1.5em 0pt

Towards understanding business via their key performance indicators: A clustering approach

Xinyu Yao

Department of Science
The University of Auckland

Supervisor: Ninh Pham

A thesis submitted in partial fulfilment of the requirements for the degree of MDataSci in Science (Data Science), The University of Auckland, 2021.

Abstract

Nowadays, data analysis is playing an increasingly important role in business and marketing. By collecting more and more data from each company, the company's owner wants to see how their company performs compared with others. In order to solve this problem, clustering has been considered as the most significant technique. In this study, we proposed several approaches for grouping companies based on their key performance indicators. We evaluated the different clustering algorithms based on several ways of preprocessing our data. So, in total, we set up three experiments. We have visualized our result by doing some data visualizations. Furthermore, we also discussed the advantages and limitations of each experiment setting as well as the capability of each clustering algorithm.

Contents

Acknowledgement

I would like to express my special thanks of gratitude to my supervisor Dr Ninh Pham, who gave me lots of suggestions and advice to guide me on this wonderful project on the topic of similarity measures and clustering algorithms, which also helped me in doing a lot of research. Secondly, I would like to thank 9Spokes Limited, who gave me the golden opportunity to do this wonderful project. Finally, I would like to thank Pietro Consoli, who is the machine learning engineer at 9Spokes Limited. In the weekly meeting with him, he shared information to help me deeper understand this project. He also provided me with several feedbacks to help me better improve my project and make that more fitted with the business perspectives. I would also like to thank my friends who helped me a lot in gathering different information, to help me discuss and review my project, gave me various ideas and fixed lots of fallacies in this project. The science department also provided space for me to finish this project. It provided a lot of study rescues as well as laboratory space. I would also like to thank my parents and family, who have given me a lot of support, not only financial but also in my life and spirit.

Chapter 1

Introduction

1.1 Background

Due to the improvement of computer computing power, faster and cheaper storage technology and the perfection of theory, data science is playing a more and more significant role to help people solve real-world problems. It has become a part of our daily lives. One aspect of their application is on the internet, such as recommender systems [1] in social media, and page-ranking algorithms [2] in web browsers. The other aspect of the application is in the business market, such as financial analysis, data visualizations, business intelligence and modelling. One example is when an organization is trying to identify the optimal conditions to maximize profits and minimize expenses, which can help deliver the right products to the customers at the right time. It can also help companies develop new products to meet their customers' needs. Several tools have been designed for doing data analysis, such as R Studio, Tableau, KNIME and Python. However, in this thesis, we are only using Python and R Studio.

1.1.1 The 9Spokes Limited

The 9Spokes is a powerful business ecosystem on a global scale. It offers modern businesses a management app that brings meaningful data together across business, its applications, and its bank. Think of 9Spokes as a virtual advisor, to guide the business in order to grow and thrive.

Powered by bank and business data, 9Spokes delivers meaningful, personalized, and shareable insights to businesses to help inform their next move and steer them towards their goals. It's a collaborative resource that facilitates holistic conversations between businesses and their banks that go beyond just the financials, helping to improve visibility and reduce risk. Businesses gain a value-added business hub, while their banks get the insights needed to offer products and services tailored to their customers' needs.

1.1.2 Key Performance Indicators (KPIs)

Key performance indicators (KPIs) is a set of performance measurements used to evaluate the company's performance in a period of time [3]. It can help business owners to better understand their company's situation, determine their company's strategic, financial, and operational achievements, especially compared to those of other businesses within the same sector. KPIs can be financial, including net profit (or the bottom line, gross profit margin), revenues minus certain expenses, or the current ratio (liquidity and cash availability).

There are many types of KPIs (Revenue, Expenses, Profit, Profit Margin, P/E Ratio, Sales, Revenue Growth Rate, Number of Customers, ...). In this thesis, the most common and most important part we used are the Revenue, Expenses, Profit and Profit margin. They are listing as below:

- Revenue: The total amount of income generated by the sale of goods or services related to the company's primary operations, also known as gross sales. It should be equal to Expenses - Profit.
- Expenses: The costs you incur in the day-to-day running of your business. (Expenses = Revenue - Profit)
- Profit: Profit describes the financial benefit realized when revenue generated from a business activity exceeds the expenses, costs, and taxes involved in sustaining the activity in question. (Profit = Revenue - Expenses)
- Profit margin: The percentage of profit to revenue. (Profit margin = Profit / Revenue)

We can notice, there exist correlations between some pairs of the KPIs. For example, the profit should equal to Revenue subtract by Expenses.

1.1.3 The Australian and New Zealand Standard Industrial Classification (ANZSIC)

In New Zealand and Australia, when a company established, the government will assign its industry type according to its predominant activities. The Australian and New Zealand Standard Industrial Classification (ANZSIC) has designed for this. It used in both countries for the production and analysis of industry statistics. It replaces the Australian Standard Industrial Classification (ASIC) and the New Zealand Standard Industrial Classification (NZSIC) that have been in use for many years. The Australian and New Zealand Standard Industrial Classification (ANZSIC) has been jointly developed by the Australian Bureau of Statistics (ABS) and Statistics New Zealand (Statistics NZ).

The ANZSIC has four levels of hierarchical classification, namely Divisions (the broadest level), Subdivisions, Groups and Classes (the finest level). At the Divisional level, the business was divided by industry type. Group and Class levels provide more detailed dissections of these

categories. (see Table 1.1) In our experiment, we will only use "Division" as the classification of each company's industry type.

