University Of Manchester

School of Computer Science

Pattern Classification of Stock Price Movements

Tong Li

Supervisor: Dr. Xiao-Jun Zeng

A final year report submitted for the degree of

BSc Computer Science

July 14, 2019

Abstract

In the stock market, the share price movement of each individual company forms a time series data with different patterns. Extracting an enormous amount of valuable information from those patterns can help investors understand fluctuations in share price values and select stocks accordingly. In this project, typical stock prices patterns are identified by applying clustering algorithms to FTSE100, which contains 101 companies listed on the London Stock Exchange. Besides testing with conventional clustering algorithms (K-Means, Expectation Maximisation, Hierarchical Clustering), a new algorithm (Hierarchical Based K-Means), which utilises the advantages of K-Means, Hierarchical Clustering and Greedy algorithm, is designed to achieve better clustering results. The performance of each algorithm is evaluated with internal clustering indices (Silhouette Coefficient, Davies-Bouldin Index, Calinski-Harabaz Index) as well as via data visualisation. As well as exploring static patterns for one year, dynamic patterns for three years are also considered because a company's strategic plan is normally three to five years. Next, with the aim of making good use of stock patterns, more analysis, based on the results of dynamic patterns, is carried out. One advanced investment strategy and several basic investment strategies (Daily Investment, Overnight Investment,Intraday Investment) are evaluated by Sharpe Ratio, which examines the performance of an investment by considering its return against its risk. In addition, Monte Carlo Simulation is used for portfolio optimisation, which aims to find the target portfolio of assets (a set of stocks) with the highest Sharpe Ratio for a given range of returns. The final part of the project tries to explore and prove the potential correlation between Twitter's sentiment and stock patterns.

Acknowledgement

Firstly, I would like to express my most sincere appreciation to my supervisor Dr Xiao-Jun Zeng for all his patience and support throughout the whole project.

Secondly, I would like to thank my second marker Dr Carole Twining for providing me with valuable feedback and suggestions.

Finally, I would like to extend my gratitude to all those, especially my family, who cared for me and encouraged me all the time.

Contents

List of Figures

1 Introduction

In this chapter, an overview of the whole project is provided. Firstly, the motivation and objectives of this project are explained. Then, the methodology and design of the project are described. Finally, a brief introduction to each chapter is given to help the reader understand the structure of this report.

1.1 Motivation

Stocks represent fractional ownership in a company, give companies the ability to access capital from the public and in turn, in the stock market, investors can invest their money in companies with promising products[1]. The stock market is considered critical to economic development and regarded as a reflection of the current economic environment.

William believes that no matter who you are, investing in common stocks is a skill that everyone should learn[2]. However, the dynamic economic environment adds more uncertainties to the stock market, increasing the chance of failure. Once investors have identified an uptrend, they are more likely to make a profit.

Investors may have the following interests. They may want to:

- Find ways to understand fluctuations in share price values and select stocks accordingly;
- Know which investment strategies / portfolio (set of stocks) can make more profit but also carry lower risk;
- Know which common factors influence or reflect the trend of target stocks.

In the stock market, the share price movement of each individual company forms a time series data with different patterns. Extracting an enormous amount of valuable information from those patterns can help investors understand the stock market better and address the above concerns.

1.2 Aims and Objectives

The objectives of this project are explained as follows:

- To develop machine learning algorithms that identify the different patterns of share price movements.
- To evaluate the clustering results using multiple approaches.
- To develop portfolio selection approaches based on the Modern Portfolio Theory.
- To explore reasonable investment strategies that help investors make rational decisions.
- To evaluate related factors which may influence or reflect the movements of the stock market.

1.3 Report Structure

The report is organised as follows:

- **Chapter 1 - Introduction**: In this chapter, a brief overview of the whole project is given. It contains the motivation and objectives of this project and the structure of this report.
- **Chapter 2 - Background**: In this chapter, the background behind this project is provided. It mentions the availability of the stock sets, the basic machine learning techniques and natural language processing techniques used in this project.
- **Chapter 3 - Methodology and Design**: In this chapter, the methodology and design of this project are introduced, and the tools and packages used in this project are also mentioned.
- **Chapter 4 - Implementation and Experiments**: This chapter discusses how the basic techniques of data collection, data pre-processing and three conventional machine learning algorithms are implemented.
- **Chapter 5 - Selection and Evaluation**: Three evaluation indices are used in this chapter to compare the clustering results numerically and to choose a suitable number of clusters for further analysis.
- **Chapter 6 - New Algorithm Design (Hierarchical Based K-Means)**: The focus is on the new algorithm that was designed to obtain better clustering results.
- **Chapter 7 - Dynamic Pattern Analysis**: This chapter focuses on the dynamic pattern for each stock for three years, the aim of which is to find stable stocks for further analysis.
- **Chapter 8 - Applications on Stock Patterns**: Chapter 8 firstly compares different investment strategies and develops portfolio selection approaches based on the Modern Portfolio Theory. Then, natural language processing techniques are employed to analyse the sentiment of each corresponding stock and explore the potential correlation between stock patterns and their sentiments on social media
- **Chapter 9 - Reflection and Conclusion**: This chapter reflects on the planning and management of the project and outlines its main achievements.

2 Background

This chapter mainly focuses on the background investigation, aiming to introduce the time series data and investigate the related techniques and methods that can be used in this project.

2.1 Time Series Data

Classified as dynamic data, time series data contains feature value changes as a function of time, is naturally high dimensional and normally has a large data size. It is a highly popular field as clustering such complex data allows us to discover interesting patterns and extract valuable information in time-series data sets. The amount of time-series data in different domains such as Finance, Energy and Biology, can be stored and kept for a long time by real-world applications. Several analyses can be done with time series data in various areas for different purposes such as subsequence matching, pattern detection, and a considerable number of studies have focused on this area with updated techniques in the last decade[3].

2.1.1 FTSE 100

The FTSE100(Financial Times Stock Exchange 100 index) contains the top 101 companies with the largest market value that are listed on the London Stock Exchange. It serves as a leading indicator of prosperity for companies in the United Kingdom, and also reflects the economic environment as it is impacted by UK's daily development[4]. The stock price data is stored in the form of time-series data, and thus provides the opportunity for valuable information to be extracted and analysed.

2.2 Machine Learning

As a vast amount of valuable information can be extracted from the stock market, researchers are attracted by the hidden values within the stock market and try to use machine learning techniques to explore this problem domain.

Some researchers have focused on stock market investment issues within the Taiwan stock market using the two-stage data mining approach.In the first stage, they utilised the Apriori algorithm to propose stock category association based on association rules, while in the second stage, the K-Means algorithm was used for mining stock category clusters for investment information. By implementing this approach, they could find valuable information such as the primary factors that affect TAIEX (Taiwan Capitalization Weighted Stock Index)[5].

Due to the lack of practical approaches for high dimensional data and shifting stock prices, other researchers at the University of Malaya proposed a three-phase (pre-clustering, purifying and summarisation, merging) clustering model to categorise companies in the stock market. The first phase approximated the results within different categories based on low-resolution data, which was generated by dimension reduction. Then the pre-clustered companies were split into sub-clusters. The final step was to merge sub-clusters until the target number of clusters was reached. This novel approach performed better than conventional approaches[6]. The idea behind this approach inspired the author of this project to design a new algorithm that is discussed and evaluated in subsequent chapters.

Finally, some researchers from India investigated India stock market data and selected stocks for building portfolios. Some machine learning techniques and models such as the Markowitz model, K-Means, Fuzzy C-Means, Self organising maps (SOM) were used to group stocks into several categories, and various evaluation indices (Silhouette Coefficient, Davies-Douldin index, Calinski-Harabasz index) were used to compare the performance of each clustering algorithm.[7].

2.3 Natural Language Processing

As one of the fastest growing research areas in natural language processing, sentiment analysis provides a series of methods and techniques that detect and analyse subjective information. In recent years, studies have focused more and more on sentiment analysis, the use of which has reached other research domains such as the prediction of financial markets[8].

Some researchers explored stock market predictions based on large scale collections of tweets and the results of sentiment analysis. They established a model by utilising OpinionFinder and GPOMS to analyse the public's mood, and by correlating the results of this analysis to the Dow Jones Industrial Average (DJIA). This model was then used to predict changes. The results showed an accuracy of 87.6% in predicting the up and down change for DJIA which indicated a potential correlation between sentiment analysis and stock market change[9]. The following figure outlines the three-phase methodology they used to predict the value of DJIA. Phase one involved the collection and analysis of public mood time series, and Granger causality analysis to determine correlation between stock prices and public mood. The final step was to predict the stock prices by implementing a Self-Organising Fuzzy Neural Network.

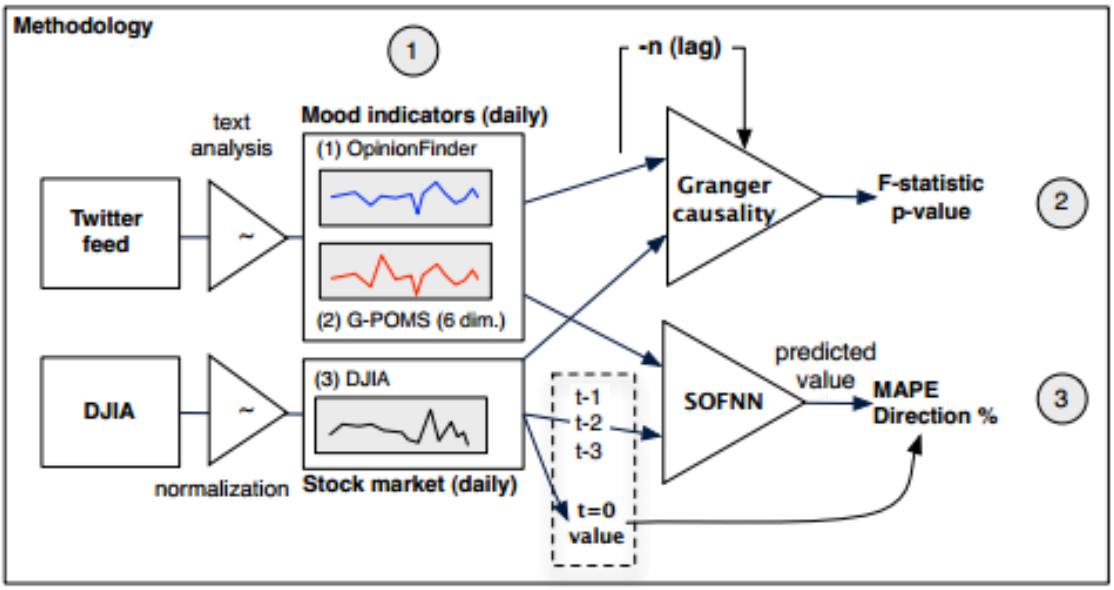


Figure 1: Three-phase methodology for stock prediction

2.4 Applications on Stock Patterns

Based on the clustering results, several applications were investigated by researchers in order to make good use of the stock patterns.

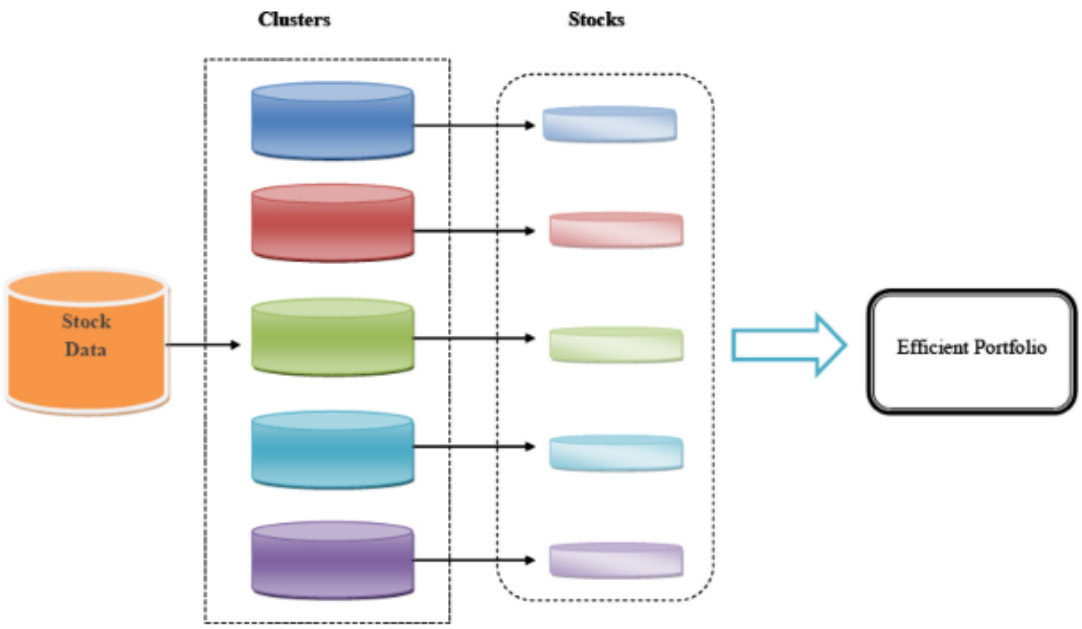


Figure 2: Creation of efficient portfolios

The above figure shows the creation process of efficient portfolios; India stock market data was clustered first and stocks from those patterns were selected. The selected data was used for building portfolios by using the Markowitz model, where the portfolio return was calculated by the weighted returns of stocks. The aim was to find a portfolio with minimum portfolio risk when a set of stocks and a target return range were given[7].

As the above theory defined a multivariate problem, some statistical methods such as Monte Carlo Analysis were widely used in a financial domain. It allowed researchers to run multiple trials and simulate the potential outcomes of an investment. Based on these simulation results, along with optimisation methods, the optimal values could be found.

3 Methodology and Design

3.1 Design of this Project

This section mainly focuses on the methodology and design of this project.

Figure 3: Methodology and Design

There are five stages in this project:

- **Objectives Analysis**: The first stage of this project is to give an analysis of this topic; it contains a literature review and an insight into the basic techniques used in this project.
- **Data Collection & Pre-processing**: suitable data are collected and the missing data problem is addressed. Then it is necessary to normalise the data and prepare them for further analysis.
- **Clustering Modelling**: stocks are grouped into different numbers of clusters by using conventional machine learning algorithms and design a novel algorithm to achieve better results.
- **Evaluation**: the clustering results are evaluated by using internal indices as well as via visualisation. The evaluation results are used in experiments for designing new algorithms.
- **Pattern-based Analysis**: In this stage, possible applications of stock patterns that aim to help investors in decision-making are investigated. In previous experiments, a set of promising stocks was selected from dynamic pattern analysis; thus, all experiments in this stage use these promising stocks as examples. At the same time, it is necessary to review the results and recheck if they meet the requirements towards achieving the project's objectives.
  - **Investment Strategy Analysis**: possible investment strategies are investigated and positive strategies are applied to the most promising stocks selected from dynamic patterns.

  - **Portfolio Selection and Optimisation**: from the dynamic pattern analysis, a set of promising stocks is chosen as an example. In this section, a framework to help investors to find the optimal portfolios is provided.
  - **Sentiment Analysis**: the potential correlations between the static stock patterns and their corresponding sentiments on Twitter are explored.

3.2 Methodologies of this Project

This section shows the step by step process in this project and describes the methodologies used in each step, as follows:

- Collect stock prices in FTSE100.
- Normalise stock prices by using three different ways and deal with missing data.
- Implement three conventional clustering methods (K-Means, Expectation Maximisation, Hierarchical Clustering).
- Evaluate and compare the performance of each clustering methods with three internal indices (Silhouette Coefficient, Davies-Bouldin Index, Calinski-Harabaz Index) as well as via visualisation. Then decide the number of clusters based on those evaluation methods.
- Identify the weakness of each method and design a new algorithm.
- Evaluate the performance of the new algorithm and compare it with conventional methods.
- Analyse dynamic patterns for three years and select promising stocks.
- Apply different investment strategies to the most promising stocks and evaluate different strategies with Sharp Ratio. An advanced strategy based on the Support Vector Machine is described.
- Utilise the Monte Carlo Simulation based on the Modern Portfolio Theory to find the optimal portfolio.
- Gather corresponding tweets of each stock and analyse their sentiments. Bind stock patterns and their sentiments together to explore the potential correlations.

3.3 Implementation Tools

Several tools are used in this project.

- **Weka**: A complete interactive tool[10] which was used during the first stage to help the author to understand the objectives of this project and to provide simple results.
  - Advantages: A complete interactive tool; user friendly and easy to learn. Not necessary to write any codes.
  - Disadvantages: Does not support data visualisation; difficult to evaluate results.

