected stocks for portfolio construction were NHH, UIC, VC7, and PLC for 2021.

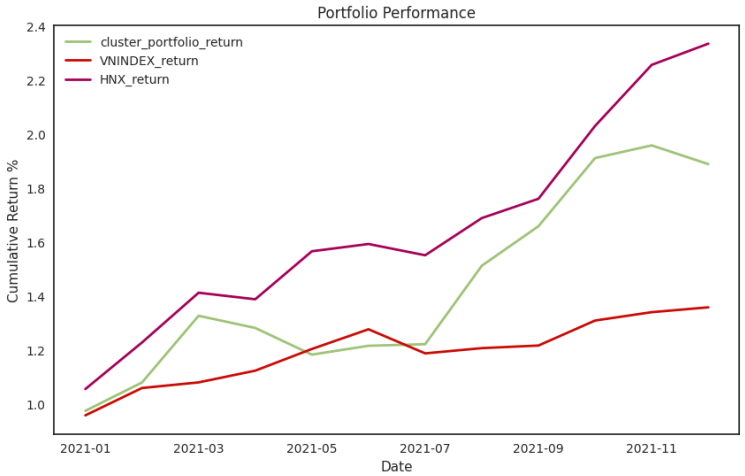

Figure 25. Efficiency of investment portfolios clustered in 2021

These stocks come from various industries and are listed on both the HOSE and HNX exchanges. According to the results in Figure 25, the portfolio's return after one year was nearly 90%, much higher than the VNINDEX and lower than HNX. These stocks can also be considered for building a portfolio.

Next, in 2021, the selected stocks for portfolio construction include the 4 stocks TLD, SJD, CMG, and SJE for 2022.

It can be seen that this portfolio completely failed in 2022, with a negative return of nearly 40% and revolving around the HOSE market return. These stocks are listed on both the HOSE and HNX exchanges and come from various industries. However, they are not ideal stocks to combine for building an efficient portfolio in the future.

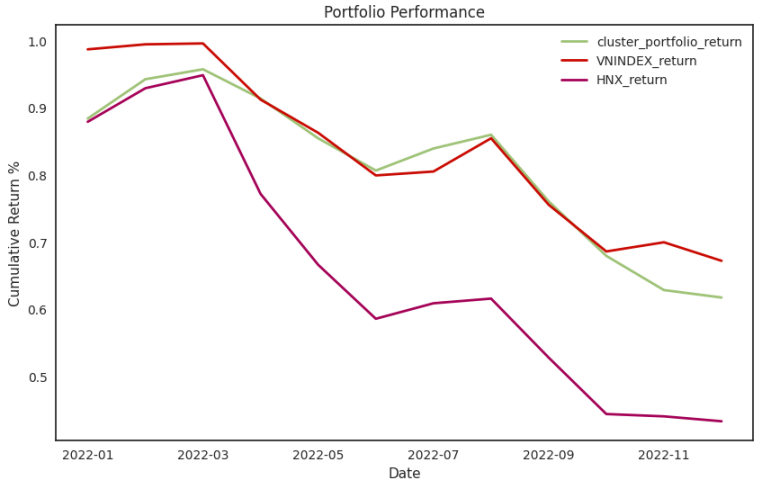

Figure 26. Efficiency of investment portfolios clustered in 2022

Overall assessment from 2017 to 2021

The stock market in 2018 witnessed the most severe volatility in 10 years since the global financial crisis of 2008. This was the first year that the stock market experienced a decline. However, the portfolio constructed by the machine learning algorithm outperformed and generated a profit of approximately 7%. The use of machine learning in selecting investment stocks proved effective in diversifying the portfolio and mitigating risks in 2018. After that, return of porfolio increased by about 14% and the market return relative recovery compared to the end of 2018. In 2020, from the second quarter to now, the market has recovered sustainably, grew spectacularly and lasted until the end of year and porfolio’s return reached more 20%. In 2021, stocks will become an important capital conduit for the economy, with many new records being set and market profits having grown strongly and porfolio’s return hitted record levels. The portfolio failed and the market crashed in 2022. This shows that machine learning has not really diversified risk yet. It can only partially support people when building a porfolio.