Division	C	Manufacturing
Subdivision	11	Food Product Manufacturing
Group	111	Meat and Meat Product Manufacturing
Class	1111	Meat Processing

Table 1.1: An example of ANZSIC code

1.2 Data Set

The original data set is provided by 9 Spokes Limited, which is collected from customers' connected applications and stored in their database. The type of this data sets is comma-separated value (CSV) file. It contains revenue data for companies in different regions. The size for this data set is 12.5 megabytes (MB), containing 78201 records and 11 variables.

The **user** is the hashed value of the customer name with the **connection** between the customer and the 9Spokes Ltd. The **osp** is their online service provider. The **currency** is the currency used by companies with their **date recorded**, **account name** (one user may have several accounts) and **account types**. The **revenue to date** is the cumulative revenue since the beginning of the financial year. The **daily revenue** is what has cumulated in its original currency. The **NZD** is the daily revenue converted in NZD currency. (See Figure 1.1)

user	connection	osp	currency	balance_date	account_name	account_type	revenue_to_date	daily_revenue	company	NZD
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	30/11/2018	Sales	sales	50000	50000	12c1a125-d07d-4ef9-b0e7-4d034802d387	47007.12733
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	30/06/2019	Sales	sales	52000	2000	12c1a125-d07d-4ef9-b0e7-4d034802d387	1915.566038
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	30/09/2019	Sales	sales	500	500	12c1a125-d07d-4ef9-b0e7-4d034802d387	464.057554
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	31/10/2019	Sales	sales	3850	3350	12c1a125-d07d-4ef9-b0e7-4d034802d387	3114.174204
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	30/11/2019	Sales	sales	16548	12698	12c1a125-d07d-4ef9-b0e7-4d034802d387	12057.52744
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	31/12/2019	Sales	sales	18180.48	1632.48	12c1a125-d07d-4ef9-b0e7-4d034802d387	1567.976797
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	31/01/2020	Sales	sales	20778.48	2598	12c1a125-d07d-4ef9-b0e7-4d034802d387	2508.424282
0037b938-8c33-4c26-92ea-653e699dc6f9	3549f165-2a07-4005-925b-6ee774d7821d	xero	AUD	25/03/2020	Sales	sales	37796.35	17017.87	12c1a125-d07d-4ef9-b0e7-4d034802d387	16570.2979
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/10/2017	Sales	sales	19430	19430	1b043141-b305-470d-b810-3899b64f036d	19430
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	30/11/2017	Sales	sales	19738	308	1b043141-b305-470d-b810-3899b64f036d	308
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/12/2017	Sales	sales	20266	528	1b043141-b305-470d-b810-3899b64f036d	528
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/01/2018	Sales	sales	23616	3350	1b043141-b305-470d-b810-3899b64f036d	3350
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	28/02/2018	Sales	sales	25356	1740	1b043141-b305-470d-b810-3899b64f036d	1740
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/03/2018	Sales	sales	27780	2424	1b043141-b305-470d-b810-3899b64f036d	2424
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	30/04/2018	Sales	sales	475.2	475.2	1b043141-b305-470d-b810-3899b64f036d	475.2
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/05/2018	Sales	sales	3312.2	2837	1b043141-b305-470d-b810-3899b64f036d	2837
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	30/06/2018	Sales	sales	6832.46	3520.26	1b043141-b305-470d-b810-3899b64f036d	3520.26
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/07/2018	Sales	sales	9612.15	2779.69	1b043141-b305-470d-b810-3899b64f036d	2779.69
00415952-3309-4e6b-af4b-b1c02dc2d08b	0928841b-173f-4064-9974-651fd32d8e2b	xero	NZD	31/08/2018	Sales	sales	14769.84	5157.69	1b043141-b305-470d-b810-3899b64f036d	5157.69

Figure 1.1: Figure showing the original data set.

The second data set also shared by 9Spokes Limited. It is a CSV file as well, containing the customer name and their industry type. The size for this data set is 5 MB, with 80295 records and only two variables **user** and **industry** type. The **user** in the second data set is the foreign-key reference to the "user" at the original data set. The **industry** is a text which briefly introduced their industry type.(See Figure 1.2)

	user	industry
1		
2	7cff027a-1655-4e08-be3e-0e4dbc1c4009	HomeandGarden
3	5de776c7-b578-4e1b-b7ad-51735a493aa8	HomeandGarden
4	486e88f9-a0ae-40fc-a33a-950ba9d2674c	HomeandGarden
5	30728058-8fbf-49d4-b776-4bcac0e2688a	HomeandGarden
6	63e4e234-9e44-46dc-8712-9ef119e8ce91	HomeandGarden
7	d225e75f-c3ce-4126-8aaf-1452f13f2e02	IndustrialandManufacturing
8	d70b94da-307c-48ad-b546-4f785789e146	IndustrialandManufacturing
9	a0469103-ce82-4554-9943-137c11e9af24	IndustrialandManufacturing
10	cfcef80c-611e-4ad2-a254-06c534419a06	IndustrialandManufacturing
11	f1c44b03-18b1-4e70-8ff6-b3ca1b98c741	IndustrialandManufacturing
12	6cb68212-d705-4475-8183-9d19e0827f76	IndustrialandManufacturing
13	fdd3b018-c1fb-407e-82b7-797971dd0382	IndustrialandManufacturing
14	1ec1a42a-786c-4d7e-90b5-443e11c44893	IndustrialandManufacturing
15	eab16c5b-0b15-4eb4-a51f-ba2607e6ee2a	IndustrialandManufacturing
16	5d1544a1-1186-4905-84d4-e1f3275366666	IndustrialandManufacturing
17	5cea11ef-bdcc-485f-b073-7b28c8679220	IndustrialandManufacturing
18	b153a1bd-a60b-4805-b56c-5a98b6d88241	IndustrialandManufacturing
19	727681da-b5f9-4a6b-8db1-4708d6a524c8	IndustrialandManufacturing
20	bd795eaf-bb45-42b9-b264-cad22aca0c91	IndustrialandManufacturing

Figure 1.2: Figure showing the second data set.