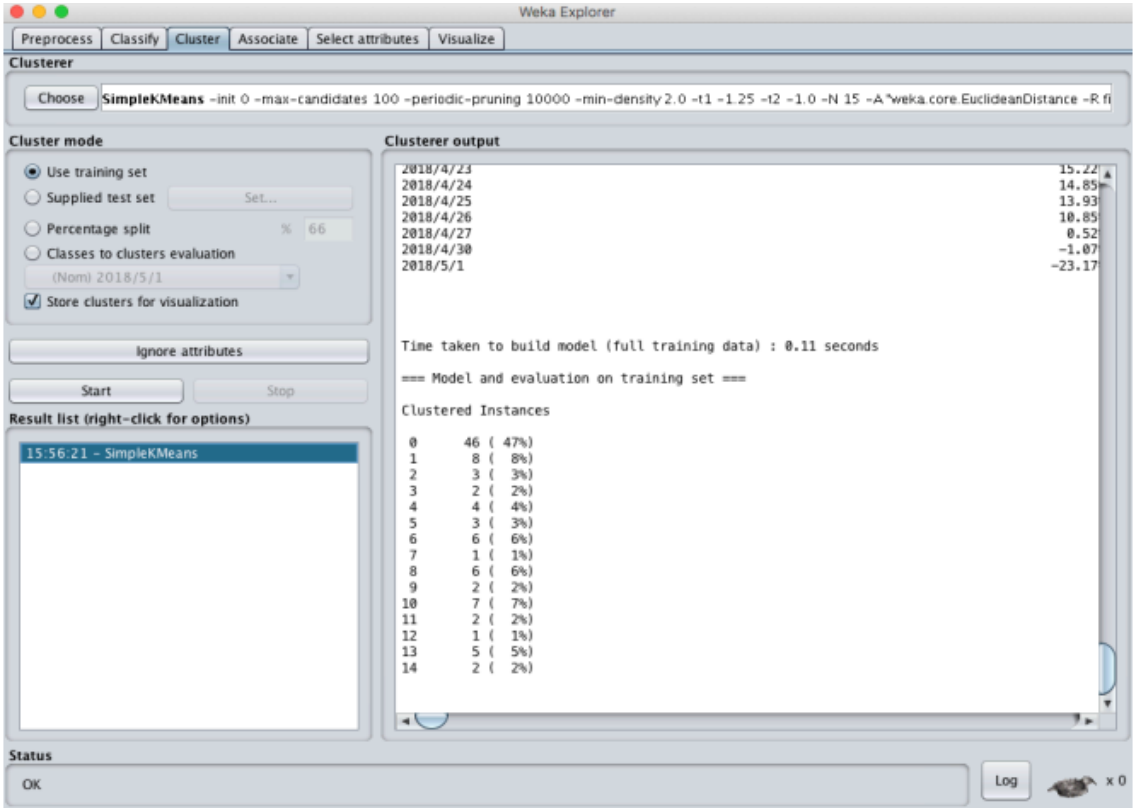


Figure 4: Sample K-Means clustering output from Weka

- **Python**: It is mainly used throughout the project. Several packages such as Pandas, Matplotlib, Scikit-learn, NLTK were used in this project.
- **Excel**: It is used for plotting charts and figures.

4 Implementation and Experiments

In this Chapter, the basic techniques for data collection and data preprocessing are described. Then, three classic clustering algorithms are described and the implementation of data visualisation is showed as well.

4.1 Data Collection

Providing extremely competent and comprehensive financial services, Yahoo Finance stands out among most other financial sites[11]. It has approximately 70 million unique visitors each month and its API can be used to collect historical stock prices. In this project, only the close price for each stock is given. To successfully obtain raw data from Yahoo Finance, it is necessary to provide a list of symbols which represent the stocks in FTSE100.

Symbol	Company name
VOD.L	Vodafone Group Plc
BA.L	BAE Systems plc
BATS.L	British American Tobacco p.l.c.
CCH.L	Coca-Cola HBC AG
SDR.L	Schroders plc
SMT.L	Scottish Mortgage Investment Trust plc
PRU.L	Prudential plc

Figure 5: List of Symbols

Based on the stock symbols in the list, the corresponding close prices of each stock are collected from Yahoo Finance. Stored by Pandas Dataframe, the stock prices must have a suitable format for both Python and Weka. Each row represents the stock prices for each company, and each column represents the stock prices for different companies on a specific date.

	A	B	C	D	E	F	G
1	index	2017/1/3 0:00	2017/1/4 0:00	2017/1/5 0:00	2017/1/6 0:00	2017/1/9 0:00	2017/1/10 0:00
2	III.L	-15.62485703	-15.66194075	-15.69639077	-15.63401444	-15.52725874	-15.54742774
3	ABF.L	-7.038027863	-10.50138968	-7.929597834	-8.958309454	-8.512524469	-8.649700232
4	ADM.L	-6.000619898	-6.61332208	-6.664383372	-6.562267453	-6.664383372	-9.115212094
5	AAL.L	-9.75501075	-11.26944545	-11.07528504	-11.85191636	-10.33748167	-3.891422056
6	ANTO.L	-22.43294222	-22.26432452	-21.36499125	-21.98327559	-20.85911628	-18.72321432
7	AHT.L	-7.356790574	-7.646835283	-7.124743113	-7.646835283	-8.168942071	-7.124743113
8	AZN.L	-11.60239536	-10.85561999	-9.391955765	-9.381990057	-8.246907629	-7.201431665
9	AV.L	-6.527426257	-6.603222341	-7.152726285	-7.266423623	-7.815927567	-7.759075686
10	BA.L	-4.934644143	-5.899774445	-5.497642099	-2.682673439	-3.889081037	-1.878398187

Figure 6: Sample Data

4.2 Data Preprocessing Techniques

4.2.1 Data Cleaning

Missing and erroneous data can pose significant problems with regard to the reliability and validity of study outcomes[12]; thus, the data cleaning process is used to avoid the above two problems and prepare integrated data for further analysis. Due to the competent services that Yahoo Finance provides, the collected data contains few outliers and noises. However, as some companies may have closure periods, it is still necessary to deal with the missing data.

The missing data is detected and replaced with the previous valid data as there is not too much loss in the provided data set. After the process of data cleaning, the quality of the provided data is improved. The following figures show the results before and after data cleaning.

HD	HE	HF
2017/10/31 0:00	2017/11/1 0:00	2017/11/2 0:00
1.450925433		-0.966215558
14.69478329	12.285223	13.73096258
0.983193207	-2.53337753	-2.95326518
14.49007305	18.40047629	20.61771247

(a) Before cleaning

HD	HE	HF
2017/10/31 0:00	2017/11/1 0:00	2017/11/2 0:00
1.450925433	1.450925433	-0.966215558
14.69478329	12.285223	13.73096258
0.983193207	-2.53337753	-2.95326518
14.49007305	18.40047629	20.61771247

(b) After cleaning

Figure 7: Data Cleaning

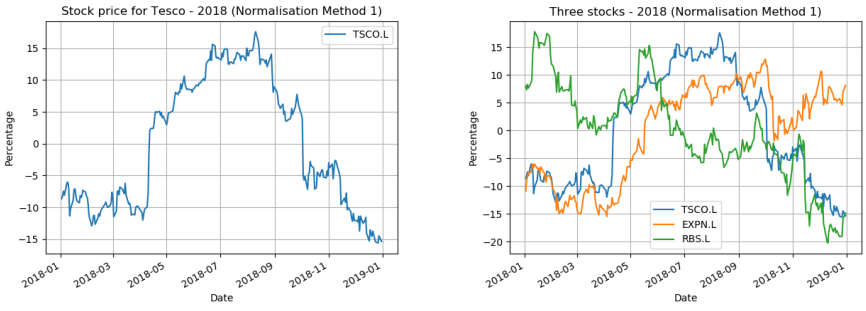
4.2.2 Data Normalisation

The major objective of this project is to identify the different patterns of share price movements. However, different companies may have the same patterns even though their prices are quite different. Normalising each company's close prices allows us to identify patterns based on their daily change rather than their daily prices. At the same time, it helps us to consider the relative change rather than the absolute change. In this project, three normalisation methods were tested, one dynamic method and two static methods.

The **daily stock price** can be considered as $X_i = (X_0, X_1, ...X_{n-1}, X_n)$ and the **mean price** in the date range $\overline{X}$.

- **Static Normalisation - Method 1**: It shows the relative change based on the mean price in the date range.

$$X_i = \frac{X_i - \overline{X}}{\overline{X}}$$



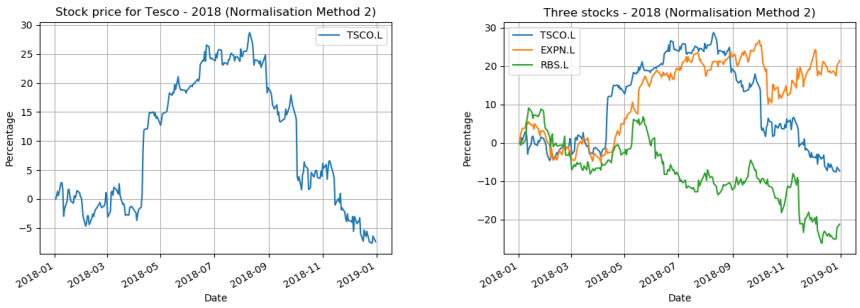
(a) Tesco's Stock Prices

(b) Three Stock Prices

Figure 8: Static normalisation method 1 for one stock and three stocks respectively

- **Static Normalisation - Method 2**: It shows the relative change based on the first day's price in the date range.

$$X_i = \frac{X_i - X_0}{X_0}$$

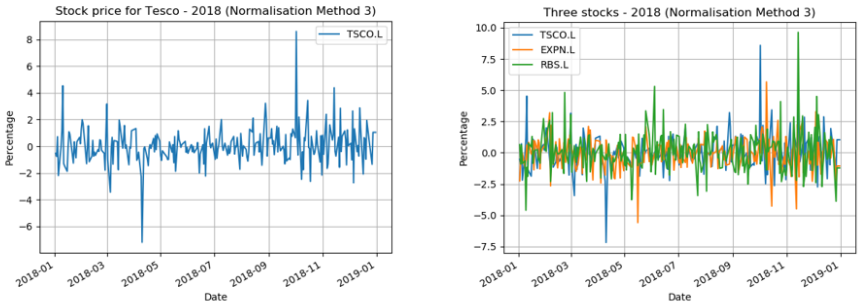


(a) Tesco's Stock Prices (b) Three Stock Prices

Figure 9: Static normalisation method2 for one stock and three stocks respectively

- **Dynamic Normalisation**: It shows the relative change over a period of two consecutive days. It is the relative change that is normally considered in each day's trading, but it is difficult to find a pattern.

$$X_i = \frac{X_i - X_{i-1}}{X_{i-1}}$$



(a) Tesco's Stock Prices (b) Three Stock Prices

Figure 10: Dynamic normalisation for one stock and three stocks respectively

It is clear that the Dynamic Normalisation is not suitable for identifying patterns for stock price movements. In terms of the two static methods, Method 1 uses the mean price in the date range to compare each day's price, while for Method 2, the first day's price is used as a baseline. If only one stock's movement is plotted, the shapes of those two movements are the same, but the starting percentages are different. However, if more than one stock is plotted, (for example, three stocks in one figure) it is found that, when Method 2 is used, the impact of the first few days is reduced, whereas when Method 1 is used, there are more distinguishable movements and clearer patterns.

4.3 Clustering Methods

Firstly, three conventional clustering algorithms (K-Mean, Expectation Maximisation, Hierarchical Clustering) are implemented in this project. The normalised data is in the form of $N \times M$ matrix where N is the number of stocks, and M represents the attributes of each stock. For time-series data, the attributes are different dates in a pre-defined date range.

4.3.1 K-Means

K-Means clustering is used to find groups for an unlabelled data set; it works iteratively to assign each data point to one of the K groups where K needs to be decided in advance[13]. There are two steps in K-Means clustering:

- **Data Assignment**: Assign each data point x to its nearest centroid c_i based on a specific distance metric dist(), the assignment process is based on the following equation:

$$\underset{c_i \in C}{argmin}\, dist(c_i, x)^2$$

  Where C is a collection of centroids.

- **Centroid Update**: Recompute the centroids by taking the mean value of all data point within the same cluster, the Update process is based on the following equation:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

  where S_i is the new computed cluster.

These two steps are repeated until a satisfactory result is obtained.

The following figure[14] shows 3 iterations of K-Means clustering for 2-Dimensional space. The coloured points represent the changes in each centroid.

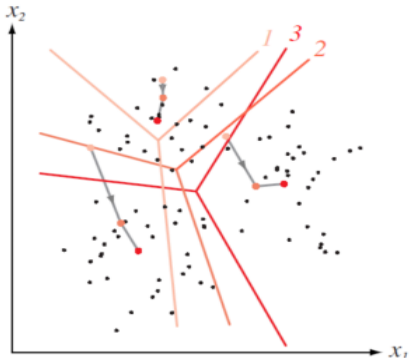


Figure 11: 2-D Simulation for K-Means

The *K-Means* library in *sklearn* is used and the number of clusters must be specified ahead of time. The K-Means algorithm is applied to the data set with different numbers of clusters and the results can be visualised through the codes in section 4.4.1 and section 4.4.2.

```
cluster_model = KMeans(n_clusters=num).fit(data_set)
```

Figure 12: Utility of K-Means library in sklearn

4.3.2 Expectation Maximisation

Based on the Gaussian Mixture Model, the Expectation Maximisation(EM) algorithm is a probabilistic model used to find the maximum a posteriori probability (MAP) and the estimates of parameters[15]. There are two steps in the EM algorithm:

- **Expectation step**: The expected value of the log-likelihood function θ is computed, with respect to the conditional distribution of Z, given the normalised stock prices X_i and the current estimates of the parameters $\theta^{(t)}$

$$Q(\theta|\theta^{(t)}) = E_{Z|X_i,\theta^{(t)}}[logL(\theta; X_i, Z)]$$

- **Maximisation step**: the parameter is updated to maximise Q

$$\theta^{(t+1)} = \underset{\theta}{argmax} Q(\theta|\theta^{(t)})$$

These two steps are to be repeated until θ convergences.

The following figure shows 6 iterations of an EM algorithm for 2-Dimensional space. The unlabelled points are grouped into two clusters.

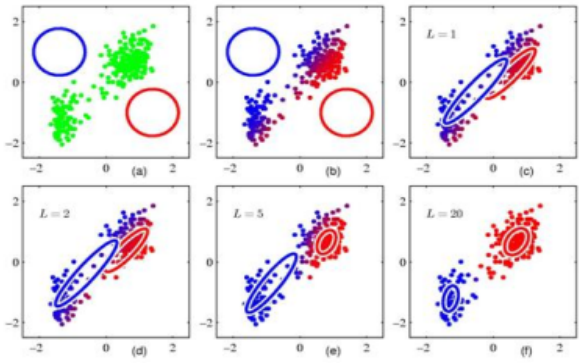


Figure 13: 2-D Simulation for EM

The EM algorithm is implemented by using Weka and the results can be visualised through section 4.4.1 and section 4.4.2.

4.3.3 Hierarchical Clustering

Based on similarities between groups, Hierarchical Clustering is designed to find the hierarchy of the observed data[16]. There are two approaches for Hierarchical Clustering:

- Agglomerative (a bottom-up strategy) : Initially each data object is a cluster; they are then merged into larger clusters based on the distance matrices.
- Divisive (a top-down strategy) : Initially all data objects are in one cluster; they are then split into smaller clusters based on the distance matrices.

The distance matrices determine the similarities between each cluster and the hierarchy of the observed data.

- **Single link** : the smallest distance between two clusters is considered.

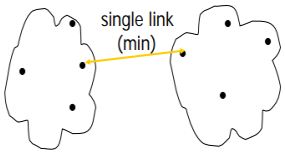


Figure 14: Single Link

- **Complete link**: the greatest distance between two clusters is considered.

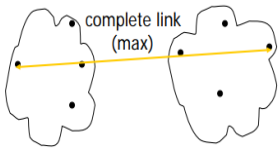


Figure 15: Complete Link

- **Average link**: the average distance between two clusters is considered.

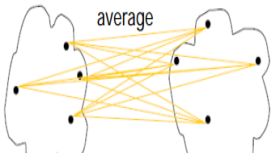


Figure 16: Average Link

- **Ward link**: the total within-cluster variance is considered. This is calculated by the weighted sum squared distance between every two cluster centroids and the merged cluster. The distance $d_W(s,t)$ between cluster s and cluster t is calculated by:

$$d_W(s,t) = \frac{n_s n_t}{n_s + n_t} d^2(\overline{x}_s \overline{x}_t)$$

where n_s and n_t are the size of two clusters, the $\overline{x}_s$ and $\overline{x}_t$ are the centres of the two clusters.

The following figure[17] shows the process of the Agglomerative approach.