An effective investment portfolio requires investors to pay attention to the weighting of stocks in the portfolio to achieve maximum returns and minimize risk. However, in this study, the author used average weighting because the research purpose was to focus on clustering stocks to suggest for investors to refer to and build their own portfolios.

5. Conclusion and Recommendations

5.1. Conclusion

Based on the results of this study, the K-means algorithm can help investors cluster stocks with similar financial characteristics to build investment portfolios and diversify risks. This result is consistent with other studies, such as Nguyen Cong Long, Nawaporn (2014). Clustering using machine learning helps investors save time and achieve a level of accuracy. The evidence that in each year, the number of stocks used to build investment portfolios was reduced from 500-600 to 3-4 stocks. This result is also similar to the study by S.R. Nanda, B. Mahanty, M.K. Tiwari (2010) the number of stocks was reduced from 69 to 3.

Using the ROA, ROE, P/B, P/E, D/E, and SIZE ratios have helped the K-means algorithm operate relatively effectively. Investment portfolios built by machine learning based on these financial characteristics have generated profits over 4 years of analysis. However, this method is not yet fully optimized as the investment portfolio built by machine learning failed in 2022. Clustering stocks over the years based on various financial ratios helps provide a more comprehensive view because the method applied in the study is not 100% successful.

To achieve an efficient investment portfolio, investors must consider the weight of stocks in the portfolio to achieve maximum return and minimize risk. However, in this study, the author used average weights because the purpose of the research was to focus on clustering stocks to provide suggestions for investors to reference and build their own portfolios. Additionally, the Silhouette index only reached about 0.3, which is a clustering level that is not really clear and separate, although it is higher than some previous studies.

Besides that, in this study, not all stocks in the three clusters were fully evaluated, and only one stock was selected as a limitation. In the future, the author will further develop this research by constructing a portfolio that combines not only the closest stocks in each cluster but also evaluates all stocks within each cluster when combined in various ways, such as testing random iterations for each cluster. Using financial indicators to forecast for the following year does not guarantee that the predictions will accurately reflect the actual circumstances. There are several reasons why forecasting financial indicators may be inaccurate or fail to capture specific fluctuations in stock prices in the future. Financial indicators are based on data and information that have occurred in the past. However, in the future, new events and fluctuations may occur that are not reflected in the past data. Therefore, relying solely on annual forecasts may limit the ability to estimate and predict new fluctuations

in the future. Furthermore, another limitation is that comparing the returns of a portfolio to the market may not be appropriate, despite previous studies employing this method. The reason is that comparing a portfolio to market returns does not accurately reflect the portfolio's specific investment objectives and strategies, and it may not always provide a comprehensive view of the portfolio's performance due to differences in the structure and distribution of stocks.

5.2. Recommendations

In the future, author needs to research and explore more clustering algorithms such as Hierarchical Clustering or DBSCAN to compare their effectiveness and choose a more suitable algorithm for our investment purposes. Beside that, the author needs to expand the scope of research by adding other financial ratios such as P/S, P/CF, EBITDA/GP, etc. to increase the accuracy of clustering. The weight of stocks in the investment portfolio needs to be calculated and determined carefully to achieve maximum profitability and minimum risk. Morveover, the author needs to regularly update the investment portfolio based on the latest economic, political, and financial situations to minimize risks and optimize profits.

We must be aware that machine learning is only a decision support tool and cannot completely replace human investment decisions. Investors still need knowledge and practical experience to make accurate and effective investment decisions.