1.3 Problem Formulation

As part of the business benchmarking, the current dashboard for small and medium-sized enterprises (SME) provides a good overview of how their business performing based on data from the connected applications. Other research shows that SME would like to see how they are performing against similar businesses. To be able to benchmark, they want to establish a function that makes comparable businesses for each metrics. More generally, 9Spokes Ltd wants to investigate some suitable approaches that allow their customers to better understand how their company performance compared with others. The 9Spokes platform provides a large amount of data about SME that used to build company profiles. They expected to see that the project can produce an ideal company profile and matrix that matches the company profile to each benchmark area.

Some other researchers proposed some techniques that may be helpful for solve this kind of questions. In this paper [4], the author has proposed a clustering approach for financial data analysis. The author implemented K-means [5] clustering as the partitioning method, DBSCAN [5] as a density-based clustering and hierarchical agglomerative [5] clustering for a time-based data stream. In this paper [6], the author used document frequency (DF), term frequency (TF), and inverse document frequency (IDF) [7] as the similarity measure to cluster the text format E-business contents. Other papers [8, 9] also showed how to use different similarity measures (i.e. Euclidean distance, dynamic time warping, Pearson correlation, ...) to perform clustering based on the input type.

1.4 Thesis Structure

The purpose of this thesis is to investigate suitable approaches that may use to group companies based on their key performance indicators. Our problem statement can be divided into two aspects. The first is the choice of the similarity measure functions to compute the simi-

larity matrices. The second is the choice of the clustering algorithm based on our similarity matrix. Therefore, the first section of this thesis presents a brief overall literature review on potential similarity measure functions and clustering algorithms that are potentially worthwhile candidates to be employed in the context of our clustering problem. More specifically, the similarity measure functions discussed in the next section include Euclidean distance, dynamic time warping, and Gower's distance. The clustering algorithms discussed include some representative techniques for both partitional and hierarchical clustering: K-Means, partitioning around medoids (PAM), density-based spatial clustering of applications (DBSCAN), agglomerative hierarchical clustering, and mean-shift clustering. We noticed that there were correlations between some pairs of KPIs, so we also introduce the correlation matrix.

First, we introduced our data sets provided by 9 Spokes Limited. To better understand them, we also explained some Key Performance Indicators that were included in our data set. There are many types of industry running in our world, such as mining, manufacturing and construction. In order to distinguish them, we also introduced the ANZSIC code, which is a standard in Australia and New Zealand also referred to as business industry codes.

Second, we introduced all methodologies we used. We divided them into three parts, correlation and regression, similarity measure, and clustering algorithms. In the first part, we introduced some statistical tools to evaluate the relationship between two variables, such as Pearson correlation and hypothesis tests. We use a simple linear regression model to describe them. In the second part, we introduced three similarity measures. They are corresponding to three different input data types. In the third part, based on these similarity measures, we introduced clustering algorithms and shown the time and space complexity for them.

In the following section, we discuss how to set up our experiment, including how to perform data pre-processing, which similarity function we used and how to apply different clustering algorithms based on these similarity functions. The difference similarity measure function may require a different type of input, so we categorize them individually. For each experiment, based on different clustering algorithms, we show our results and further discussed their limitation and precautions. Our result has shown if we used the same similarity measure, the results will be the same whatever we used any clustering algorithm. But the results have shown a significant difference if we used the different similarity measures. The result will only become less interpretable if we used Glower's distance as the similarity measure. We can generate their label for each company based on their clustering result and merge it back to our original data set. Although all the approaches are unsupervised learning, we either have the validation set or the true label, we can also estimate their label by doing some data visualizations.

As part of our conclusions, we have found two approaches that successfully grouped companies based on their KPIs. By providing a company name with its account name, we can get its industry division and estimate its performance compare with other companies in the same industry division. There are still some drawbacks and bottlenecks for our approaches and the limitation on this data set. We hope those can get solved in future works.

Chapter 2

Methodologies and Analysis

2.1 Correlation and Linear Regression

The result of the business revenue shows some correlation between some business KPIs. For example, there is always a correlation between the number of customers and revenue. In this section, we will introduce statistical concepts of correlation and regression, which will help us to evaluate the relationship between different KPIs.

2.1.1 Pearson Correlation

In machine learning, correlation is an important method to measure and interpret the strength of the linear or nonlinear relation between multiple data variables. In our case study, we will focus on the Pearson correlation [10, 11] $r \in [-1, 1]$. Ranking between completely positive correlation $r = 1$ to completely negative correlation $r = -1$, and $r = 0$ means uncorrelated. The direction and strength of the correlation have shown in Figure 2.1. In our case study, before we ran any clustering algorithms we need to compute the Pearson correlation between each pair of the variables in our data set. We can remove one of those if two variables show either a strong positive relationship or a strong negative relationship. Here is the formula showing how to compute the Pearson correlation coefficient:

$$r_{X,Y} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

The $cov(X, Y)$ is the covariance between X and Y. Its formula is as follows:

$$Cov(X, Y) = E[(X - \mu_x)(Y - \mu_y)], \tag{2.2}$$

where μ_x and μ_y are the mean of random variables X and Y respectively.
The σ_X and σ_Y are standard deviations of X and Y. Its formula is as follows:

$$\sigma_x = \sqrt{E[(X - \mu_x)^2]}, \sigma_y = \sqrt{E[(X - \mu_y)^2]} \tag{2.3}$$