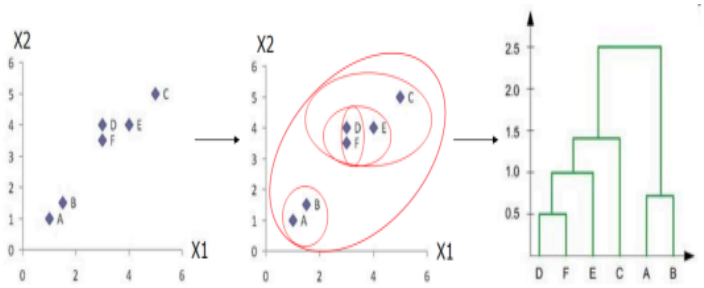


Figure 17: 2-D Simulation for Hierarchical Clustering

The *AgglomerativeClustering* library in *sklearn* is used. There are three parameters that need to be specified : distance metrics, linkage method and number of clusters. The Hierarchical Clustering method is applied to the data set with different numbers of clusters and different linkage methods, and then the results are visualised through section 4.4.1 and section 4.4.2.

```
cluster_model = AgglomerativeClustering(affinity='euclidean',linkage='ward',n_clusters=i).fit(data_set)
```

Figure 18: Utility of AgglomerativeClustering library in sklearn

4.4 Data Visualisation

The Data Visualisation is regarded as the most straightforward way to show clustering results and evaluate the performance of each clustering algorithm.

4.4.1 Pattern Visualisation

The following codes show the implementation of Pattern Visualisation, which is used to visualise the identified patterns of a given data set.

```
def pattern_visualization(data_set, cluster_result, cluster_num):
    gp_col = 'Cluster'
    header_list = list(data_set)
    temp_set = data_set.copy()
    temp_set[gp_col] = cluster_result

    df_mean = temp_set.groupby(gp_col)[header_list].mean()
    df_mean = df_mean.transpose()

    fig_title = 'Hierarchical_Based_KMeans algorithm - ' + str(cluster_num) + ' clusters'
    file_name = 'Hierarchical_Based_KMeans algorithm - ' + str(cluster_num) + ' clusters'
    fig = df_mean.plot(grid = True, title = fig_title)
    fig.set_xlabel('Date')
    fig.set_ylabel('Percentage')
    #fig.set_xticklabels(df_mean.index.tolist())
    fig = fig.get_figure()
    file_path = '../Graph/'
    fig.savefig(file_path + file_name)
    plt.show()
```

Figure 19: Code for Pattern Visualisation

4.4.2 Cluster Visualisation

The following codes show the implementation of Cluster Visualisation, which is used to visualise the movements of all the stocks in the same cluster.

```
def single_cluster_visualization(data_set, cluster_result, cluster_num):
    temp_set = data_set.copy()
    temp_set = temp_set.transpose()
    header_list = list(temp_set)
    temp_df = pd.DataFrame()
    for i in range(len(header_list)):
        if cluster_result[i] == cluster_num:
            temp_df[header_list[i]] = temp_set[header_list[i]]

    fig_title = 'K-means Clustering - Cluster' + str(cluster_num)
    file_name = 'K-means Clustering - Cluster' + str(cluster_num)
    fig = temp_df.plot(grid = True, title = fig_title)
    fig.set_xlabel('Date')
    fig.set_ylabel('Percentage')
    fig = fig.get_figure()
    file_path = '../Graph/'
    # fig.savefig(file_path + file_name)
    plt.show()
```

Figure 20: Code for Cluster Visualisation

5 Selection and Evaluation

In this chapter, some appropriate indices that evaluate the quality of the clustering results are given. The indices allow us to compare the quality of the clustering results numerically. Based on the evaluation results, it is first necessary to decide which number of clusters performs better and then to choose the number of clusters for further analysis. Finally, the results are then presented and this straightforward method is used to evaluate the performance of each algorithm.

5.1 Evaluation indices

5.1.1 Silhouette Coefficient

Previous studies mostly define 'Cluster Cohesion' as an indicator to show how closely related two samples are in one cluster, and they use 'Cluster Separation' to measure how distinct one cluster is from another clusters[18].

Combining the idea of cohesion and separation together, Silhouette Coefficient provides a numerical method that shows the quality of clustering results. The equation is defined as follows:

$$s = \frac{b - a}{max(a, b)}$$

where **a** is the mean distance between an object and all other objects in the same cluster, **b** is the mean distance between an object and all other objects in the next nearest cluster.

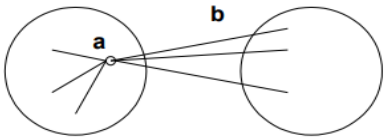


Figure 21: Example for Silhouette Coefficient

Typically, the score is bounded between -1 and 1, where -1 means incorrect clustering and 1 means highly dense clustering; thus, a **higher score** indicates a better clustering result[19].

5.1.2 Davies-Bouldin Index

The Davies-Bouldin Index[20] evaluates the performance of clustering results by considering the ratio of separation of clusters and intra-cluster distance. Assuming that n clusters is defined as C_i for i = 1,2....n, and among all the clusters in the range 1-n, it is necessary to find the most similar cluster C_j to C_i.

- s_i: for each data point within cluster i, the distance between the data point and the centroid of cluster i is measured . Then the average distance is measured.
- d_{ij}: the distance between cluster centroid i and j.

Then the term R_{ij} can be expressed as follows:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

Then the Davies-Bouldin Index is defined as follows:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \underset{i \neq j}{max} R_{ij}$$

Typically, zero is the lowest score for Davies-Bouldin Index and a **lower score** means a better clustering result.

5.1.3 Calinski-Harabaz Index

Proposed by Calinski and Harabasz in 1974, Calinski-Harabaz Index is calculated as follows if there are k clusters:

$$s(k) = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{N-k}{k-1}$$

where B_k is the between-group dispersion matrix, W_k is the within-cluster dispersion matrix, N is the total number of samples.

B_k and W_k are defined as follows:

$$B_k = \sum_{q} n_q (c_q - c)(c_q - c)^T$$

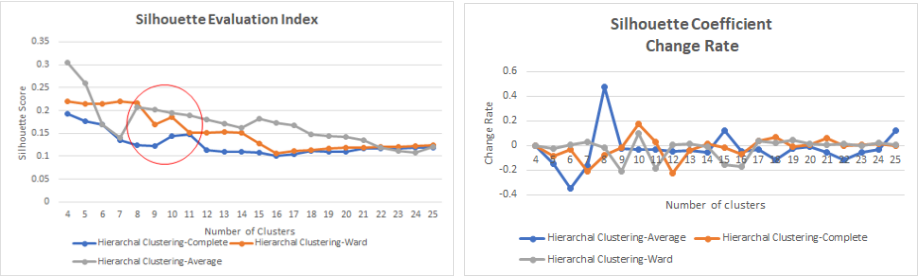
$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

where N is the number of samples, C_q is the set of samples in cluster q, c_q represents the centre of cluster q, n_q is the number of samples in cluster q[21].

The score can be computed quickly and a **higher score** means a better clustering result.

5.2 Number of Clusters Selection

The previous chapter mainly discusses three conventional clustering algorithms. However, since most clustering methods are unsupervised learning techniques, they require the number of clusters k to be specified ahead of time. The optimal number of clusters can be estimated from the evaluation indices, but due to the variety of the given data sets, no indices can accurately reflect the quality of the clustering results. To improve the reliability of the selected number of clusters, multiple indices are used to find an agreement on some certain numbers of clusters. If there is barely any agreement between the indices, it might indicate that no significant clustering structure exists in the given data set[22]. As not all indices show a potential optimal number of clusters, only the indices that are more likely to find the optimal number of clusters are considered.



(a) Silhouette Scores (b) Silhouette Change Rate

Figure 22: Silhouette Coefficient Evaluation

- In the above figure (a), with the increase in the cluster number, the Silhouette scores mainly decrease for different clustering algorithms. As the higher Silhouette Coefficient means a better clustering result, only the first highest peak is considered because apparently a very small number of clusters(e.g. 2 or 3) for more than 100 stocks is not a reasonable choice. The red circle indicates the first highest peak for Silhouette Coefficient and 8,9,10,11 clusters stand out among the given numbers.
- In the above figure (b), the change rates also show that Silhouette Coefficient remains steady after 12 clusters, which means that changing the number of clusters after 12 clusters has less influence.

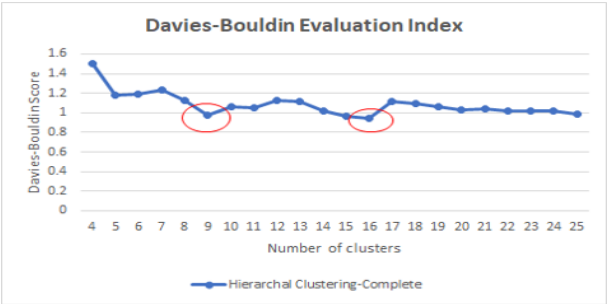


Figure 23: Davies-Bouldin score for different numbers of clusters

- In the above figure, 9 clusters and 16 clusters perform better than other numbers of clusters as the lower Davies-Bouldin score shows a better clustering result.

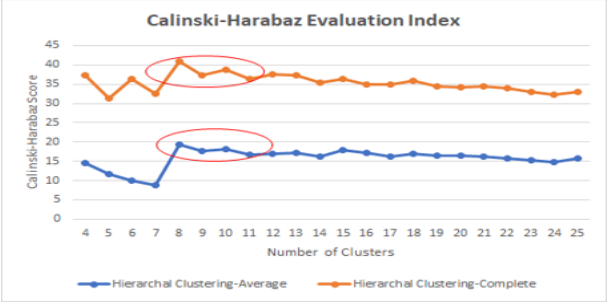


Figure 24: Calinski-Harabaz score for different numbers of clusters

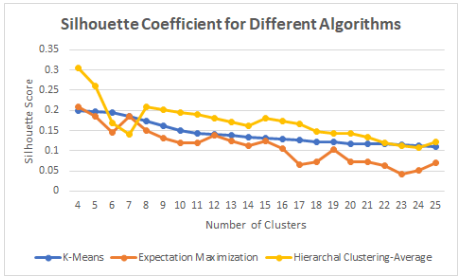
- In the above figure, 8,9,10 clusters mainly achieve higher Calinski-Harabaz scores than other numbers of clusters. Since the higher score means a higher quality of cluster, it is clear that these numbers of clusters are better.

Overall, it is found that 8-11 clusters have higher quality than other numbers of clusters and **10 clusters** are chosen for further analysis.

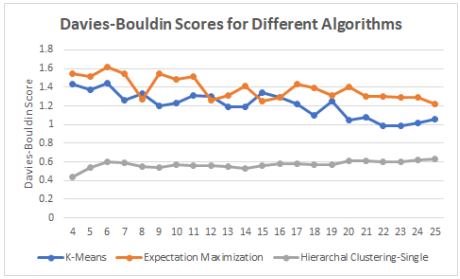
5.3 Performance Evaluation

5.3.1 Comparison of Clustering Algorithms

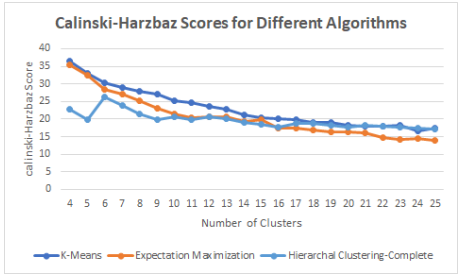
With different numbers of clusters, three different evaluation indices are calculated to compare the performance of different clustering algorithms. It is worth mentioning that the linkage method used in hierarchical clustering with the best performance is chosen for comparison with the other two algorithms.



(a) Silhouette Coefficient



(b) Davies-Douldin

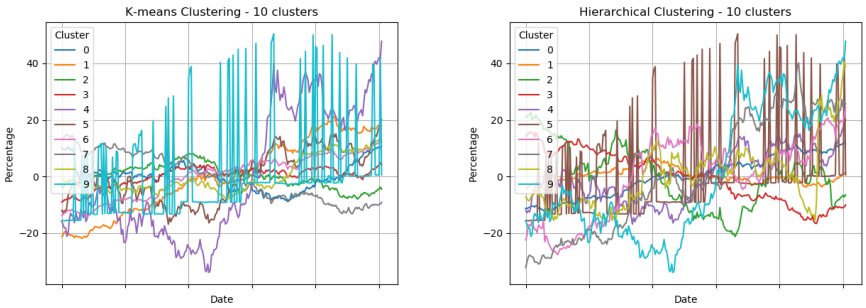


(c) Calinski-Harzbaz

Figure 25: Comparison between Algorithms

Taking these three figures together, the results suggest that the Hierarchical Clustering algorithm performs better than the other two; the results showing the worst quality are generated by Expectation Maximisation.

As K-Means and Hierarchical Clustering perform better than Expectation Maximisation, these two methods and 10 clusters are chosen for more detailed analysis.

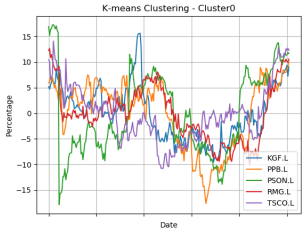


(a) K-Means (b) Hierarchical Clustering

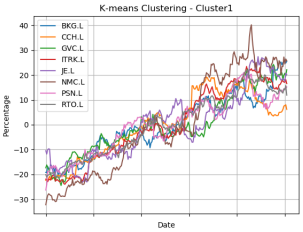
Figure 26: Ten Patterns for K-Means and Hierarchical Clustering

The above figures show the 10 patterns for K-Means and Hierarchical clustering. It is clear that they both generate distinguishable patterns and some of them are quite similar.

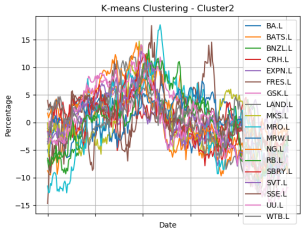
5.3.2 Visualisation For K-Means



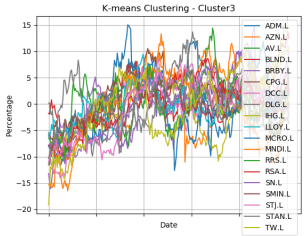
(a) Cluster 0



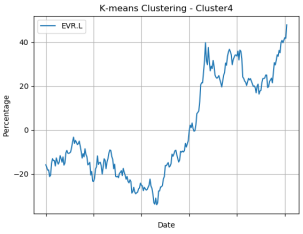
(b) Cluster 1



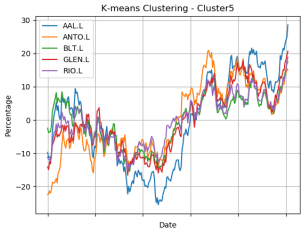
(c) Cluster 2



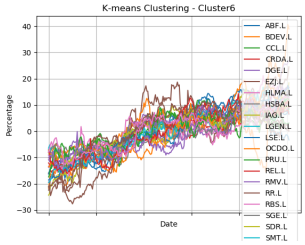
(d) Cluster 3



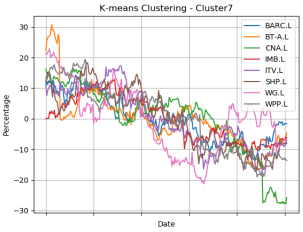
(e) Cluster 4



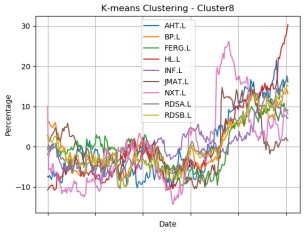
(f) Cluster 5



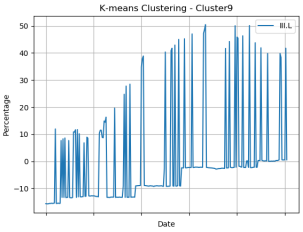
(g) Cluster 6



(h) Cluster 7



(i) Cluster 8



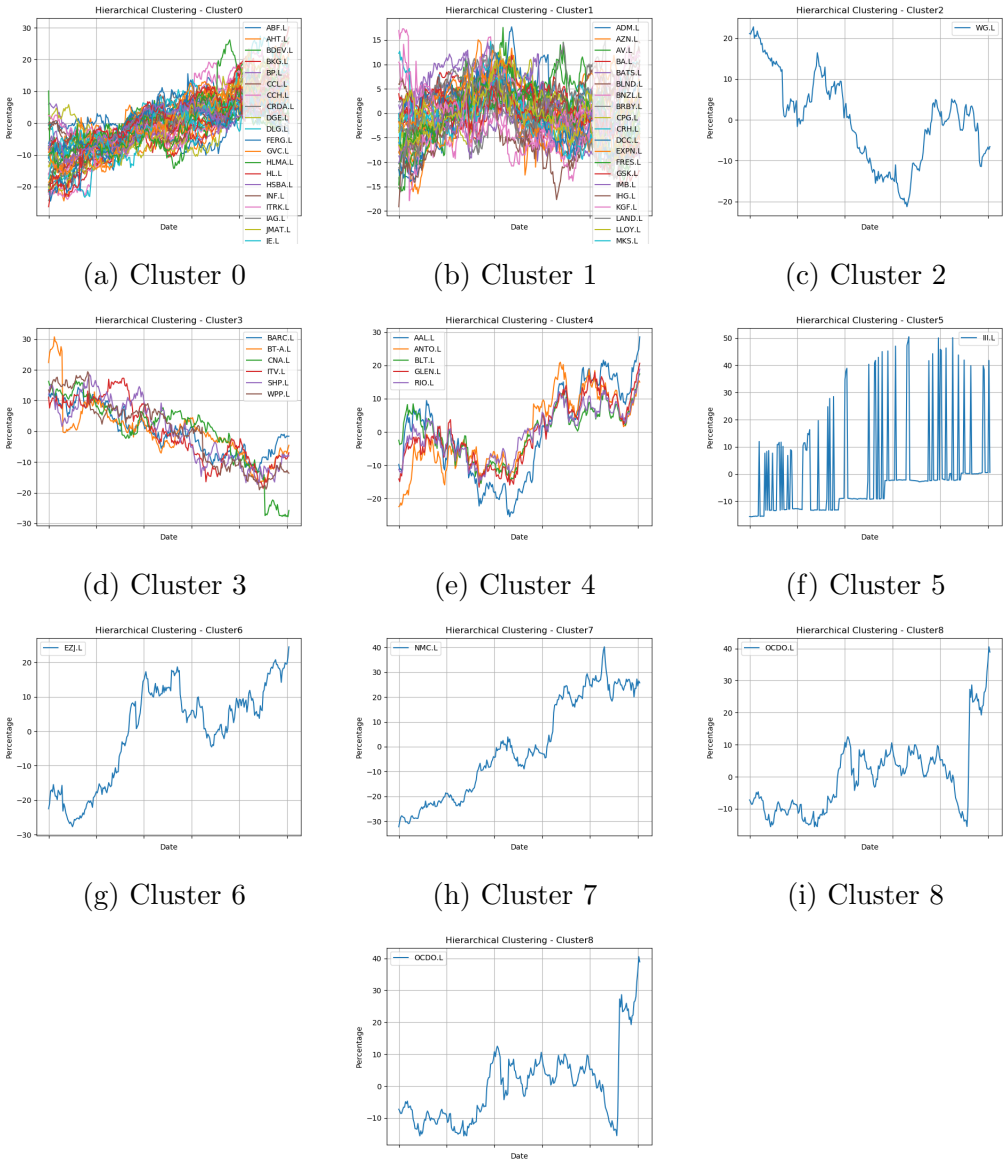
(j) Cluster 9

Figure 27: 10 Clusters For K-Means

The above figures show the specific stocks and their price movements in each cluster. Some initial observations are given as follows:

- For K-Means method, several clusters show satisfactory results such as cluster 1, cluster 5, cluster 6, and even cluster 8. Stocks in these clusters have similar movements so they are grouped correctly.
- However, it is clear that some other clusters, such as cluster 0 and cluster 3 contain some stocks with different patterns but they are still grouped into the same cluster, which is unexpected.
- In addition, due to the design of the K-Means method, the initial centroids have a strong impact on the final results, which means running K-Means multiple times may produce different results and sometimes the results converge to a local optimum rather than a global optimum.