Appendix 1

```
import numpy as np
from numpy import sqrt
import pandas as pd
import datetime
from datetime import datetime
# Libraries to help with data visualization
import matplotlib.pyplot as plt
import seaborn as sns
sns.color_palette("Paired")
#library optimal
import quandl
import scipy.optimize as sco
# Removes the limit for the number of displayed columns
pd.set_option("display.max_columns", None)
# Sets the limit for the number of displayed rows
pd.set_option("display.max_rows", 200)
# to scale the data using z-score
from sklearn.preprocessing import StandardScaler
# to compute distances
from scipy.spatial.distance import pdist, cdist
# to perform k-means clustering, compute metric
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
#!pip install yellowbrick
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer
# to perform PCA
from sklearn.decomposition import PCA
#ignore warnings
import warnings
warnings.filterwarnings("ignore")
from google.colab import files
upload = files.upload()
from google.colab import files
```

```
upload = files.upload()
from google.colab import files
upload = files.upload()
from google.colab import files
upload = files.upload()
hose = pd.read_csv('VN Index Historical Data.csv')
hose = hose.iloc[:,:2]
hose.head()
#read data
roe = pd.read_excel('data_thanh (1).xlsx', sheet_name ='roe' )
ta = pd.read_excel('data_thanh (1).xlsx', sheet_name ='total assets')
debt = pd.read_excel('data_thanh (1).xlsx', sheet_name ='total debt')
pb = pd.read_excel('data_thanh (1).xlsx', sheet_name ='PB')
equity = pd.read_excel('data_thanh_3 (1).xlsx', sheet_name ='Equity')
ni = pd.read_excel('data_thanh (2).xlsx', sheet_name ='net income')
cap = pd.read_excel('data_thanh (2).xlsx', sheet_name ='Market Cap')
indus = pd.read_excel('data_thanh (2).xlsx', sheet_name ='code')
hnx = pd.read_excel('data_thanh (1).xlsx', sheet_name ='HNX index')
hnx = hnx.iloc[:,[0,-1]]
#clean data
def add(df):
  df = df.iloc[:,1:]
  df.columns = ['Stock','Code','amount']
  df=df.iloc[1:,:]
  df['Code'] = pd.to_datetime(df['Code'])
  df['Code'] = df['Code'].dt.strftime("%Y")
  df.dropna(axis=0, subset='Code', inplace = True)
  df['Code'] = df.Code.astype('int')
  df = df[df['Code'].isin([2021,2018,2019,2017,2020])]
  return df
#apply def
equity = add(equity)
ni = add(ni)
cap = add(cap)
```

```
#clean data
def aggr(df):
  df = df.rename(columns=df.iloc[0])
  df =  df.iloc[2:,:]
  df = pd.melt(df, id_vars='Code', value_vars = list(df.columns[1:]), var_name='Stock', value_name='amount')
  df['Code'] = df.Code.astype('int')
  df = df[df['Code'].isin([2021,2018,2019,2017,2020])]
  return df
# apply def
roe = aggr(roe)
ta = aggr(ta)
debt = aggr(debt)
pb = aggr(pb)
#@title
# clean indus dataframe
indus = indus.iloc[:,:4]
indus.rename({'Updated at 14:23:40':'Stock', 'Company Common Name': 'Name','GICS Industry Name':'Industry'},axis = 1,inplace=True)
indus.head()
# def split ticker
def stock(ticker):
   s = ticker.split('.')
   return s[0]
#split dataframe
def split_df(df):
  df.dropna(axis=0, subset=list(df.columns), inplace = True)
  df['Stock'] = df.Stock.apply(stock)
  return df
# #apply def
roe = split_df(roe)
ta = split_df(ta)
debt = split_df(debt)
pb = split_df(pb)
indus = split_df(indus)
```