Therefore, the Pearson correlation can be rewritten as follows:

$$r_{X,Y} = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sqrt{E[(X - \mu_x)^2]}\sqrt{E[(X - \mu_y)^2]}}, \tag{2.4}$$

where X and Y are the different variables (They are both including the same number of the observations), μ_x and μ_y are the means of observations in X and Y. From the above equation, it is not hard to see if variable X is positively correlated with variable Y, then the $X_i - \mu_x$ and $Y_i - \mu_y$ are more inclined to have the same symbols (both positive). So, the expected value of $(X - \mu_x)\ (X - \mu_y)$ will always be positive. The more positive correlation between X and Y, the larger expected value of $(X - \mu_x)\ (X - \mu_y)$. On the other hand, if the variable X is negatively correlated with variable Y, then the expected value of $(X - \mu_x)\ (X - \mu_y)$ will always be negative. The more negative correlation between X and Y, the smaller expected value of $(X - \mu_x)\ (X - \mu_y)$. If it is uncorrelated between X and Y, then the $E[(X - \mu_i)(Y - \mu_y)]$ will be close to 0. The closer to 0 the less correlation between X and Y. By knowing the covariance between X and Y, we can further divide it by the standard deviations of X and Y to keep the correlation in the range of 1 to -1.

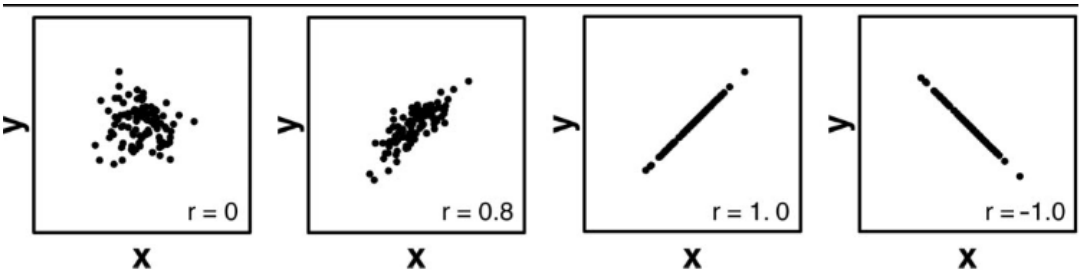


Figure 2.1: Scatter plots of four set data set showing the Pearson correlation at $r = 0$, $r = 0.8$, $r = 1.0$, and $r = -1.0$.

2.1.2 Simple Linear Regression

The purpose of simple regression analysis is to evaluate the relationship between two variables, by given one variable as the input, we can predict the outcome for the other variable. There are many types of the regression model (e.g., Logistic Regression [12], Linear Regression, Lasso Regression [13] and so on). In this thesis, we are going to exploit the simple linear model [14] with one continuous variable on another continuous variable with no gaps on each measurement scale.

A simple linear model always contains two variables: The independent variables, i.e. X, and dependent variables, i.e. Y. The dependent variable Y is linear with respect to both the

regression parameters and the independent variable X. This model can be written as follows:

$$Y_i = aX_i + b + \varepsilon_i, \tag{2.5}$$

where the linear regression parameter a is the coefficient (slope), and the linear regression parameter b is the intercept (on y-axis). The parameter ε is a random error term following the distribution $N(0, 1)$ (see Figure 2.2). Analyses always assume the errors are independent and identically distributed (iid) for easy inference and improved efficiency [15].

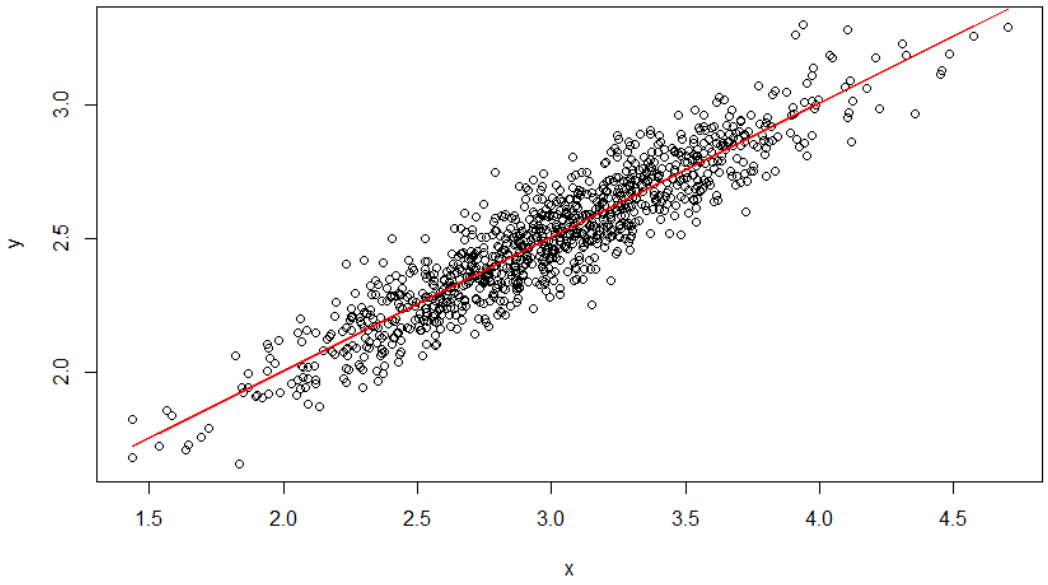


Figure 2.2: A example of simple linear regression at simulation data (for a = 0.5 and b = 1).

By providing two variables there are several steps to fit a regression model [10].

- Determine whether there is a relation between variables X and Y or not (e.g. plot or correlation matrix).
- Obtain the most suitable model that best fits our data.