5.3.3 Visualisation For Hierarchical Clustering



(a) Cluster 0 (b) Cluster 1 (c) Cluster 2

(d) Cluster 3 (e) Cluster 4 (f) Cluster 5

(g) Cluster 6 (h) Cluster 7 (i) Cluster 8

(j) Cluster 9

Figure 28: 10 Clusters For Hierarchical Clustering

The above figures show the results for the Hierarchical Clustering method. The findings from these figures suggest the following:

- Hierarchical clustering, compared with the K-Means algorithm, shows less satisfactory results. At the same time, cluster 3 and cluster 4 still produce high quality results.
- However, the common problem is that for cluster 0 and cluster 1, too many stocks are gradually grouped into one cluster, leading to inaccurate results.
- By contrast, the other clusters, such as cluster 2, cluster 6, cluster 7, cluster 8 and cluster 9 contain only one stock in each cluster.

6 New Algorithm Design - Hierarchical Based K-Means

In this chapter, a new algorithm is designed in order to overcome the limitations of both K-Means and hierarchical clustering. The origin of this algorithm is explained and a simple example given to simulate the process of the algorithm. Then the pseudocode for this algorithm is provided. Finally, a comparison is made between the new algorithm and conventional clustering algorithms.

6.1 Algorithm Design

6.1.1 From Human-Machine Interaction to New Design

The following figure indicates the general idea behind the design of this new algorithm.

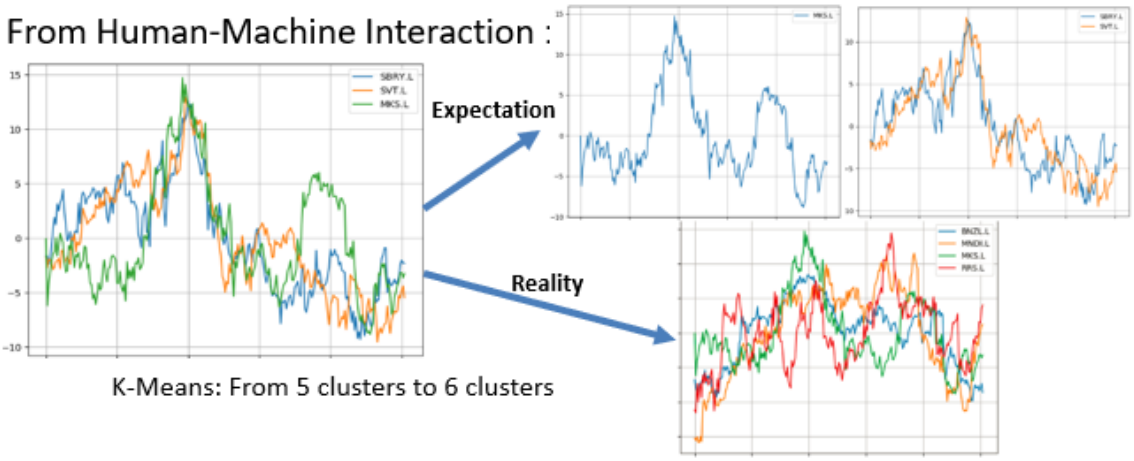


Figure 29: Idea behind New Design

The left sub-figure shows one cluster among the results after the K-Means algorithm is applied to the FTSE100 data set with the desired 5 clusters. If humans can be involved in the clustering process, they can evaluate the quality of the clustering results at each step and adjust the results manually. In the left sub-figure, it is clear that the stock coloured green should not belong to this cluster. In this situation, it is found that increasing the number of clusters by one gives a better clustering result. The K-Means is then applied to the same data set again with 6 clusters. The upper right two figures are the expected results in which the three stocks within one cluster is divided into two sub-clusters and the stock with green colour becomes one cluster. However, the lower right figure is the real result and it shows disappointing results in that other stocks are grouped into this cluster and the previous cluster is not divided as expected.

Based on the above analysis, some new ideas are generated from the process of Human-Machine Interaction:

- **Idea 1**:
  - Hierarchical approach allows us to construct nested partitions layer by layer; thus, the data points can be grouped into a tree of clusters. From

the process of Human-Machine Interaction, it can be found that the hierarchical approach helps us adjust the result at each step so better results can be achieved. For example, if 6 clusters are needed, the number of clusters can be gradually increased to 6 rather than setting 6 directly.

– For K-Means algorithm, the initial centroids have a strong impact on the final results, but the experiment indicates that a small number of centroids (e.g. 2) or a very large number of clusters (e.g. near the given number of stocks) have less influence on the final results. The following figure shows the standard deviation against different numbers of clusters for the K-Means algorithm. It is almost certain that with the increase in the number of clusters and a small number of clusters, the volatility of the clustering results decreases.

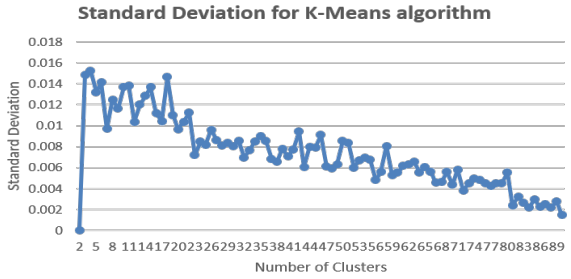


Figure 30: Standard deviation with different numbers of clusters

- **Solution 1**: The new algorithm can adopt the idea behind hierarchical clustering, which means a small number of clusters at the start, gradually increasing to the target number of clusters.
- **Idea 2**:
  - Solution 1 allows us to start from a small number of clusters and increase to the target number of clusters, but at each step, the random centroids generated by K-Means do not consider the results produced in the previous step.
  - If the centroids can be controlled at each step, the results based on the previous steps can be used.
- **Solution 2**: The centroids of each step can be calculated by considering the mean values of all stocks in one cluster and the K-Means allows us to set the computed centroids as the initial centroids.
- **Idea 3**:
  - At each step, as only the number of clusters is increased by 1, it is necessary to decide which cluster should be divided. For example, as the previous example shows, the number of clusters increases from 5 to 6. In the previous step, there are 5 clusters, so it is necessary to decide which cluster needs to be divided into two sub-clusters.
- **Solution 3**: Since the greedy algorithm always makes the optimal choice at each step, it is good practice to adopt this algorithm.

6.1.2 Lightning example

The following figure shows a simple example that illustrates the process of the new algorithm.

- **Input**:
  - The whole stock data set.
  - The starting number of clusters S (normally 2).
  - The target number of clusters T.
- **Output**:
  - The labels for each stock.
  - The centroid for each cluster
- **Error Score**: Sum of squared error
- **Difference**: old error - new error score (E.g. E11 - E12)
- **Example**: starting number: 2; target number: 3.

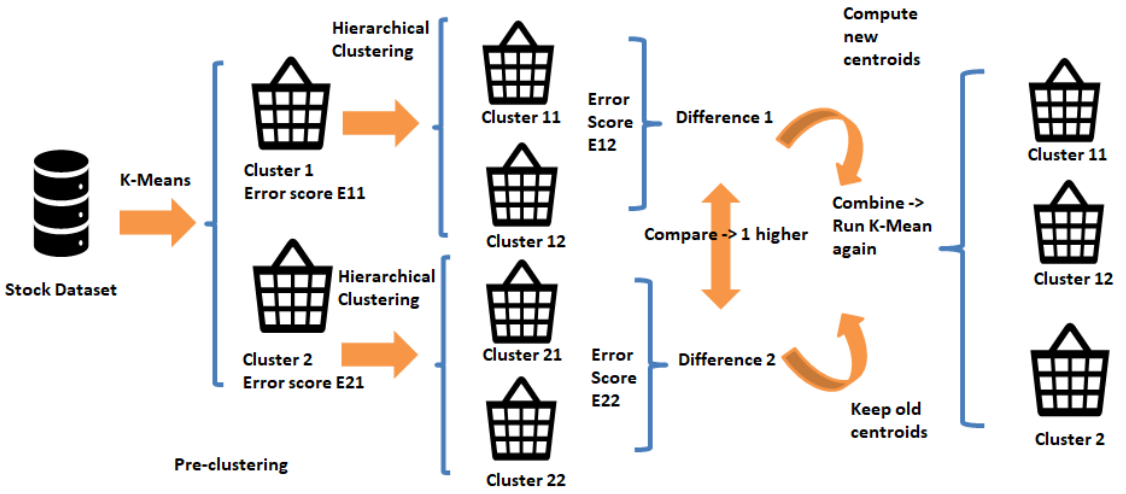


Figure 31: Hierarchical Based K-Means - Simple Example

- The first stage of this algorithm is the pre-clustering stage. During the first stage, the given stock data set is grouped into 2 clusters by using the K-Means algorithm.
- Then two error scores E11 and E21 are computed.
- Each cluster(1 and 2) is divided into two sub-clusters by using hierarchical clustering. The linkage method with the best clustering results is chosen.
- Then the new error scores, based on the four sub-clusters, are calculated.
- The next step is to compute the two scores (Difference 1 and Difference 2) respectively and to compare them.

- The cluster with the higher difference score is finally divided and the new centroids are calculated. The cluster with the lower difference score remains unchanged.
- The new centroids and old centroids are combined and the K-Means is run again with the new computed centroids.
- Finally, the given stock data set is grouped into 3 clusters.

6.1.3 Pseudocode

Algorithm 1: Hierarchical Based K-Means

Input: the whole stock data set; S : the starting number of clusters; T : the target number of clusters

Output: the labels for each stock; the centroid for each cluster

```
1  the given data set is clustered into S groups by using the K-Means algorithm;
2  current number of clusters N = S;
3  while N ≠ T do
4      for each cluster i do
5          compute sum of squared error E1;
6          split it into 2 groups by using hierarchical clustering;
7          compute sum of squared error E2;
8          difference[i] = E1 - E2;
9      cluster_to_divide = max(difference[i]);
10     split cluster i into two groups;
11     compute the current number of clusters N and their centroids as initial
        centroids ;
12     cluster the given data set into N groups by using K-Means with
        pre-computed initial centroids;
13     update current number of clusters N;
14 return the labels and centroids from the last K-Means result;
```

- The above shows the pseudocode for this Hierarchical Based K-Means. It is important to note that the process will stop until the current number of clusters is the same as the target number of clusters.
- The theory behind the greedy algorithm is utilised in line 9 as in each iteration only the cluster with the highest difference score is divided, which means that dividing this cluster causes a maximum decrease in the sum of squared errors.
- The implementation of this algorithm is in appendix E.

6.2 Performance Evaluation

With the aim of achieving the best clustering results, the starting number of clusters for FTSE100 needs to be defined ahead of time. In order to choose the best starting number, 8 numbers from 2 to 9 are chosen for testing. The following figure shows the comparison between different numbers and for the data set FTSE100, 2 is chosen as the starting number of clusters for further analysis:

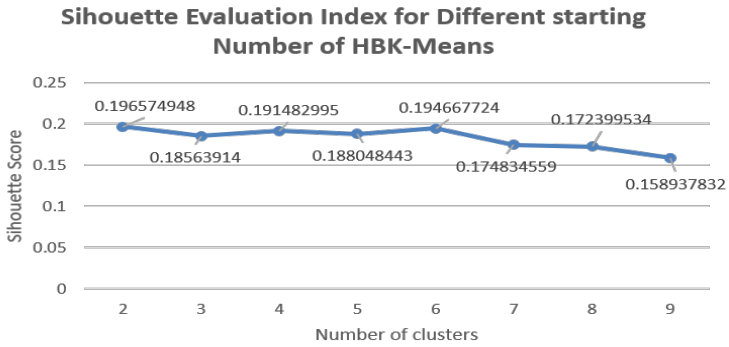
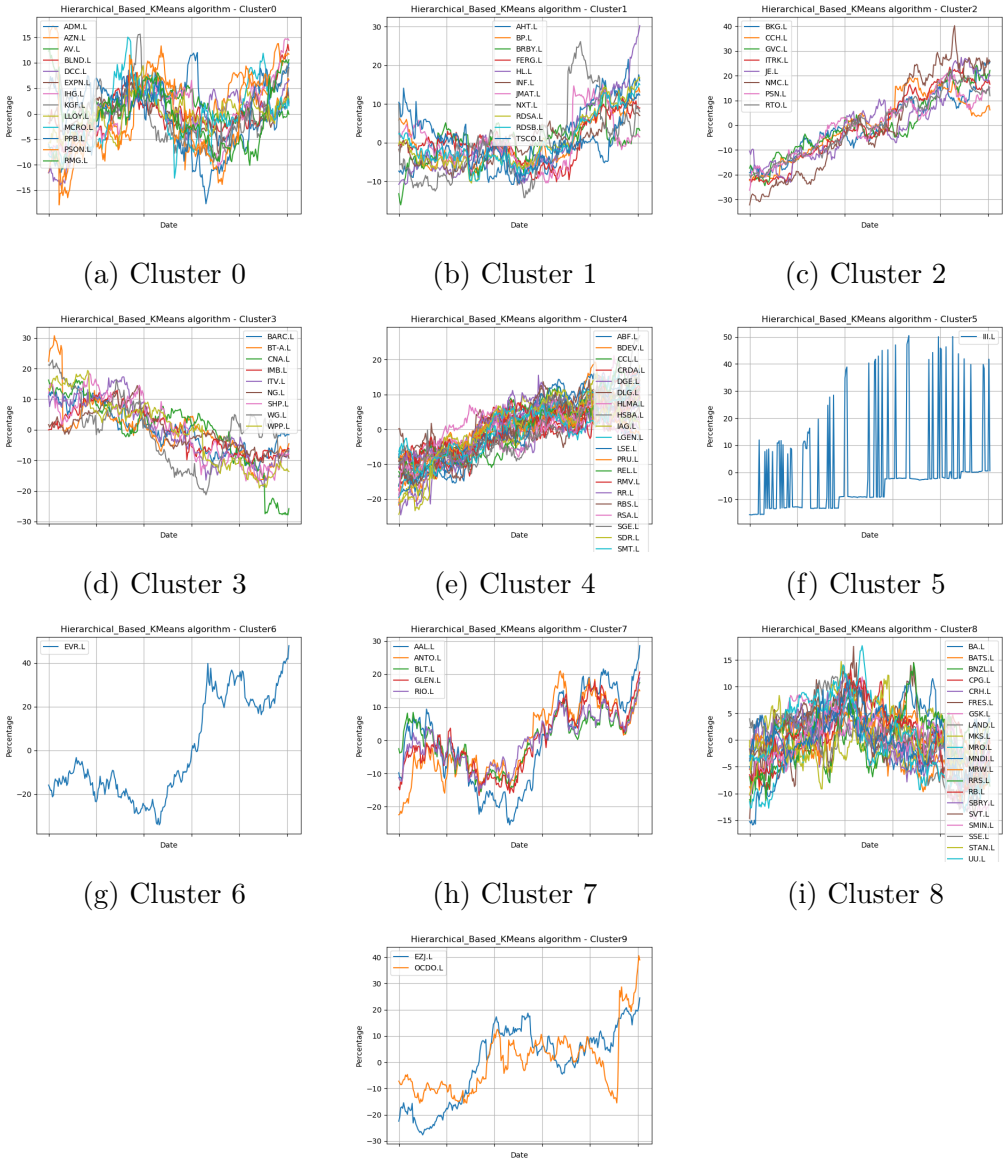


Figure 32: Starting Number Selection

6.2.1 Visualisation For Hierarchical Based K-Means



(a) Cluster 0 (b) Cluster 1 (c) Cluster 2

(d) Cluster 3 (e) Cluster 4 (f) Cluster 5

(g) Cluster 6 (h) Cluster 7 (i) Cluster 8

(j) Cluster 9

Figure 33: 10 Clusters For Hierarchical Based K-Means

The above figures show the results for the Hierarchical Based K-Means method. Compared with traditional K-Means and Hierarchical Clustering, the new designed algorithm overcomes the limitations discussed before. The findings from these figures suggest that most of the clusters show reasonable results, and there are only two clusters which contain only one stock.