```
equity = split_df(equity)
cap = split_df(cap)
ni = split_df(ni)
indus = indus.set_index('Stock')
# convert type
def tp(df):
  df['amount'] = df['amount'].astype('float')
  df['Code'] = df['Code'].astype('float')
  return df
#apply def
roe = tp(roe)
ta = tp(ta)
debt = tp(debt)
pb = tp(pb)
equity = tp(equity)
cap = tp(cap)
ni = tp(ni)
# merge data
result = pd.merge(roe, ta, on=["Code", "Stock"], how = 'outer', suffixes=('_roe', '_size'))
result = pd.merge(result, pb, on=["Code", "Stock"], how = 'outer', suffixes=('', '_pb'))
result = pd.merge(result,equity, on=["Code", "Stock"], how = 'outer', suffixes=('', '_equity'))
result = pd.merge(result,debt, on=["Code", "Stock"], how = 'outer', suffixes=('', '_debt'))
result = pd.merge(result, ni, on=["Code", "Stock"], how = 'outer', suffixes=('', '_ni'))
result = pd.merge(result, cap, on=["Code", "Stock"], how = 'outer', suffixes=('', '_cap'))
result.dropna(axis=0, subset=list(result.columns), inplace = True)
result.head()
# calculate some ratios
result['size'] = np.log(result['amount_size'])
result['roa'] = result['amount_ni']/(result['amount_size']*1000)
result['d/e'] = (result['amount_debt']*1000)/result['amount_equity']
result['p/e'] = result['amount_cap']/result['amount_ni']
result = result.rename({'amount':'p/b'},axis=1)
result.drop(['amount_size','amount_debt','amount_equity','amount_cap','amount_ni'],axis=1,inplace=True)
data = result.copy()
```

```
data.head(15)
# clean roe
def clean_roa(x):
  return float(x/100)
for col in  data.loc[:,data.columns.str.startswith('amount_roe')].columns:
   data[col] = data[col].apply(lambda x:clean_roa(x))
# # clean roa
# def clean_roe(x):
#    return float(x/100)
# for col in  data.loc[:,data.columns.str.startswith('roa')].columns:
#     data[col] = data[col].apply(lambda x:clean_roe(x))
# split data into each year
df_2021 = data[data['Code']==2021]
df_2017 = data[data['Code']==2017]
df_2018 = data[data['Code']==2018]
df_2019 = data[data['Code']==2019]
df_2020 = data[data['Code']==2020]
#remove duplicate
df_2021=df_2021.drop_duplicates(subset=['Stock'])
df_2020=df_2020.drop_duplicates(subset=['Stock'])
df_2019=df_2019.drop_duplicates(subset=['Stock'])
df_2018=df_2018.drop_duplicates(subset=['Stock'])
df_2017=df_2017.drop_duplicates(subset=['Stock'])
remove_out(data.drop(['Code'],axis=1).set_index('Stock'))
pdList = [subset_scaled_df_2021,
subset_scaled_df_2020,
subset_scaled_df_2019,
subset_scaled_df_2018,
subset_scaled_df_2017]  # List of your dataframes
new_df = pd.concat(pdList)
remove_out(new_df).describe().round(2)
# amount of companies of each year
print(df_2021.shape)
print(df_2020.shape)
```

```
print(df_2019.shape)
print(df_2018.shape)
print(df_2017.shape)
# print(df_2022.shape)
**II. PRE PROCESSING DATA**
# #remove oulier
def  remove_out(df):
  exclude_df=df.copy()
  for col in exclude_df.columns:
    mean_col = exclude_df[col].mean()
    std_col = exclude_df[col].std()
    left_bound = mean_col - 3*std_col
    right_bound = mean_col + 3*std_col
    exclude_df = exclude_df[(exclude_df[col]>=left_bound) & (exclude_df[col]<=right_bound)]
  return exclude_df
def remove(exclude_df):
  df_remove = remove_out(exclude_df.drop(['Code'],axis=1).set_index('Stock'))