- Evaluate the model to determine the confidence of the relationship for prediction and estimation.
- Fit the original data back to model, to see whether the outcomes meet these criteria.

2.1.3 Statistical Hypothesis Tests

When we fit our model, we try to evaluate the relationship between variables X and Y, but how much evidence we have to say that they are related? To evaluate them, we need to use hypothesis tests. The null hypothesis h_0 will be $a = 0$ (there is no relation between variables X and Y) and the hypothesis h_1 will be $a \neq 0$ (there is a relation between variables A and B). Then, we can use the mean of a t-statistic to compute the significance of the intercept (b) and the coefficient (a). Here is the equation of how to compute the t-score [16]:

$$t = \frac{\overline{x} - h_0}{\frac{s}{\sqrt{n}}} \tag{2.6}$$

Where $\overline{x}$ is the sample mean, h_0 is the hypothesized population mean, s is the sample standard deviation and n is the sample size. By knowing the t-score, we can further map to the p-value $p(t - score|h_0)$. If the p-value is smaller or equal than 0.05, then we have strong evidence to against our null hypothesis in favour our hypothesis, that is — there is a relation between X and Y [16].

2.1.4 Limitations and Precautions

Although we found there is a correlation between X and Y, but is not a sufficient proof of causation between X and Y. For example, the investigate [17] found that the lung cancer is correlated with the tobacco smoke, but we can not say that tobacco smoke causes the lung cancer. The real world is always complex, one variable Y may cause by other variables X that also involved the third variable Z. In statistics, X is called confounder and it both affect variable Y and Z. For example, the number of people drowned at the beach is correlating with the ice cream sales. People may make the wrong conclusion that ice cream sales cause people drowning, however there is a third variable called temperature that both affect the ice cream sales and the number of drowning, that people always forget to consider.

At first, simple linear model looks very simple and straightforward however, it has many limitations and precautions. Some of these have been listed below. [16].

- The simple linear model is only restricted to continuous variables.
- The real-world is always complicated, some of the observations may not suitable for the linear model.

- No matter how strong the evidence is, it should not be interpreted as X leads to Y.
- The regression model should not be used to predict the estimated value ($\hat{Y}$) outside of the range of observations variables.

In our work, we exploit linear regression to extract more relevant features for our clustering algorithm.

2.2 Similarity Search

When we assign points to the closet centroid we need to measure how "close" it is by using the similarity measure. There are many types of similarity measures based on given types of data. The Euclidean distance are widely used for computing the similarity between two points in the Euclidean space. The advantage of using Euclidean distance is that it can suit for data in higher dimensional and easy to be implemented. However, Euclidean distance can not work for all types of data. For example, the Jaccard similarity [18] or cosine similarity [19] measures are always used in the text documents. The Gower's distance [20] can even satisfy the data that contains a combination of logical, categorical, numerical or text data. The dynamic time warping has been widely used to handle the time series data. In our case study, based on our goal and the data types, we will use the Euclidean distance, Gower's distance, and dynamic time warping as the distance measures.

2.2.1 Euclidean Distance

Euclidean distance is used for computing the difference between two points in the Euclidean space. We briefly describe its notion and show how it works for the similarity search problem. Let us assume our data can be represented in the d-dimension Euclidean space R^d, we want to classify similarity items in R^d, the Brute force way is to compute Euclidean distance for each pair of items in this space, sort them and get the result. In mathematics, Given a set of training examples with items $(x_1, y_1), ..., (x_d, y_d)$ in d-dimensional space. The Euclidean distance's between X and Y is given by [21]:

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_d - y_d)^2} \tag{2.7}$$

$$= \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{2.8}$$

Where $x \in R^d$, $y \in R^d$.

It is not difficult to see that it takes $O(dn^2)$ time complexity to calculate the distance of all item pairs in the data set.

2.2.2 Dynamic Time Warping

We have introduced the Euclidean distance, its advantages and limitations. In this subsection, we present the dynamic time warping (DTW) distance is a popular technique to find an optimal alignment between two given time-dependent sequences (time series). The time series can be generated everywhere, for example, the stock price over a period of time, business transactions, and the weather records. Figure 2.3 is showing the comparison between dynamic time warping and euclidean matching on two time series. We can see that some traditional similarity measurement like Euclidean distance may not satisfy all time series types, because they are "parallel" computing the similarity between two time series (Figure 2.3 A). Dynamic time warping is the technique that has been used in fields such as data mining and information retrieval. By finding similar patterns between two time series, it can achieve better performance for classification and clustering tasks (Figure 2.3 B).

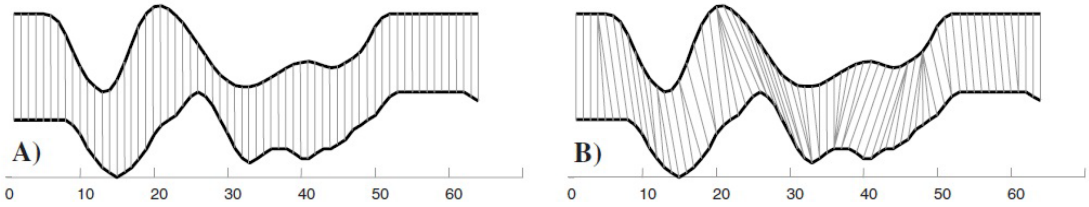


Figure 2.3: A example of Euclidean matching and Dynamic Time Warping matching [22].