6.2.2 Evaluated with Internal Indices

With the aim of proving that the newly designed algorithm stands out among conventional clustering algorithms, quantitative analysis based on the internal indices applies to all clustering algorithms. The following three figures show the scores for different internal indices against different clustering algorithms.

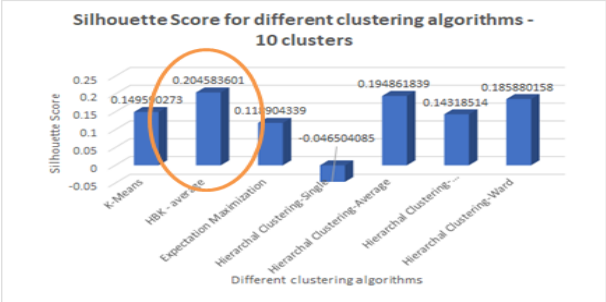


Figure 34: Silhouette Coefficient for different algorithms - plus HBK

- The Hierarchical Based K-Means achieves the highest Silhouette score which shows that its performance is better than the others.

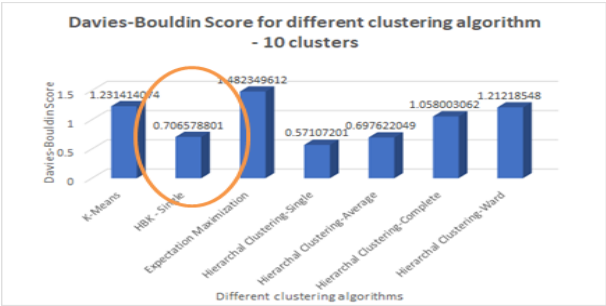


Figure 35: Davies-Bouldin Score for different algorithms - plus HBK

- The Davies-Bouldin score for Hierarchical Based K-Means stays at a low level; only two other methods show slightly lower scores.

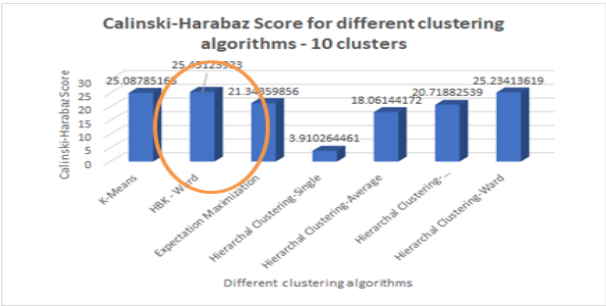


Figure 36: Calinski-Harabaz Score for different algorithms - plus HBK

- Again, the newly designed algorithm produces the highest score for Calinski-Harabaz score which proves it performs better than the others.

6.2.3 Evaluated with Labelled Dataset

With the internal indices and pattern visualisation, the newly designed method has been proved to achieve better clustering results. However, in order to show a more convincing conclusion, the algorithm must be applied to a different time series data set and then it can be widely used if it produces better results. The data set is mapped from the spectral evolution of a "pixel" to a time-series data that represents the wheat crop and this data set is used to prove that this new algorithm is better than the others. The following figure shows the production of a time-series data set from a satellite image series. The change in pixel colour reflects the growth of crops with temporal evolution[23].

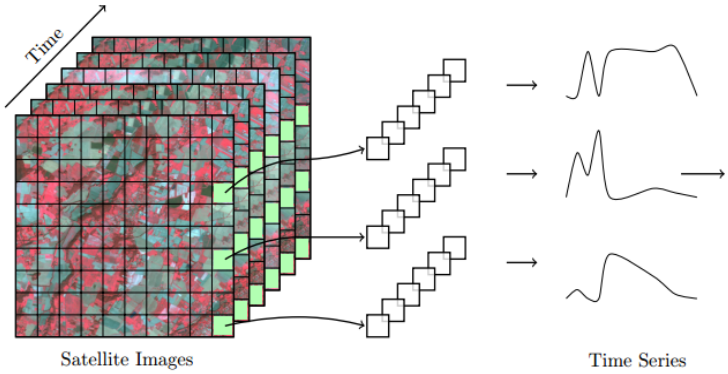


Figure 37: Production of time series data sets from a satellite image series

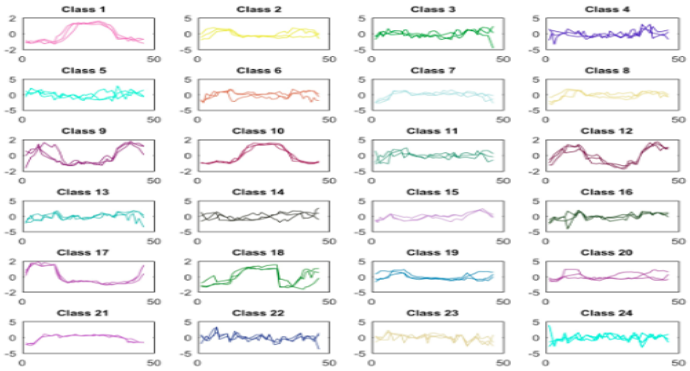


Figure 38: Time Series Dataset for Wheat Crop

The above figure shows the 24 classes, which represent 24 patterns for a crop data set. Each series data has a pre-defined label showing which pattern it should belong to. Then the K-Means and Hierarchical Based K-Means can be applied respectively to those labelled data and these can then be grouped into 24 clusters. By checking the accuracy of each clustering method, it can be found that the new method(57.70%) achieves higher accuracy than the original K-Means(53.07%).

7 Dynamic Pattern Analysis

In this chapter, unlike in previous chapters, the focus is on the dynamic patterns of each stock. Two strategies are used to find the stable stocks which can be used for further analysis. Some examples are then given of the analysis carried out, based on the dynamic patterns.

7.1 Pattern Recognition for FTSE 100

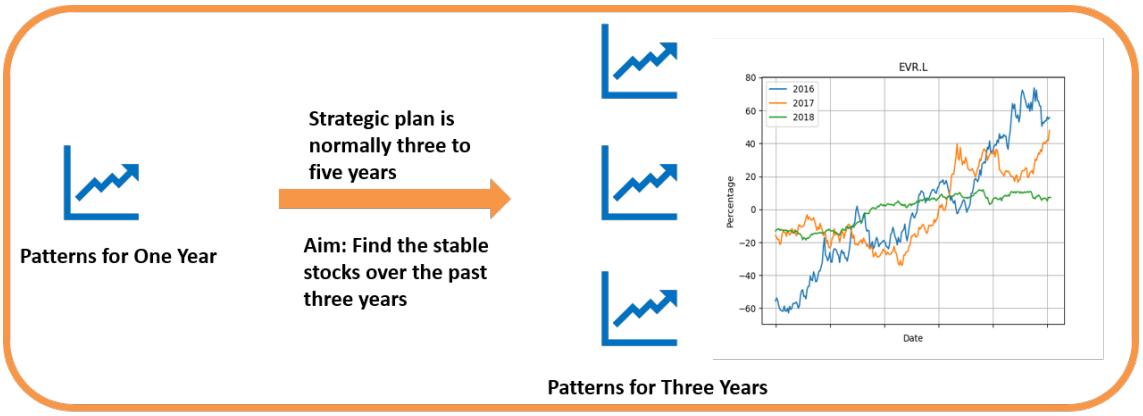


Figure 39: Dynamic Pattern Analysis

The term "dynamic pattern" is used to refer to the pattern change for three years as some studies indicate that the strategic plans for some companies are normally three to five years. The above figure shows the three-year dynamic pattern for Evraz, a multinational vertically integrated steel making and mining company with headquarters in London. In this project, stock prices in the FTSE100 are collected from 2016-2018 and further analysis is all based on this data set.

In general, the patterns for stocks can be divided into three categories: increasing pattern, steady pattern, and decreasing pattern. The following figure shows an example of the above three patterns over the past three years:

Figure 40: Example with three patterns

Based on the initial pattern analysis, two strategies are used to help us find the promising stocks:

- **Strategy 1**: focus on the stocks with only increasing patterns over the past three years.
  - 2016 clusters: 1,3,6,7 (e.g. EVR.L, BP.L)
  - 2017 clusters: 1,2,4,6,8 (e.g. EVR.L)
  - 2018 clusters: 2,6 (e.g. EVR.L, TSCO.L)

  For each year, a set of stocks can be computed from the chosen clusters, and the intersection between each year's stock sets is noted. However, unfortunately only one stock (EVR.L) is found from this strategy. It seems possible that from the dynamic pattern analysis, this stock shows a promising trend, so some further analysis, such as investment strategy analysis, can be applied to this stock because it is more likely to return more profits.

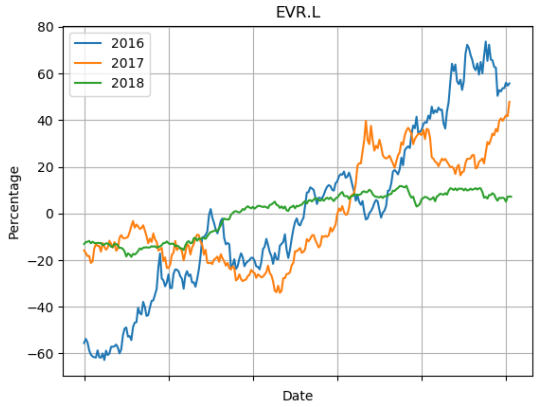


Figure 41: Three-year dynamic patterns for EVR.L

- **Strategy 2**: focus on the stocks with steady and increasing patterns over the past three years.
  - 2016 clusters: 0,1,3,4,5,6,7,9
  - 2017 clusters: 0,1,2,4,6,8
  - 2018 clusters: 0,2,6

  In the second strategy, it is possible to choose from where the stocks are selected if they do not show decreasing patterns. Then a set of stocks with a stable (non-decreasing) pattern are chosen for further analysis. The following figure shows the symbols and cluster indices of each selected stock; the specific patterns are in appendix D.

index	Cluster201	Cluster201	Cluster201
ADM.L	0	0	0
AZN.L	0	0	6
BNZL.L	0	0	6
CCH.L	1	2	0
CPG.L	0	0	0
CRDA.L	0	4	0
DGE.L	0	1	0
EVR.L	3	6	6
EXPN.L	1	0	6
HLMA.L	0	4	0
LSE.L	5	4	0
MRW.L	5	0	0
RDSB.L	1	1	0
RB.L	0	0	0
REL.L	0	4	0
RTO.L	1	2	0
RMV.L	0	4	0
SMT.L	1	4	0
SVT.L	0	0	0
SN.L	0	4	0
ULVR.L	0	4	0

Figure 42: A set of stable stocks

7.2 Analysis on Dynamic Patterns

In the previous section, stocks with similar patterns over three years could be grouped into one set. Based on the stocks with similar patterns, more valuable information can be extracted from those patterns.

- Dynamic pattern analysis provides a way to select promising stocks based on movements over three years; this long term dynamic change means that stocks with stable patterns can be chosen. Compared with other stocks, the chosen stocks have more potential to rise and carry less risk. Further investment strategies and portfolio selection are all based on the results of dynamic pattern analysis.

- A stock that rose over the past three years is more likely to rise this year.

(a) Prices over the past three years (b) This year's prices

Figure 43: Stock price movements for EVR.L

The above two figures show an example of this observation. The observation is reasonable as persistent increasing patterns over three years indicate a successful strategic plan for this company. Under this circumstance, the company is more likely to make profits in the following year.

- Companies in the same industry (e.g. financial sector, retail sector) may have similar patterns, especially without an ideal economic environment.

Figure 44: Stock prices for financial companies in 2018

From the above figure, it can be seen that the stock prices for financial sectors decreased in 2018. As the economic environment was not ideal in 2018, it is found that most financial houses have a highly consistent pattern.

- Three-year cluster distribution indicates the economic development of a stock market, and thus, reflects the economic environment in the UK. The following figure presents the three-year cluster distribution and some useful information can be found from this figure, along with each year's specific pattern.
  Please note that the same cluster index does not mean their patterns are the same.

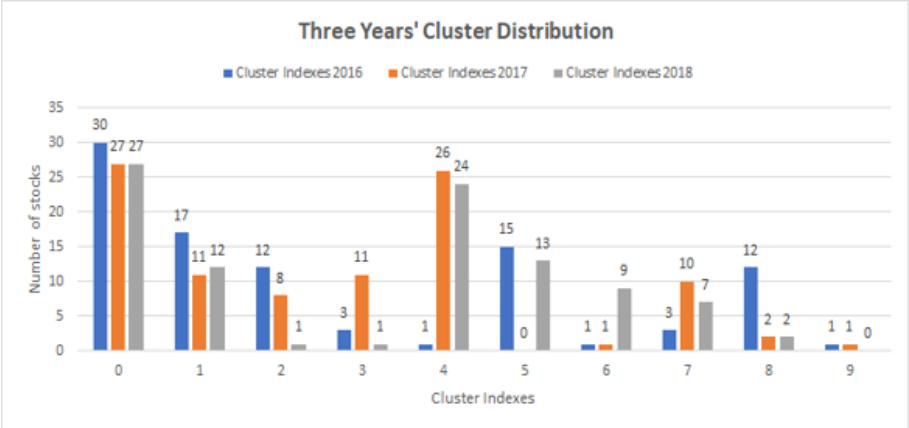


Figure 45: Three-Year Cluster Distribution

For example, as the above figure shows, most stocks are in cluster 0 and cluster 4 in year 2017 and year 2018. The stock patterns for those two clusters can be checked in appendix B and appendix C. It is then found that in 2017 most stocks show an increasing or stable pattern while in 2018 most stocks show a decreasing pattern, which indicates that the economic environment in 2018 is not favourable.

8 Applications on Stock Patterns

This chapter discusses some useful applications based on stock patterns. In the previous chapter, dynamic patterns for three years are investigated due to the fact that the strategic plan for large firms is normally three to five years. In the dynamic pattern analysis discussed previously, we selected one stock (EVR.L) from strategy one and a set of promising stocks from strategy two. In this chapter, we explore some investment strategies, such as daily investment and then, we investigate portfolio selection based on the Modern Portfolio Theory. With some stocks selected from strategy two, it is necessary to optimise the portfolio to find some extreme values using the Monte Carlo Simulation. In addition, the performance of each investment strategy and the outcome of each portfolio are evaluated by the Sharpe Ratio. The last section in this chapter focuses on exploring possible factors that may influence or reflect the movements of stock prices. As there has been a huge rise in the use of social media such as Twitter in recent years, the sentiments on Twitter may have potential correlations with stock patterns.

8.1 Sharpe Ratio

Firstly we introduce a financial concept called the Sharpe Ratio. This allows us to evaluate the performance of a financial strategy. It is widely used in investment strategy analysis and portfolio selection. Developed by William F. Sharpe, the Sharpe Ratio is designed to examine the performance of an investment by considering its return against its risk[24]. More importantly, it provides a more straightforward way of comparing two different investments and shows numerical results.

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

where R_p is the return of a portfolio, R_f represents the risk-free rate and σ_p is the standard deviation of the portfolio's excess return. The risk-free rate is normally zero with respect to stock portfolios.

Both a **higher** expected return and lower volatility result in a **higher** Sharpe Ratio. Thus a **higher** Sharpe Ratio indicates a better strategy.

8.2 Investment Strategy Analysis

8.2.1 Basic Strategies

Basic strategies define regular trading behaviours, where we buy and sell a stock at a pre-defined time no matter how the stock market changes.