  df_remove = df_remove.dropna(axis=0, subset=list(df_remove.columns))
  return df_remove

# remove outlier
df_re_2021 = remove(df_2021)
df_re_2020 = remove(df_2020)
df_re_2019 = remove(df_2019)
df_re_2018 = remove(df_2018)
df_re_2017 = remove(df_2017)
#normalize data
from sklearn.preprocessing import MinMaxScaler
def scale(df):
  minmax = MinMaxScaler()
  v_scaled = minmax.fit_transform(df)
  subset_scaled_df = pd.DataFrame(v_scaled, columns = df.columns)
  return subset_scaled_df
subset_scaled_df_2021 = scale(df_re_2021)
```

```
subset_scaled_df_2020 = scale(df_re_2020)
subset_scaled_df_2019 = scale(df_re_2019)
subset_scaled_df_2018 = scale(df_re_2018)
subset_scaled_df_2017 = scale(df_re_2017)
df_2020
**III. CLUSTER**
def dis(subset_scaled_df):
  clusters=range(1,10)
  meanDistortions=[]
  for k in clusters:
     model=KMeans(n_clusters=k, random_state=0)
     model.fit(subset_scaled_df)
     prediction=model.predict(subset_scaled_df)
     distortion=sum(np.min(cdist(subset_scaled_df, model.cluster_centers_, 'euclidean'), axis=1)) / subset_scaled_df.shape[0]
     meanDistortions.append(distortion)
     print('Number of Clusters:', k, '\tAverage Distortion:', distortion)
  sns.set_style("white")
  plt.plot(clusters, meanDistortions, 'bx-')
  plt.xlabel('k')
  plt.ylabel('Average Distortion')
  plt.title('Selecting k with the Elbow Method', fontsize=15)
  return 'Number of Clusters:', k, '\tAverage Distortion:', distortion; plt.show()
def sihou(subset_scaled_df):
  sil_score = []
  cluster_list = list(range(2,10))
  for n_clusters in cluster_list:
     clusterer = KMeans(n_clusters=n_clusters, random_state=0)
     preds = clusterer.fit_predict((subset_scaled_df))
     centers = clusterer.cluster_centers_
     score = silhouette_score(subset_scaled_df, preds)
     sil_score.append(score)
     print("For n_clusters = {}, silhouette score is {})".format(n_clusters, score))
  sns.set_style("white")
```

```
  plt.plot(cluster_list,sil_score)
  plt.xlabel('Number of cluster (k)')
  plt.ylabel('Average Silhou width')
  plt.title('Selecting k with the Silhouette', fontsize=15)
  return plt.show()
# Finding optimal no. of clusters with silhouette coefficients
def viz(df,n):
  visualizer = SilhouetteVisualizer(KMeans(n, random_state = 1))
  visualizer.fit(df)
  sns.set_style("white")
  sns.color_palette("tab10")
  return visualizer.show()
  #pca
def pca_plot(df,df_re,n):
    pca = PCA(n_components=2)
    X_reduced_pca = pca.fit_transform(df)
    reduced_df_pca = pd.DataFrame(data=X_reduced_pca, columns=["Component 1", "Component 2"])
    kmeans = KMeans(n_clusters=n, random_state=1)
    kmeans.fit(df)
    df['K_means_segments'] = kmeans.labels_
    df_re['K_means_segments'] = kmeans.labels_
    sns.set_style("white")
    sns.scatterplot(
        data=reduced_df_pca,
        x="Component 1",
        y="Component 2",
        hue=df["K_means_segments"],palette='tab10')
    plt.legend(bbox_to_anchor=(1, 1))
    return plt.show()
**YEAR 2017**
pca_plot(subset_scaled_df_2018,df_re_2018,3)
sihou(subset_scaled_df_2017)
viz(subset_scaled_df_2017,3)
pca_plot(subset_scaled_df_2017,df_re_2017,3)
```

```
**Year 2018**
dis(subset_scaled_df_2018)
sihou(subset_scaled_df_2018)
viz(subset_scaled_df_2018,3)
pca_plot(subset_scaled_df_2018, df_re_2018, 3)