We are now introducing the basic concept of the dynamic time warping. We are trying to find the common patterns between different time series. To achieve this, we are aligning two time sequences $X = (x_1, x_2, ..., x_N)$ of length $N \in \mathbb{N}$ and $Y = (y_1, y_2, ..., y_M)$ of length $M \in \mathbb{N}$. The feature space denoted by F, then $x_n, y_m \in F$ for $n \in [1 : n]$, $m \in [1 : m]$. To compare two features x_n and y_m between two sequence X and Y, we need a local cost measure function [23] defined as follows:

$$c : F \times F \to \mathbb{R}_{\geq 0}, \tag{2.9}$$

where c is called local cost measure, and $c(x_n, y_m)$ will be small (low cost) if x_n and y_m are similar with each other otherwise $c(x_n, y_m)$ will be large if they are less similar. By computing the local cost for each pair of elements in the sequence X and Y, we can form the cost matrix $c \in \mathbb{R}^{N \times M}$ (where $c(n, m) = c(x_n, y_m)$). Figure 2.4 shows an example of the cost matrix between sequences X and Y. We want to find an alignment between X and Y that has the minimal overall cost [23].

An alignment can represent by an (N, M) warping path $p = (p_1, p_2, ..., p_L)$ with $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$. It will comply with the following three constraints [23]:

- Boundary Conditions: $p_1 = (1, 1)$ and $p_L = (N, M)$.
- Monotonicity Conditions: $n_1 \leq n_2 \leq ... \leq n_L$ and $m_1 \leq m_2 \leq ... \leq m_L$.
- Step size condition: $p_{l+1} - p_l \in (1, 0), (0, 1), (1, 1)$ for $l \in [1 : L - 1]$.

For example, the alignment between two sequences $X = x_1, x_2, ..., x_N$ and $Y = y_1, y_2, ..., y_M$ can be define as an (N, M) warping path $p = (p_1, ..., p_L)$ by assigning the element x_{nl} of X to the element y_{ml} of Y [23]. In this example, the boundary conditions ensure that the first and last elements of sequences X and Y are aligned. The monotonicity condition ensures that the sequences must be ordered by time, if the element in X precedes a second one it must hold in Y. Finally, the step size condition ensures we must consider the elements in X and Y one by one, no element can be omitted or repeated.

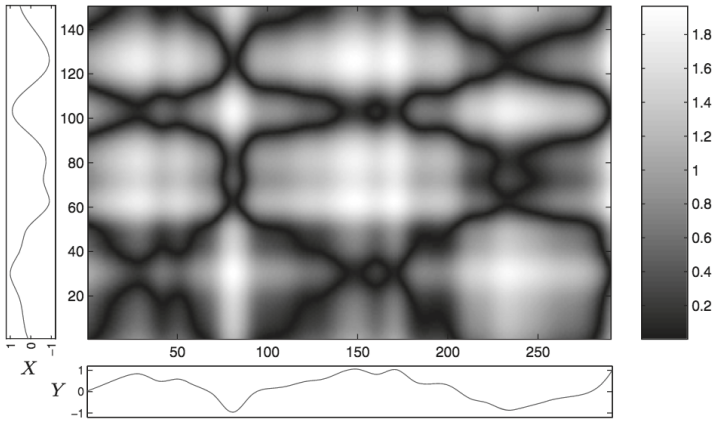


Figure 2.4: Figure shows the cost between two time sequences X and Y using the Euclidean distance as the local cost measure c. The darker the place represents the higher of the cost [23].

Once we define the total cost $c_p(X, Y)$, we can use that to formally define for each warping path p between time sequences X and Y.

$$c_p(X, Y) = \sum_{l=1}^{L} c(x_{nl}, y_{ml}). \tag{2.10}$$

Because we want to find a warping path between X and Y, which is having the minimal overall cost c compared with all possible warping paths. Then, we can define the dynamic time warping DTW between X and Y as the minimal cost of c_p (called c_p*):

$$DTW(X,Y) = min\{c_p(X,Y) \mid p\ is\ an\ (N,M)\ warping\ path\} = c_p^*(X,Y) \tag{2.11}$$

Searching over all the possible warping paths between X and Y to determine the minimal warping path c_p* is very time inefficient. Hence, the author [23] introduces an $O(NM)$ algorithm based on dynamic programming. We define the prefix sequences $X(1:n) = (x_1, ..., x_n)$ for $n \in [1:N]$ and $Y(1:m) = (y_1, ..., y_m)$ for $m \in [1:M]$. We can set

$$D(n,m) = DTW(X(1:n), Y(1:m)), \tag{2.12}$$

where the $D(N,M)$ shows an $N \times M$ matrix, which is called as the accumulated cost matrix D. We have $D(N,M) = DTW(X,Y)$, and a tuple (n,m) will represent a matrix entry of the accumulated cost matrix D.

By knowing the above definitions and restrictions, we can dynamicly compute our accumulated cost matrix D efficiently [23].

$$D(n,m) = c(x_n, y_m) + min[D(n-1,m), D(n,m-1), D(n-1,m-1)] \tag{2.13}$$

That is, for each element at the sequence X and the sequence Y. We are "parallel" matching $(i-1, j-1)$ or matching with the previous one $(i-1, j)$ and following one $(i, j-1)$. Their cumulative distances will take the minimum one. Applying that function to all of the elements in the warping window we can get the minimum cumulative distance. So, we can say we found the minimal cost warping path between X and Y, or in order words, we found their common patterns.

Figure 2.5 shows the minimal cost warping path (as the white line) for Figure 2.4. Figure 2.5 (a) shows the result by drawing the low costs entry (n,m) on Figure 2.5. The2.5 (b) shows the minimal cost warping path by using the accumulated cost matrix D.

2.2.3 Gower Distance

Gower proposed Gower's distance [20] back in 1971. It was an improvement based on the traditional similarity measure. For a multi-dimensional data set, it is simple to compute the pair-wise distance for numeric columns, but what if there exist some categorical columns? The Gower distance was one of the solutions under this situation. It has extended to the categorical column and allowed us to compute the distance measure on mixed numeric and categorical features.