- **Daily Investment**: For this strategy, we always buy and sell at the close price for a given day. This occurs on all trading days. The return for this investment can be calculated as follows:

$$Daily\ Return = \frac{Today's\ close\ price - Yesterday's\ close\ price}{Yesterday's\ close\ price}$$

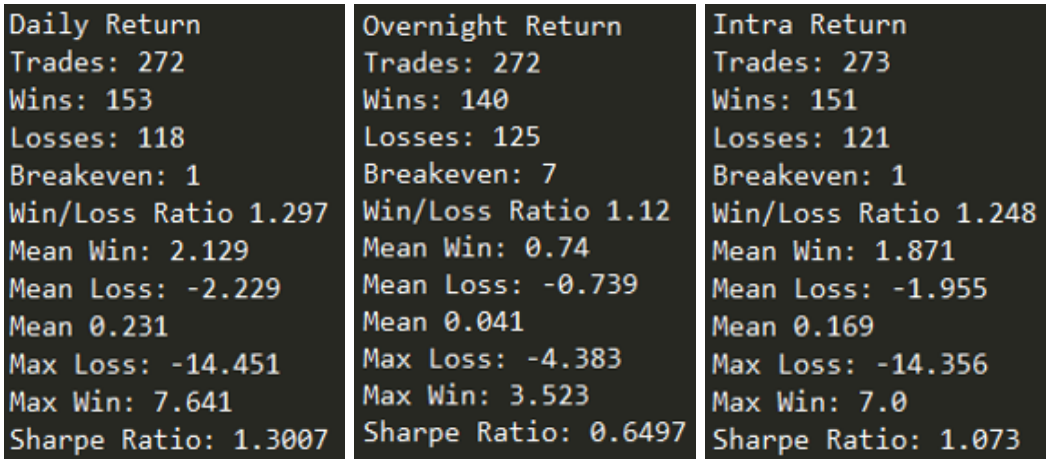
- **Overnight Investment**: For this strategy, we always buy at the close price on one day and sell at the open price the following day. This occurs on all trading days. The return can be calculated at follows:

$$Overnight\ Return = \frac{Today's\ open\ price - Yesterday's\ close\ price}{Yesterday's\ close\ price}$$

- **Intraday Investment**: In this case, we always buy at the open price on one day and sell at the close price the same day. This occurs on all trading days. The return can be calculated at follows:

$$Intraday\ Return = \frac{Today's\ close\ price - Today's\ open\ price}{Today's\ open\ price}$$

From the above three basic strategies and the promising stock selected from previous chapters, some statistics can be obtained. These statistics allow us to compare the performance of each strategy and give some suggestions for future investment.



(a) Daily Investment (b) Overnight Investment (c) Intraday Investment

Figure 46: Statistics for Basic Strategies

The above three figures present the detailed statistics for three basic strategies. Each strategy provides some basic information for one year such as the number of trading activities performed, and the number of positive returns. However, the most valuable information used to compare the performance of each strategy is the Sharpe Ratio. It is obvious that the daily investment strategy performs better than the others for stock EVR.L as it shows the highest Sharpe Ratio of the three.

8.2.2 Advanced Strategies

The previous section describes three basic investment strategies; in this section, a more advanced method is attempted in order to see if it is possible to obtain a higher Sharpe Ratio.

- **Prediction Based Investment**: based on machine learning techniques, this advanced model provides a more reasonable strategy to help investors make decisions.
  - Support Vector Machine: As a supervised machine learning technique, the support vector machine can be used for both regression and classification. Since this chapter involves the prediction of stock prices, regression is discussed.

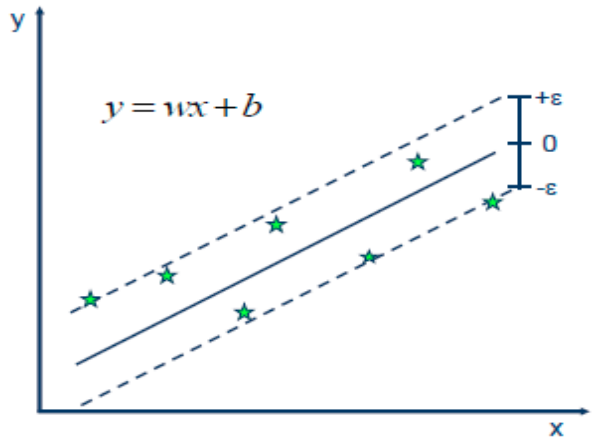


Figure 47: Example of the optimal hyperplane

In the case of regression, a margin of tolerance (epsilon) needs to be set ahead of time in the Support Vector Machine, and the convex optimisation problem solved[25]:

Minimise:

$$\frac{1}{2}||w||^2$$

Subject to:

$$y_i - wx_i - b \leq \epsilon$$
$$wx_i + b - y_i \leq \epsilon$$

Based on the above figure and equations, a line of best fit can be found and used for prediction.

– We use 2016-2017 stock prices as training data and 2018 stock prices as testing data. The following figure shows the stock EVR.L with predicted next day close prices and real close prices. It is worth mentioning that the predicted prices are similar to the real prices as only today's prices are used to predict tomorrow's prices. The Implementation of predicting stock prices by Support Vector Machine is in appendix F.

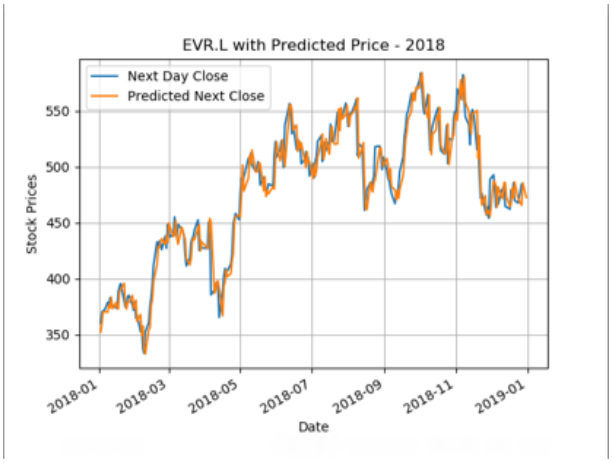


Figure 48: EVR.L with predicted prices

– The predicted close prices are compared with the real open prices to determine whether it is favourable to buy or not.

Figure 49: Statistics for Predicted Based Investment

As the above figure illustrates, the prediction based method shows a higher Sharpe Ratio than the three basic strategies.

This chapter describes four investment strategies that aim to provide some inspiration for financial investment. With the stock EVR.L, the predicted-based investment strategy shows better performance than the other three, but it may not be suitable for other stocks.

8.3 Portfolio Selection and Optimisation

8.3.1 Modern Portfolio Theory

Pioneered by Harry Markowitz in 1952[26], the Modern Portfolio Theory is a mathematical framework which helps investors construct portfolios (a combination of stocks) to maximise expected returns with respect to a given level of risk.

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

where R_p represents the return on this portfolio, R_i is the return on each component i, and W_i refers to the weighting of each component asset i.

The theory may allow us to construct an "Efficient Frontier" of an optimal portfolio. With the efficient frontier, the maximum expected return may be found for a given level of risk or vice versa.

8.3.2 Monte Carlo Simulation

Monte Carlo simulation is a technique used to model the probability of different outcomes in a process that involves repeated random sampling. This technique helps us understand the impact of uncertainty and risk in financial models. It helps us estimate the outcomes based on a range of values rather than one value. Estimating a range of values allows us to create a more realistic picture of portfolio outcomes such as expected return and volatility[27].

In Monte Carlo simulation, the model is calculated based on random values. For the portfolio selection task, each simulation generates a random weighting vector based

on the number of stocks chosen. The weight of each stock represents the proportion of capital invested in this stock.

8.3.3 Portfolio Optimisation

- Five stocks example:
  - Five stocks are chosen from the results of Dynamic Pattern Analysis (Strategy two).
  - Five stocks: [ADM.L, AZN.L, BNZL.L, CCH.L, CPG.L]
  - One year investment: For each stock we use Daily Investment Strategy, where we always buy and sell at the close price on a given day.
  - 2500 simulations

  For each weighting vector, it is imperative that the weights add up to 1 all the time. The following figure shows the results of 2500 simulations. For each simulation, its expected return, expected volatility,Sharpe Ratio are calculated and plotted onto a figure.

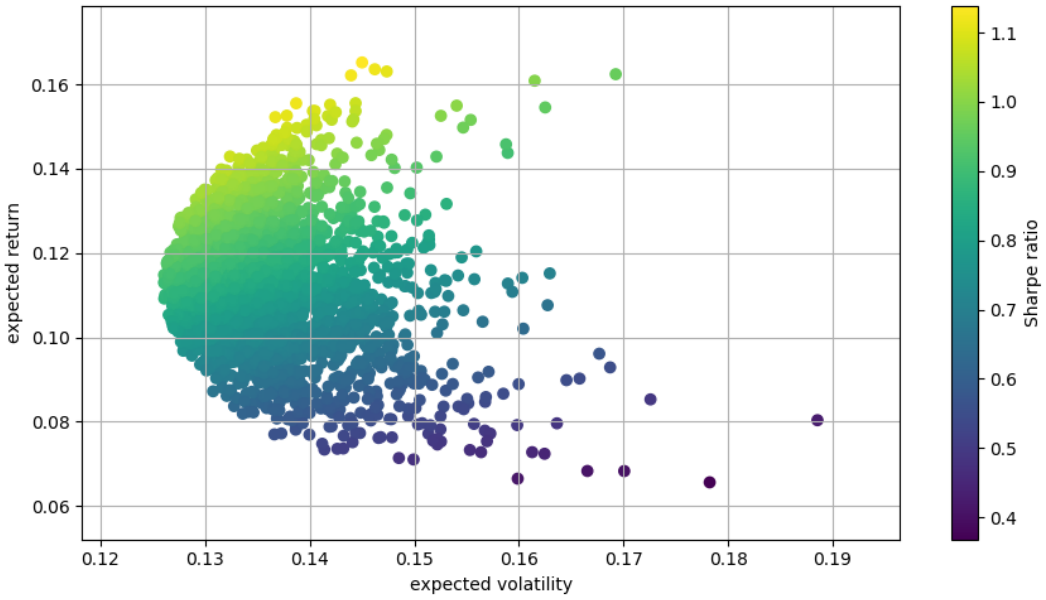


Figure 50: Monte Carlo Simulation for Portfolio Selection

The above figure presents the results of 2500 simulations and each coloured point represents one simulation. The colour change from dark blue to yellow shows an increasing Sharpe Ratio. However, not all simulations produce good results, especially the points in the lower right corner.

- Optimisation: The expected return is set from **0.05 - 0.18** and from this range, it is necessary to find the optimal portfolio which either has the highest Sharpe Ratio or the lowest volatility.
  - Optimisation Method: **SLSQP** (Sequential Least Squares Programming)
    - This employs the Han–Powell quasi–Newton method with a BFGS update of the B–matrix and an L1–test function in the step–length algorithm[28].

- **Maximising the Sharpe Ratio**: this problem is the same as minimising the negative value of the Sharpe Ratio; the constraint is that all weights should add up to 1 and the weight value should be greater than 0 but less than 1. The results for this case is [0.043 0.421 0.536 0. 0.], which means 4.3% of the money should be invested into the first stock, 42.1% should be invested into the second stock and so on.
- **Minimising the Variance**: this approach is similar to that used to maximise the Sharpe Ratio; the results of this case are [0.297 0.107 0.283 0.07 0.244], which means 5 stocks are all used in order to obtain the lowest volatility.
- **Efficient Frontier**: by iterating over multiple starting conditions, all portfolios with minimum volatility or maximum Sharpe Ratio for a given target return[29] are obtained. The results are shown in the following figure and the crosses represent the Efficient Frontier for expected returns between 0.05 and 0.18.

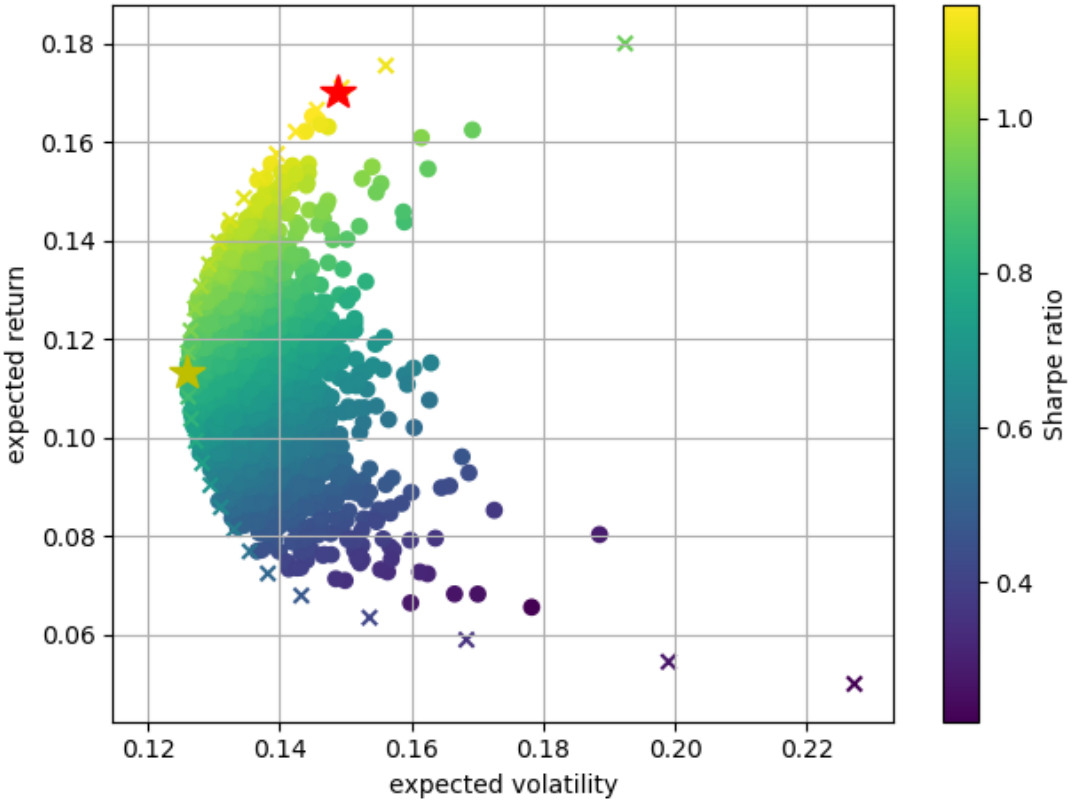


Figure 51: Monte Carlo Simulation for Portfolio Optimisation

With the expected returns between 0.05 to 0.18, 2500 Monte Carlo simulations are run and generate 2500 different weighting vectors. The above figure presents the results of 2500 simulations and each coloured point represents one simulation.

The coloured crosses indicate the optimal portfolios with a certain target return. as the target returns are fixed between 0.05 and 0.18, the above figure only shows the simulations within that interval. The yellow star indicate the portfolio with the lowest volatility and the red star represents the portfolio with the highest Sharp Ratio. The implementation of this optimisation can be found through appendix G.

8.4 Sentiment Analysis on Stock Patterns

The normalised stock prices data is already available, so the first step for sentiment analysis on Twitter is to collect the corresponding tweets of each stock. Twitter API allows us to connect and collect limited numbers of tweets for each request. Considering the speed and capability of each request, a Python package Twitterscraper is used for collecting a large number of tweets. By using this script, thousands of corresponding tweets over a pre-defined time period can be gathered instantaneously and stored in the JSON file.

After collecting the corresponding tweets of each stock, a simple pre-processing technique is applied to those tweets. The stop words can be removed as they do not show any sentiments. When the data set is ready, the approach used for sentiment analysis is VADER Sentiment Analyzer.

8.4.1 VADER

Introduced in 2014, VADER(Valence Aware Dictionary for sEntiment Reasoning) utilises qualitative analysis and empirical validation to calculate the sentiment score of an input text. Unlike the machine learning approach, it is a lexical approach which utilises a dictionary and maps lexical features to emotional intensities.[30].

The dictionary is created by human raters from Amazon Mechanical Turk, which is a crowdsourcing marketplace maintained by Amazon.

The sentiment score for each word is measured on a scale between -4 to +4, from most negative to most positive. The sentiment score of a sentence is calculated by summing up each word in the sentence, and normalising it to a value between -1 to 1. The equation for normalisation is expressed as follows:

$$\frac{x}{\sqrt{x^2 + k}}$$

where x is the sum of sentiment scores of constituent words in each sentence and k represents the normalisation parameter.

The following figure shows the results of sentiment analysis for Royal Bank of Scotland in 2017. It presents the proportion of each polarity but the positive and negative polarities are more important for us. The implementation of the Sentiment Analysis is in appendix H.