**YEAR 2019**
dis(subset_scaled_df_2019)
sihou(subset_scaled_df_2019)
viz(subset_scaled_df_2019,3)
pca_plot(subset_scaled_df_2019, df_re_2019, 3)
dis(subset_scaled_df_2020)
sihou(subset_scaled_df_2020)
viz(subset_scaled_df_2020,4)
pca_plot(subset_scaled_df_2020, df_re_2020, 4)
**YEAR 2021**
dis(subset_scaled_df_2021)
sihou(subset_scaled_df_2021)
viz(subset_scaled_df_2021,4)
pca_plot(subset_scaled_df_2021, df_re_2021, 4)
**IV. EDA CLUSTER**
def eda(subset_scaled_df, df_re):
  cluster_profile = df_re.groupby('K_means_segments').mean().round(2)
  cluster_profile['count_in_each_segments'] =
df_re.reset_index().groupby('K_means_segments')['Stock'].count().values
  fig, axes = plt.subplots(2, 3,  figsize=(10, 8))
  fig.suptitle('Boxplot of numerical variables for each cluster', fontsize=20)
  counter = 0
  for ii in range(2):
    for jj in range(3):
      if counter <11:
        sns.boxplot(ax=axes[ii,
jj],y=subset_scaled_df[df_re.columns[counter]],x=subset_scaled_df['K_means_segments'])
        counter = counter+1
  return cluster_profile
eda(subset_scaled_df_2018,df_re_2018)
```

```
# let's see the names of the securities in each cluster
def cluser (df_re):
  for cl in df_re["K_means_segments"].unique():
    print(
        "The",
        df_re[df_re["K_means_segments"] == cl].index.nunique(),
        "Securities in cluster",
        cl,
        "are:",
    )
    print(list(df_re[df_re["K_means_segments"] == cl].index.unique()))
    print("-" * 100, "\n")
cluser(df_re_2017)
cluser(df_re_2018)
cluser(df_re_2019)
cluser(df_re_2020)
cluser(df_re_2021)
eda(subset_scaled_df_2017, df_re_2017)
eda(subset_scaled_df_2018, df_re_2018)
eda(subset_scaled_df_2019, df_re_2019)
eda(subset_scaled_df_2020, df_re_2020)
eda(subset_scaled_df_2021, df_re_2021)
**V. OPTIMAL PORFOLIO**
def add(df):
  df.columns = ['Date','HNX']
  df['Date'] = pd.to_datetime(df['Date'])
  # df['Date'] = df['Date'].dt.strftime("%Y-%M-%D")
  df.fillna(method='ffill')
  return df
def add_index(df):
  df.columns = ['Date','VNINDEX']
  df['Date'] = pd.to_datetime(df['Date'])
  # df['Date'] = df['Date'].dt.strftime("%Y-%M-%D")
  df.fillna(method='ffill')
```

```
  return df
def spli(x):
  x = x.replace(',',")
  return x
def hose_time(df):
  df = df.groupby(df.index.strftime('%Y-%m')).first()
  return df


hnx = add(hnx)
hose = add_index(hose)
hose['VNINDEX']=hose['VNINDEX'].apply(spli).astype('float')
hose.head()
price= pd.read_excel('data_thanh (1).xlsx', sheet_name ='close price' )
code = price.iloc[3]
b=[]
# clean price
def change_delist_code(s):
  b = s.split('.')
  return b[0]
code_labels = list(code.apply(change_delist_code))
price.columns = code_labels
price = price.iloc[5:,:]
# split price by year
price_2022 = price.loc[(price['Code'] > datetime.strptime('2021-12-01', '%Y-%m-%d'))].set_index('Code').sort_index()
price_2021 = price.loc[(price['Code'] > datetime.strptime('2020-11-30', '%Y-%m-%d')) &(price['Code'] < datetime.strptime('2022-01-01', '%Y-%m-%d'))].set_index('Code').sort_index()
price_2020 = price.loc[(price['Code'] > datetime.strptime('2019-11-30', '%Y-%m-%d')) &(price['Code'] < datetime.strptime('2021-01-01', '%Y-%m-%d'))].set_index('Code').sort_index()