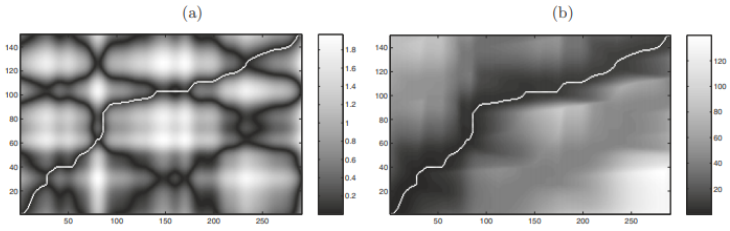


Figure 2.5: (a) the minimal cost warping path (as the white line) for Figure 2.4. (b) the minimal cost warping path (as the white line) by using the accumulated cost matrix D [23].

Let a data point X be in q dimensional space, and $X = \{x_1, ..., x_p, x_{p+1}, ..., x_q\}$ be the mixture of variables. There are p categorical variables and $q - p$ continuous variables. Thus, we can rewrite the vector X as follows [24]:

$$X = (z_1, ..., z_p, c_1, ..., c_{q-p}) = (z, c), \tag{2.14}$$

where z represents the subset of X containing the p categorical variables and c represents the subset of X containing the q–p continuous variables. Let x_i and x_j be two difference points with $x_i = \{z_i, c_i\}$ and $x_i = \{z_j, c_j\}$. We provide the weight for each categorical variables z and continuous variables c. We can form the equation for calculating the Gower's dissimilarity coefficient [25] as follows:

$$D_{X_i,X_j} = \frac{\sum_{r=1}^{p} w_{ijz_r} D_{ijz_r}}{\sum_{r=1}^{p} w_{ijz_r}} + \frac{\sum_{r=1}^{q-p} w_{ijc_r} D_{ijc_r}}{\sum_{r=1}^{q-p} w_{ijc_r}}, \tag{2.15}$$

where w_{ijz_r} and w_{ijc_r} respectively are the weight of the p categorical variables in dimension z_r and the $q - p$ continuous variables in current dimension c_r. The D_{ijz_r} will be the distance measure for categorical variable z_r between dimension i and j, that can be obtained as follows [24]:

$$D_{ijz_r} = \begin{cases} 1 & otherwise, \\ 0 & z_r^i = z_r^j. \end{cases} \tag{2.16}$$

The $D_{x_i x_j c_r}$ will be the distance measure for continuous variables c_r between dimension i and j, which is the Manhattan distance (i.e., L1 norm) shows as follows [24]:

$$D_{ijc_r} = \frac{\left| c_r^i - c_r^j \right|}{max(c_r) - min(c_r)}. \tag{2.17}$$

Although the Manhattan distance is the default matrix to computing the Gower's dissimilarity measure, we can also use other distance measure matrix instead (i.e., Euclidean distance).

It is not hard to see by tuning the weight w_{ijz_r} and $w_{x_i x_j c_r}$, we can control our cluster algorithm to "pay more attention" on some special variables and "ignore" at some special variables. However, in the real world, choosing the most suitable weights is complicated and highly-dependent on the application.

2.3 Clustering Algorithms

Clustering refers to the analysis process of grouping a collection of physical or abstract objects into multiple classes composed of similar objects [5]. It has been widely used in the area of data analysis. In common words, clustering is the tool of grouping objects that are similar to each other. There are mainly two categories of clustering — hierarchical and partitional. In partitional clustering we divide objects into non-overlapping groups, but in hierarchical clustering, we nest objects as a hierarchical tree. In this thesis, we introduce several clustering techniques for solving the clustering problem.

2.3.1 K-means

The K-means algorithm [26] was first proposed by Stuart Lloyd of Bell Labs in 1957. It is known for its simplicity and efficiency. It also has many variants such as Mini-batch k-means [27] and k-means++ [28]. The "K-mean" is represented for the K centroids, which are usually the mean of points in the same cluster [5]. It is a partitioning-based clustering algorithm using distance measures, it can also be applied to high-dimensional data sets. We will introduce the basic K-mean algorithm in this section.

Firstly, we need to choose K initial centroids that are corresponding to the K clusters. Secondly, for each observation in our data set, we compute the distance (similarity measuring) to each centroid and assign it to the closest one. Then, we update the centroid for each cluster and recompute the distances. We do that repeatedly until the centroids no longer change. Algorithm 1 shows the pseudocode of the K-means algorithm [5].

Algorithm 1 Basic K-means algorithm

1: Randomly select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute centroid for each cluster.
5: **until** Centroids do not change.

2.3.1.1 Choosing Initial Centroids

Finding the most suitable initial centroids can significantly reduce the number of iterations and the running time. In the worst case, selecting poor initial centroids could lead to being stuck into the local minimum. This would lead to a poor clustering result. In the traditional K-means algorithm, the initial centroids are selected randomly, but we can not guarantee a good clustering result [5].

There are several ways to overcome this problem. We can run the k-means algorithm multiple times to select the one with the best performance, or do pre-processing or post-processing to select better initial centroids. The K-means ++ method was proposed recently by David Arthur and Sergei Vassilvitskii 2007 [5], with better initialization of the centroids for the K-means algorithm. The basic-idea for the K-means++ is, we are trying to make the distance between each initial cluster centroids as far as possible. The detail of the K-mean++ method is shown in Algorithm 2.