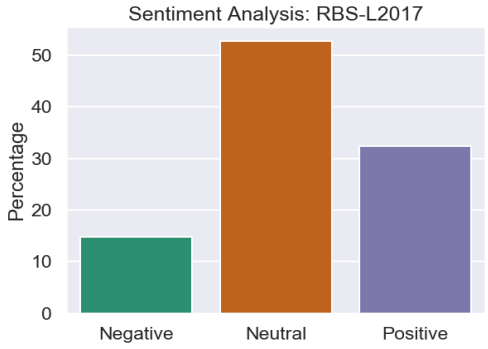


Figure 52: Results of Sentiment Analysis

8.4.2 Potential Correlation Analysis

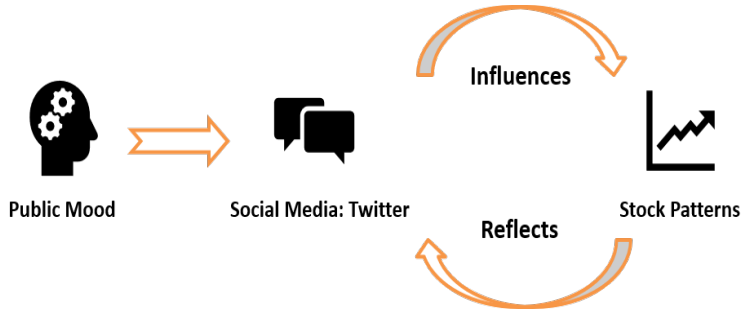
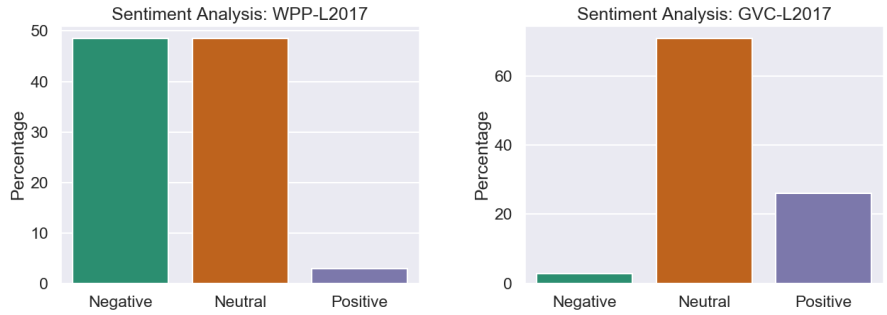


Figure 53: Correlation between Twitter and Stock Patterns

The potential correlation between Twitter's sentiments and stock patterns is illustrated above. As one of the most popular social media, Twitter is widely used by the public to show their ideas and moods, which can influence or reflect their trading behaviours. At the same time, the dynamic changes in the stock market can influence how members of the public express their thoughts on social media. In addition, some previous studies show that using Twitter to predict movements in FTSE100 can achieve over 70% accuracy, indicating an existing potential correlation between Twitter and stock patterns.

Figure 54: Two stocks from different stock patterns

The above figure shows the stock price movements for two different stocks; one is from the increasing pattern and the other one is from the decreasing pattern. The following two figures present the corresponding results of sentiment analysis.

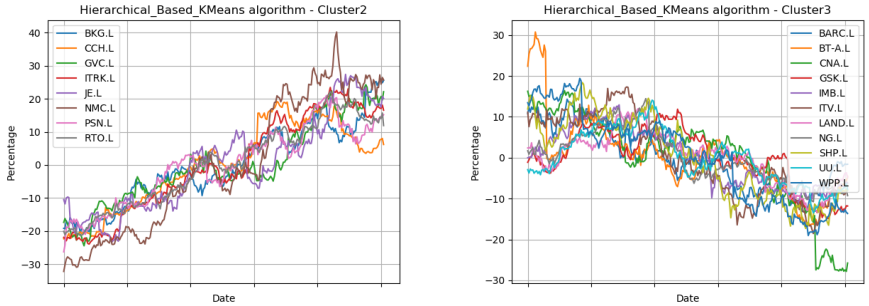


(a) Sentiment Analysis on Stock WPP

(b) Sentiment Analaysis on Stock GVC

Figure 55: Sentiment analysis of two stocks from different stock patterns

Taking the above three figures into consideration, it seems possible that with more positive tweets, the stock is more likely to show an increasing pattern, while more negative tweets means a decreasing pattern.



(a) Example for increasing pattern - cluster 2

(b) Example for decreasing pattern - cluster 3

Figure 56: Two clusters with different patterns

The above figures show two example clusters with an increasing pattern and decreasing pattern respectively. After collecting the corresponding tweets of each stock and analysing their sentiments, it is shown that 8/9 stocks have more positive tweets in cluster 2, and around a half of the stocks in cluster 3 have more negative tweets. A possible explanation for only half of stocks with decreasing patterns having more negative tweets may be because a large number of tweets are from their official Twitter accounts which rarely contain negative tweets.

9 Reflection and Conclusion

This chapter firstly focuses on the reflection of this project, which describes how the timing and the original set number of phases changed from my original plan. It also describes the knowledge gained over the past year from both technical and non-technical perspectives. Finally, a summary, which outlines the achievement of this project, is given.

9.1 Reflection

Looking back at the aims and objectives of this project, I believe that I have succeeded in accomplishing them.

During the first semester, several changes were made based on my better understanding of the objectives and the data set. It was easy for me to use Weka, so I could easily grasp the target result of this project, but in order to understand more detailed aspects of this project and polish my own programming skills, I chose Python for implementing clustering algorithms, data visualisation and etc.

The biggest challenge faced in this project was to design a novel algorithm with better performance. The first difficulty encountered during this stage was to propose a possible design and translate my ideas into code. The second difficulty was to optimise the design to achieve better results. As not many research papers focused on time series clustering, it took longer than expected. At least five versions of algorithm design were evaluated and compared with conventional clustering methods. Finally, the latest version achieved better clustering results and it was evaluated using three different methods.

During the second semester, with a better understanding of this project, more related work, such as investment strategy analysis, portfolio optimisation and sentiment analysis was done in order to explore wider aspects of this project. However, this resulted in some obstacles that made the project harder as such extended analysis required more knowledge such as finance, statistics and natural language processing.

Despite being faced with unforeseen challenges, the project was still finished in an organised and timely manner. This project has allowed me to improve both my technical and non-technical skills, which are extremely important for my future study as well as my career.

- From a technical perspective, I have gained more theoretical and practical knowledge including machine learning, natural language processing, finance and statistics.
- With regard to non-technical aspects, unforeseen problems existed all the time, especially during the process of designing the new algorithm. Because of this experience, my ability to cope with problems and setbacks has improved. In addition, this one-year long project emphasised the importance of project management, and made me better at time management.

9.2 Conclusion

In this project, the typical patterns of stock price movements are identified to help investors understand the fluctuations in share price values for a given year. Three conventional clustering algorithms (K-Means, Expectation Maximisation, HierarchicalClustering) are evaluated by internal indices (Silhouette Coefficient, Davies-Bouldin Index, Calinski-Harabaz Index) as well as via visualisation. The results showed that Hierarchical Clustering performed best among the three, followed by K-Means. However, in order to achieve better clustering results, a novel algorithm was designed, based on K-Means, and the new algorithm was evaluated in three aspects. The evaluation results prove that the newly designed algorithm achieves better results compared with the other three conventional algorithms.

In the dynamic pattern analysis, a set of stable stocks was filtered and it was shown that the company Evraz stands out among all companies in FTSE100. In order to determine promising investment strategies, four strategies were investigated and it was found that the predicted based investment performs best of the four for stock EVR.L. In addition, by implementing the Modern Portfolio Theory with suitable optimisation methods, this project provides a framework to find the best portfolio with a range of expected returns. Lastly, the sentiment analysis on Twitter shows a possible correlation between stock patterns and social media, which offers some inspiration for investigating the stock market.

To conclude, this project not only finds typical stock patterns but also provides a suitable framework for stock market analysis in the future.

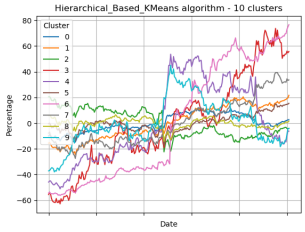
9.3 Future Work

Even though the aims and objectives of this project have been successfully realised, different aspects of this area of study can be explored and analysed further.

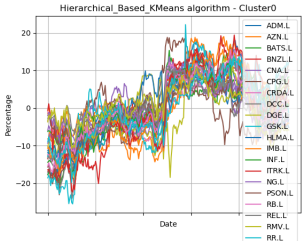
- **Time complexity for the new algorithm**: from the clustering results, the newly designed algorithm has been proved to perform better than the three conventional algorithms. However, in order to obtain better results, the improvement of this algorithm leads to an increase in the time complexity. The possible solution could be multi-threading. At each step, no two clusters influence each other, the dividing process and centroid re-calculating process only happens within the parent cluster; thus, using multiple threading to process different clusters could be a solution for addressing this issue.

- **Improvement on tweets collection**: as the previous chapter states, because of the existence of the official Twitter accounts of different companies, the results of sentiment analysis for decreasing patterns do not show strong correlation as expected. Further work might improve the quality of collected tweets by removing official tweets and attributing more weight to financial words.