price_2019 = price.loc[(price['Code'] > datetime.strptime('2018-11-30', '%Y-%m-%d')) &(price['Code'] < datetime.strptime('2020-01-01', '%Y-%m-%d'))].set_index('Code').sort_index()
price_2018 = price.loc[(price['Code'] > datetime.strptime('2017-11-30', '%Y-%m-%d')) &(price['Code'] < datetime.strptime('2019-01-01', '%Y-%m-%d'))].set_index('Code').sort_index()
price_2017 = price.loc[(price['Code'] > datetime.strptime('2016-11-30', '%Y-%m-%d')) &(price['Code'] < datetime.strptime('2018-01-01', '%Y-%m-%d'))].set_index('Code').sort_index()
# find these stocks in 2 df
```

```
def find(df,pr):
  dr=[]
  # df= df.drop('Year',axis=1).set_index('Stock')
  st = list(df.index.values)
  price_st=list(pr.columns)
  for i in range(len(price_st)):
    if price_st[i] not in st:
      dr.append(price_st[i])
  return dr
#drop stock not in 1st df
def delti(df,pr):
  pr=pr.drop(find(df,pr), axis=1)
  return pr
Year 2017
def hose_time(df):
  return df.groupby(df.set_index('Date').index.strftime('%Y-%m')).first().drop('Date',axis = 1)
#hnx exchange
def hnx_time(df):
  df = df.groupby(df.set_index('Date').index.strftime('%Y-%m')).last().drop('Date',axis=1)
  return df
#stock
def stock_time(df):
  df = df.groupby(df.index.strftime('%Y-%m')).last()
  return df
def ye_(df_re,price_2x,price_1x,subset_scaled_df,as_date, end_date, hose,hnx,n,indus):
  from sklearn.metrics import pairwise_distances_argmin_min
  year_x = delti(df_re,price_2x)
  port_x = delti(df_re,price_1x)
  km = KMeans(n_clusters=n).fit(subset_scaled_df)
  closest, _ = pairwise_distances_argmin_min(km.cluster_centers_, subset_scaled_df)
  col_year= list(closest)
  df_year= df_re.reset_index()
  col = list(df_year[df_year['Stock'].isin(list(df_year.iloc[col_year]['Stock']))]['Stock'])
  stocks =  year_x[col]
```

```
    return_ptf_index = stock_time(stocks).loc[as_date:end_date].join(hose_time(hose).loc[as_date:end_date]).join(hnx_time(hnx).loc[as_date:end_date]).pct_change().dropna()+1
#   return return_ptf_index
# def plot_por(return_ptf_index):
    return_ptf_index['port'] = return_ptf_index[col].sum(axis=1)/len(col)
    return_ptf_index['cluster_portfolio_return'] = return_ptf_index['port']
    return_ptf_index['VNINDEX_return'] = return_ptf_index['VNINDEX']
    return_ptf_index['HNX_return'] = return_ptf_index['HNX']
    plt.figure(figsize = (10,6))
    ax = plt.gca()
    plt.title("Portfolio Performance")
    return_ptf_index['cluster_portfolio_return'].cumprod().plot(ax=ax,color=sns.color_palette()[1],linewidth=2)
    return_ptf_index['VNINDEX_return'].cumprod().plot(ax=ax,color=sns.color_palette()[2],linewidth=2)
return_ptf_index['HNX_return'].cumprod().plot(ax=ax,color=sns.color_palette()[3],linewidth=2)
    plt.xlabel("Date")
    plt.ylabel("Cumulative Return %")
    plt.legend()
    indus.loc[col,]
    return col
ye_(df_re_2017,price_2018,price_2017,subset_scaled_df_2017,'2017-12','2018-12',hose,hnx,3,indus)
ye_(df_re_2018,price_2019,price_2018,subset_scaled_df_2018,'2018-12','2019-12',hose,hnx,3, indus)
ye_(df_re_2019,price_2020,price_2019,subset_scaled_df_2019,'2019-12','2020-12',hose,hnx,3, indus)
ye_(df_re_2020,price_2021,price_2020,subset_scaled_df_2020,'2020-12','2021-12',hose,hnx,4,indus)
ye_(df_re_2021,price_2022,price_2021,subset_scaled_df_2021,'2021-12','2022-12',hose,hnx,4,indus)
```

Appendix 2

Source Data: [data - Google Drive]

REFERENCE

Vietnamese

1. Lực, V. T. (2011), "Ứng dụng lý thuyết đầu tư tài chính hiện đại trong quản lý danh mục đầu tư trên TTCK Việt nam", Ha Noi.
2. Hiển, N.Đ (2012), "Hành vi của nhà đầu tư trên thị trường chứng khoán Việt Nam"

English

3. Baser, P., & Saini, J. R. (2015), "Agent based stock clustering for efficient portfolio management*", International Journal of Computer Applications*, 116(3), 36-41.
4. Statman M. (2004), "The diversification puzzle", *Financial Analysts Journal*, 60(4), 44-53.
5. Krishna K., Murty M. N. (1999), "Genetic K-means algorithm", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics*), 29(3), 433-439.
6. Rokach L., Maimon O. (2005). *Data Mining and Knowledge Discovery Handbook*, 321–352.
7. Hoss Belyadi, Alireza Haghighat (2021), A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications. *Machine Learning Guide for Oil and Gas Using Python*, 125-168.
8. Momeni M., Mohseni M., & Soofi M. (2015), Clustering stock market companies via k-means algorithm. *Kuwait Chapter of Arabian Journal of Business and Management Review*, 33(2578), 1-10.
9. Iwan Fadilah, Rini Setyo Witiastuti (2018), A Clustering Method for portfolio optimization, Indonesia.
10. Preeti Baser và Jatinderkumar R. Saini (2015), "Agent based Stock Clustering for Efficient Portfolio Management", *International Journal of Computer Applications* (0975 – 8887). India
11. Bilgehan Tekin, Fatih Burak Gümusss (2017), "The Classification of Stocks with Basic Financial Indicators: An Application of Cluster Analysis on the BIST 100 Index" *International Journal of Academic Research in Business and Social Sciences 2017, Vol. 7,* Turkey.
12. Yusuf, R., Handari, B. D., & Hertono, G. F. (2019). "Implementation of agglomerative clustering and genetic algorithm on stock portfolio optimization with possibilistic constraints", *Proceedings of the 4th international symposium on current progress in mathematics and sciences (ISCPMS2018)*, Indonesia
13. Bin Shu, (2020). K-Means Stock Clustering Analysis Based on Historical Price Movements and Financial Ratios. CMC Senior Theses. 2435. America.

14. Nanda, S. R., Mahanty, B., & Tiwari, M. K. (2010). Clustering Indian stock market data for portfolio management. Expert Systems with Applications, 37(12), India
15. Siregar, B., & Pangruruk, F. A. (2021). Portfolio Optimization Based on Clustering of Indonesia Stock Exchange: A Case Study of Index LQ45. Indonesian Journal of Business Analytics, 1(1), 59-70.
16. Kedia, V., Khalid, Z., Goswami, S., Sharma, N., & Suryawanshi, K. (2018). Portfolio Generation for Indian Stock Markets Using Unsupervised Machine Learning. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*
17. Long N, C Wisitpongphan N, Meesad, P & Unger H (2014), 'Clustering stock data for multi-objective'. *International Journal of Computational Intelligence and Applications*, 13(02), 1450011.
18. Nainggolan, R., Perangin-angin, R., Simarmata, E., & Tarigan, A. F. (2019). "Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method", *Journal of Physics: Conference Series*, 1361, 012015.