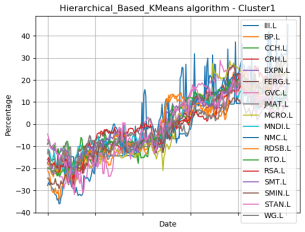
A Stock Pattern and Clusters 2016



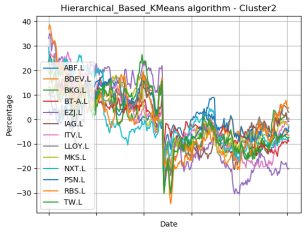
(a) 10 Patterns



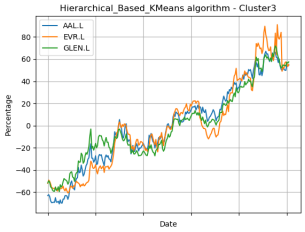
(b) Cluster 0



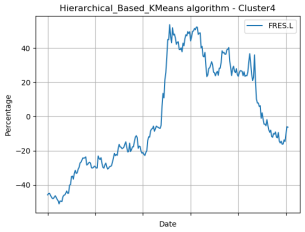
(c) Cluster 1



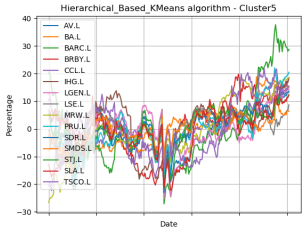
(d) Cluster 2



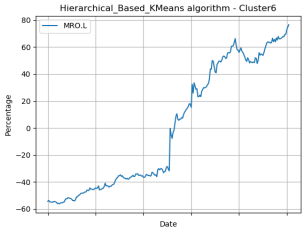
(e) Cluster 3



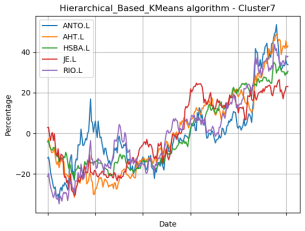
(f) Cluster 4



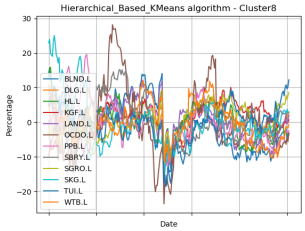
(g) Cluster 5



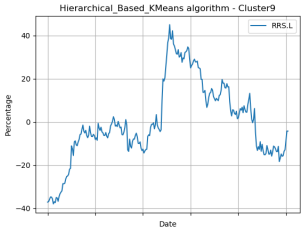
(h) Cluster 6



(i) Cluster 7



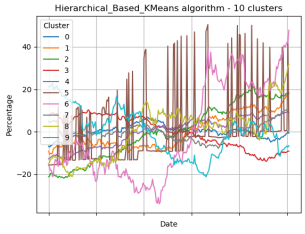
(j) Cluster 8



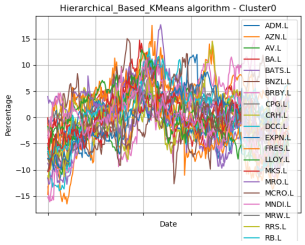
(k) Cluster 9

Figure 57: 10 Clusters For Hierarchical Based K-Means - 2016

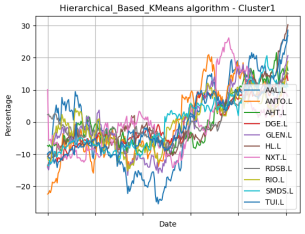
B Stock Pattern and Clusters 2017



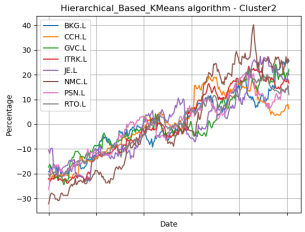
(a) 10 patterns



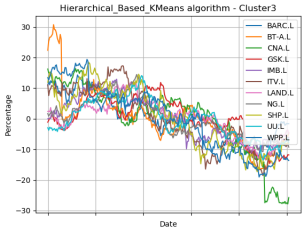
(b) Cluster 0



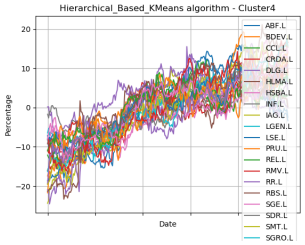
(c) Cluster 1



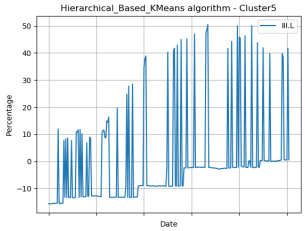
(d) Cluster 2



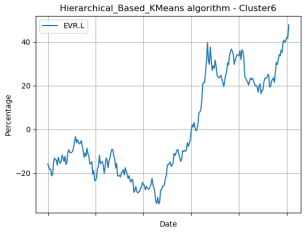
(e) Cluster 3



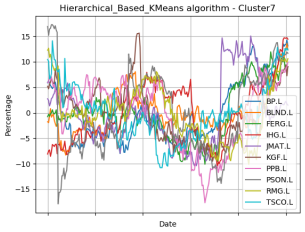
(f) Cluster 4



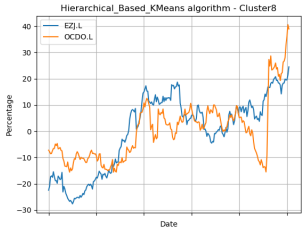
(g) Cluster 5



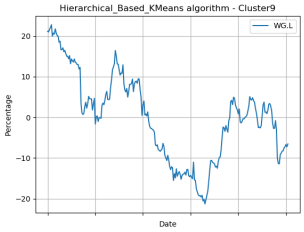
(h) Cluster 6



(i) Cluster 7



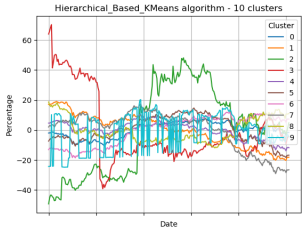
(j) Cluster 8



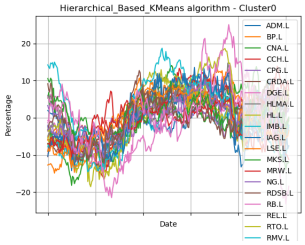
(k) Cluster 9

Figure 58: 10 Clusters For Hierarchical Based K-Means - 2017

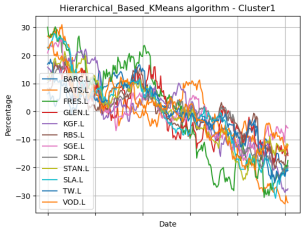
C Stock Pattern and Clusters 2018



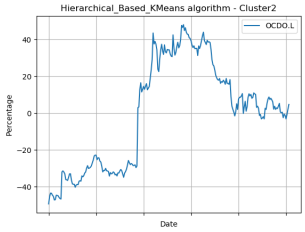
(a) 10 patterns



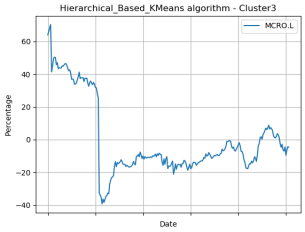
(b) Cluster 0



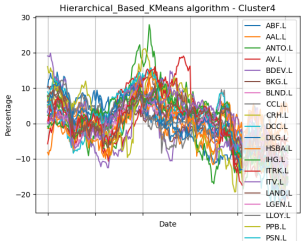
(c) Cluster 1



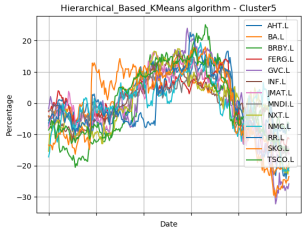
(d) Cluster 2



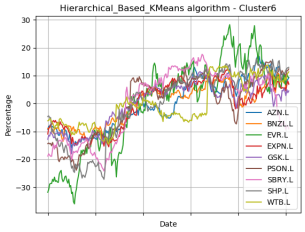
(e) Cluster 3



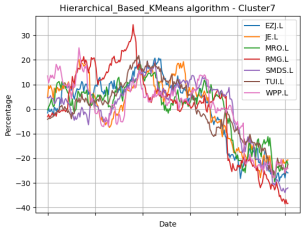
(f) Cluster 4



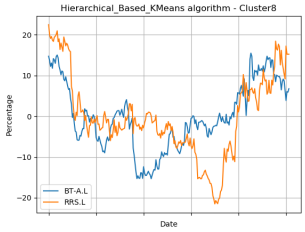
(g) Cluster 5



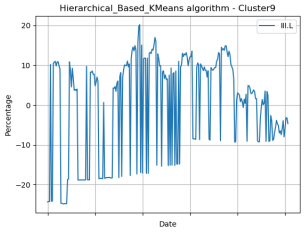
(h) Cluster 6



(i) Cluster 7



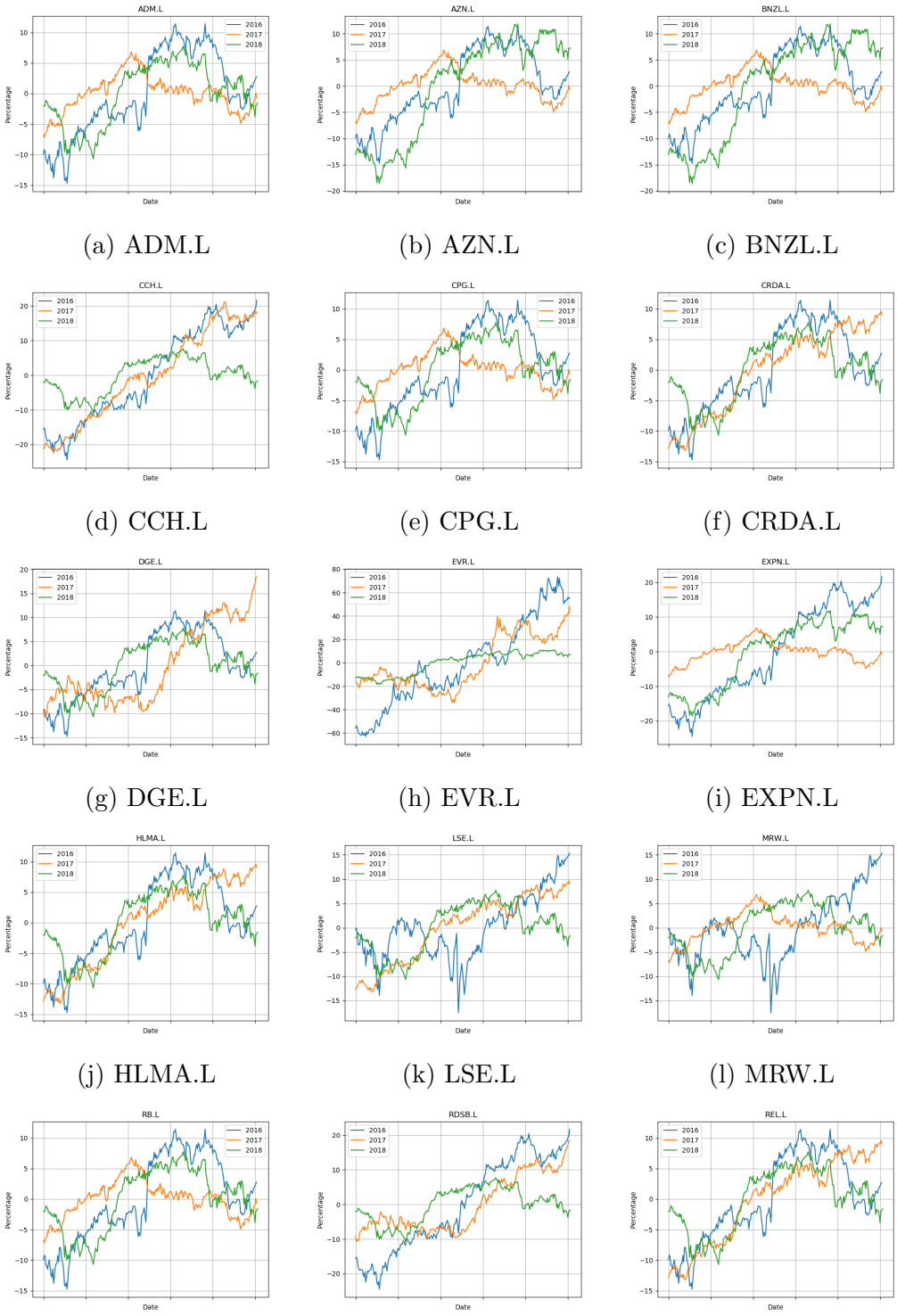
(j) Cluster 8



(k) Cluster 9

Figure 59: 10 Clusters For Hierarchical Based K-Means - 2018

D Dynamic Pattern for each stock



(a) ADM.L (b) AZN.L (c) BNZL.L

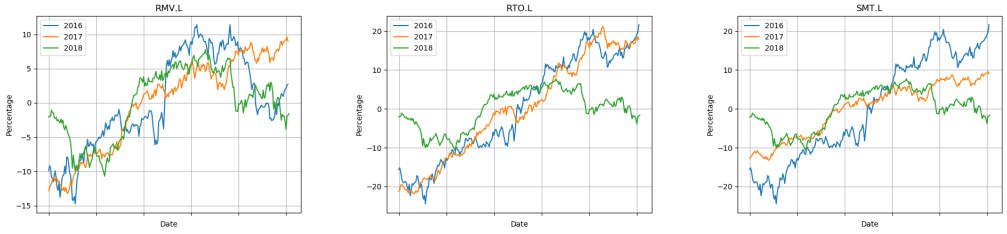
(d) CCH.L (e) CPG.L (f) CRDA.L

(g) DGE.L (h) EVR.L (i) EXPN.L

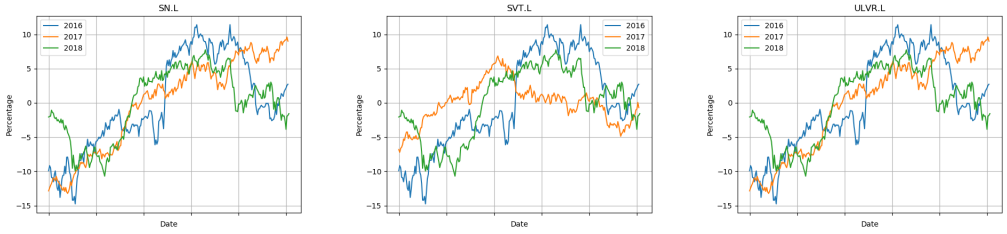
(j) HLMA.L (k) LSE.L (l) MRW.L

(m) RB.L (n) RDSB.L (o) REL.L

Figure 60: Dynamic Pattern for Three Years - 1



(a) RMV.L (b) RTO.L (c) SMT.L



(d) SN.L (e) SVT.L (f) ULVR.L

Figure 61: Dynamic Pattern for Three Years - 2

E Partial Codes for Hierarchical Based K-Means

```
def hierarchical_based_Kmeans(data_set, starting_num,target_num):
    # Step 1: Use kmeans cluster to calculate the initial cluster, and get the labels and centroids
    # it uses random seed.
    # the result would be the cluster result and centroid list.
    cluster = KMeans(n_clusters=starting_num).fit(data_set)
    cluster_result = cluster.labels_
    centroid_list = cluster.cluster_centers_
    num_clusters = starting_num

    score_list = []
    cluster_to_divide = 0

    while True:
```

Figure 62: Main function for Hierarchical Based K-Means (partial)

```
def sub_cluster_generation(data_set, cluster_num,cluster_result,centroid):
    # copy the original set and then transpose
    temp_set = data_set.copy()
    temp_set = temp_set.transpose()
    header_list = list(temp_set)
    temp_df = pd.DataFrame()

    # add each time series to the cluster where it belongs to.
    for i in range(len(header_list)):
        if cluster_result[i] == cluster_num:
            temp_df[header_list[i]] = temp_set[header_list[i]]

    # send the sub_cluster to recheck, calculate the error score and then send it back.
    error_score = sub_cluster_recheck(temp_df,centroid)
    return error_score
```

Figure 63: Function for generating sub-clusters

```
def sub_cluster_recheck(data_set, centroid):
    headers = list(data_set)
    num = len(headers)
    temp_df = data_set.copy()


    if num == 1:
        return 0
    elif num == 2:
        sum_of_squared_error = 0
        for i in range(num):
            for j in range(len(centroid)):
                # calculate the sum of squared error
                sum_of_squared_error += math.pow(centroid[j] - temp_df[headers[i]][j],2)

        return sum_of_squared_error
    else:
        # if there are more than two stocks in one cluster, compare each stock with the
        # centroid
        sum_of_squared_error = 0
```

Figure 64: Function for cluster recheck (partial)

```
def cluster_divide(data_set, cluster_num,cluster_result):
    # copy the original set and then transpose
    temp_set = data_set.copy()
    temp_set = temp_set.transpose()
    header_list = list(temp_set)
    temp_df = pd.DataFrame()

    # add each time series to the cluster where it belongs to.
    for i in range(len(header_list)):
        if cluster_result[i] == cluster_num:
            temp_df[header_list[i]] = temp_set[header_list[i]]

    cluster_result = AgglomerativeClustering(n_clusters=2, linkage = 'average').fit(temp_df.transpose())

    cluster1 = pd.DataFrame()
    cluster2 = pd.DataFrame()

    header_list = list(temp_df)
    for i in range(len(header_list)):
        if cluster_result.labels_[i] == 0:
            cluster1[header_list[i]] = temp_df[header_list[i]]
        else:
            cluster2[header_list[i]] = temp_df[header_list[i]]

    cluster1_centroid = cluster1.mean(axis=1)
    cluster2_centroid = cluster2.mean(axis=1)

    sub_centroid_list = []
    sub_centroid_list.append(cluster1_centroid)
    sub_centroid_list.append(cluster2_centroid)
    return sub_centroid_list
```

Figure 65: Function for dividing clusters

F Codes for Prices Prediction

```
# three years stock price for EVR.L, everyday Date contains the close prices for the past 20 days.
stock20 = stock[[x for x in stock.columns if 'Close Minus' in x or x == 'Close']].iloc[20:,]

stock20 = stock20.iloc[:,::-1]

train_test_split = 485

x_train = stock20[:train_test_split]
y_train = stock20['Close'].shift(-1)[:train_test_split]
x_test = stock20[train_test_split:]
y_test = stock20['Close'].shift(-1)[train_test_split:]

clf = SVR(kernel='linear')
svr_model = clf.fit(x_train, y_train)
preds = svr_model.predict(x_test)
tf = pd.DataFrame(list(zip(y_test, preds)), columns=['Next Day Close', 'Predicted Next Close'], index=y_test.index)

cdc = stock[['Close']].iloc[train_test_split:]
ndo = stock[['Open']].iloc[train_test_split:].shift(-1)
tf1 = pd.merge(tf, cdc, left_index=True, right_index=True)
tf2 = pd.merge(tf1, ndo, left_index=True, right_index=True)
tf2.columns = ['Next Day Close', 'Predicted Next Close', 'Current Day Close', 'Next Day Open']
```

Figure 66: Implementation of the Stock Prices Prediction

G Codes for Optimisation

```python
def min_func_sharpe(weights):
    return -statistics(weights)[2]

def min_func_variance(weights):
    return statistics(weights)[1] ** 2

cons = ({'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
bnds = tuple((0, 1) for x in range(stock_num))

opts = sco.minimize(min_func_sharpe, stock_num * [1./ stock_num,], method='SLSQP',bounds=bnds, constraints=cons)
print(opts['x'].round(3))

print(statistics(opts['x']).round(3))

optv = sco.minimize(min_func_variance, stock_num * [1. / stock_num,], method='SLSQP',bounds=bnds, constraints=cons)
print(optv['x'].round(3))
print(statistics(optv['x']).round(3))
```

Figure 67: Implementation of the Portfolio Optimisation

```python
# Efficient Frontier
# Evenly spaced list from 0.0 to 0.25, 30 pieces.
target_returns = np.linspace(0.05, 0.18, 30)
target_volatilities = []

# Return the volatility
def min_func_port(weights):
    return statistics(weights)[1]

for tret in target_returns:
    # Cons: equal to the return, sum equals to 1
    cons = ({'type': 'eq', 'fun': lambda x:  statistics(x)[0] - tret},
            {'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
    res = sco.minimize(min_func_port, stock_num * [1. / stock_num,], method='SLSQP',bounds=bnds, constraints=cons)
    target_volatilities.append(res['fun'])

target_volatilities = np.array(target_volatilities)
```

Figure 68: Implementation of finding Efficient Frontier

H Code for Sentiment Analysis

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA
sia = SIA()
results = []

for text in tweets_df['text']:
    pol_score = sia.polarity_scores(text)
    pol_score['tweet'] = text
    results.append(pol_score)

df = pd.DataFrame.from_records(results)


df['label'] = 0
df.loc[df['compound'] > 0.2, 'label'] = 1
df.loc[df['compound'] < -0.2, 'label'] = -1
fig, ax = plt.subplots()

counts = df.label.value_counts(normalize=True) * 100

sns.barplot(x=counts.index, y=counts, ax=ax).set_title('Sentiment Analysis: ' + stock_name)
```

Figure 69: Implementation of finding Efficient Frontier

References

[1] MarketSmith. Stock market. https://marketsmith.investors.com/stock-market/, 2019.

[2] W O'Neil. *How to Make Money in Stocks*. McGraw, Hill, 2002.

[3] SS. Teh YW Saeed, A. Ali. Time-series clustering – a decade review. https://www-sciencedirect-com.manchester.idm.oclc.org/science/article/pii/S0306437915000733#s0005, 2015.

[4] J Young. Financial times stock exchange group—ftse. https://www.investopedia.com/terms/f/ftse.asp, 2019.

[5] H. Hui-wen L Shu-Hsien, L. Hsu-hui. Mining stock category association and cluster on taiwan stock market. https://www.sciencedirect.com/science/article/pii/S0957417407002060, 2007.

[6] W T Saeed, A. Ying. Stock market co-movement assessment using a three-phase clustering method. https://umexpert.um.edu.my/file/publication/00012975_95925.pdf, 2014.

[7] B. Tiwari M K Nanda, S R. Mahanty. Clustering indian stock market data for portfolio management. https://www.sciencedirect.com/science/article/pii/S0957417410005300, 2010.

[8] D. Kuutila M Mäntylä, M. Graziotin. The evolution of sentiment analysis—a review of research topics,venues, and top cited papers. https://www.sciencedirect.com/science/article/pii/S1574013717300606, 2017.

[9] H. Zeng XJ Bollen, J. Mao. Twitter mood predicts the stock market. https://arxiv.org/abs/1010.3003, 2010.

[10] Waikato University. Weka 3: Data mining software in java. https://www.cs.waikato.ac.nz/ml/weka/, 1993.

[11] K Zucchi. How to use yahoo! finance. https://www.investopedia.com/articles/investing/091714/how-use-yahoo-finance.asp, 2014.

[12] N J Salkind. Data cleaning. http://methods.sagepub.com/reference/encyc-of-research-design/n100.xml, 2010.

[13] A Trevino. Introduction to k-means clustering. https://www.datascience.com/blog/k-means-clustering, 2016.

[14] K Chen. K-means clustering. http://syllabus.cs.manchester.ac.uk/ugt/2017/COMP24111/materials/slides/K-means.pdf, 2017.

[15] N M. Rubin D B Dempster, A P. Laird. Maximum likelihood from incomplete data via the em algorithm. http://web.mit.edu/6.435/www/Dempster77.pdf, 1977.

[16] D Kilitcioglu. Hierarchical clustering and its applications. https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6, 2018.

[17] K Chen. Hierarchical and ensemble clustering. http://syllabus.cs.manchester.ac.uk/ugt/2018/COMP24111/materials/slides/Hierarchical.pdf, 2017.

[18] R Jin. Cluster validation. http://www.cs.kent.edu/~jin/DM08/ClusterValidation.pdf, 2008.

[19] scikit-learn v0.20.3. Silhouette coefficient. https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient, 2019.

[20] scikit-learn v0.20.3. Davies-bouldin index. https://scikit-learn.org/stable/modules/clustering.html#davies-bouldin-index, 2019.

[21] scikit-learn v0.20.3. Calinski-harabaz index. https://scikit-learn.org/stable/modules/clustering.html#calinski-harabaz-index, 2019.

[22] G Milligan. Results and implications for applied analyses. https://www.researchgate.net/publication/215666104_Clustering_validation_results_and_implications_for_applied_analyses, 1996.

[23] C W Tan. Indexing and classifying gigabytes of time series under time warping. https://figshare.com/projects/Indexing_and_classifying_gigabytes_of_time_series_under_time_warping/18337, 2017.

[24] M Hargrave. Sharpe ratio definition. https://www.investopedia.com/terms/s/sharperatio.asp, 2019.

[25] B Smola, A J. Scholkopf. A tutorial on support vector regression. https://alex.smola.org/papers/2003/SmoSch03b.pdf, 2003.

[26] H Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, May 1952.

[27] W Kenton. Monte carlo simulation definition. https://www.investopedia.com/terms/m/montecarlosimulation.asp, 2019.

[28] pyOpt. Sequential least squares programming. http://www.pyopt.org/reference/optimizers.slsqp.html, 2014.

[29] Y Hilpisch. Portfolio optimization. *Python for Finance*, 11(2):322–332, Dec 2014.

[30] E Hutto, C J. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf, 2014.