Algorithm 2 K-means++ initialization algorithm

1: Randomly select a point as initial centroids.
2: **for** each observation i **do**
3: Compute the distance $d(x_i)$ to its closet centroid.
4: Assign it a probability proportion to it's $d(x_i)^2$
5: Based on the probability, select a new point as a new centroid.
6: **end for**

Form Algorithm 2, by doing step 4 and 5 repeatedly, we have a higher probability to select our new centroid that is farther than the previous centroid. Once we have chosen the K initial centroids, we can run the standard K-mean algorithm.

2.3.1.2 Time and Space Complexity

The space complexity for the K-means algorithm is $O((n + K)d)$, where n is the number of observations in our data set, K is the number of clusters, and d is the number of attributes. The time complexity for the K-means algorithm is also linear-time, which is $O(I \times K \times d \times n)$, where K is the number of clusters, I is the number of iterations for converging [28]. It is not hard to see, assuming we are choosing good initial centroids with small and low-dimensional data set is (small I, n , and d). The K-mean algorithm shows time and space efficient. But it is less efficient for huge high-dimensional data sets.

2.3.2 Partitioning Around Medoids (PAM)

The partitioning around medoids algorithm [29] was first proposed by Kaufman and Rousseeuw in 1990. It is an improved algorithm based on the K-means. Similar as the K-means algorithm,

it is a partitioning-based clustering algorithm and uses a distance matrix to specify the number of clusters. The only difference between the PAM and K-means algorithm is to compute and update our centroids. The K-means updates it with the mean of all the points in the current cluster. The centroid point selected by the K-medoids method is a point existing in the current cluster that has a minimal distance to all of the other points. It makes PAM less sensitive to outliers, but on the other hand, it shows less time efficiency. For PAM, we need to compute the distance for all pairs of the points in the current cluster. The following section shows the algorithm for the PAM clustering.

2.3.2.1 K-medoids Algorithm

The K-medoids algorithm [30] is a clustering algorithm based on PAM. It has the same idea as the K-means. Firstly, choose K observations in our data set as the K initial medoids which are also corresponding to the K clusters. Secondly, for each observation in our data set, we compute the distance (similarity measuring) to each medoid and assign it with the closest one. We calculate the sum distances from all of the observations in the current cluster to its medoid and choose one observation with the minimal sum distance as the new medoid. Doing that repeatedly until the medoids no longer change. The following section 3 showing the pseudocode of the K-medoids Algorithm.

Algorithm 3 Basic K-medoids algorithm

1: Randomly select K points as initial medoids.
2: **repeat**
3: Calculate the distance from each point to the medoids.
4: Form K clusters by assigning each point to its closest medoids.
5: Calculate the sum distances from all of the observations in the current cluster to its medoid.
6: Choose one observation with the minimal sum distance as the new medoid.
7: **until** The medoids do not change.

2.3.2.2 Time and Space Complexity

The overall time complexity of the original or K-medoids algorithm is $O(K(n-K)^2)$, where K is the number of clusters and n is the number of observations. To compute the similarity matrix for all pairs of the observations will take $O(n^2)$ and there are K clusters, so in total we have $O(K(n-K)^2)$ [30].

2.3.3 Density-based spatial clustering of applications with noise (DBSCAN)

The DBSCAN [31] is another partitional clustering approach. It is a density-based clustering and non-parametric algorithm. It was proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu back in 1996. It is resistant to noise and can handle clusters of different shapes and size. Similar to K-means, the DBSCAN also uses the Euclidean distance or some other similarity measurements to measure the distance between each observation in the data set. Therefore,it can also apply on high-dimensional feature space. However, unlike K-means, instead of directly providing K initial centroids as the number of clusters, DBSCAN takes **eps** and **min samples** as the parameters. By tuning those parameters carefully we can further control our clustering results.

Here are some definitions for the DBSCAN algorithm [5].

- **Eps**: With the point as the center, eps is the radius of this circle.
- **Minimum Samples**: For the center point to be considered as a core point, there are at least minimum number of points in the circle.
- **Core Point**: Within, eps drawing a circle, if there are more than specified number of points included in this circle then the center point is a core point.
- **Border Point**: Within eps draw a circle, if there are fewer than specified number of points included in this circle and it is the neighborhood of a core point, then the center point is a border point.
- **Noise Point**: If the point is either border point or core point, then it is a noise point.

The following Figure 2.6 shows the core, border, and noise points when the minimum samples is 7.

2.3.3.1 The DBSCAN Algorithm

By knowing the above definitions, we will introduce how DBSCAN works. Firstly, we label all points as the core, border, or noise points. For any other points within eqs of the core point if they are core points or border points, then we cluster them as the same cluster. If they are noise points we discard them. The pseudocode of the DBSCAN algorithm [5] is given in Algorithm 4.

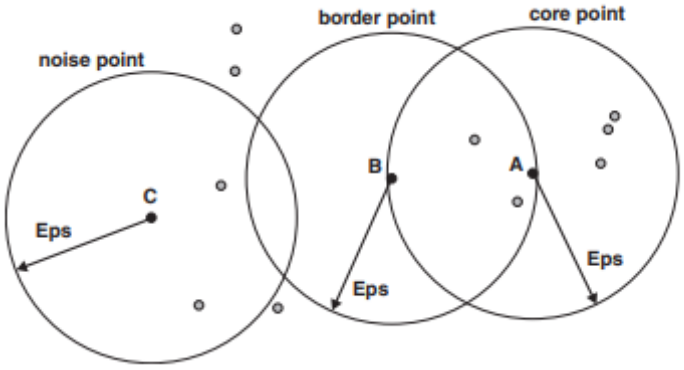


Figure 2.6: Core, border, and noise points at minimum sample = 7 [5].

Algorithm 4 The DBSCAN algorithm

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points within a distance *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

2.3.3.2 Time and Space Complexity

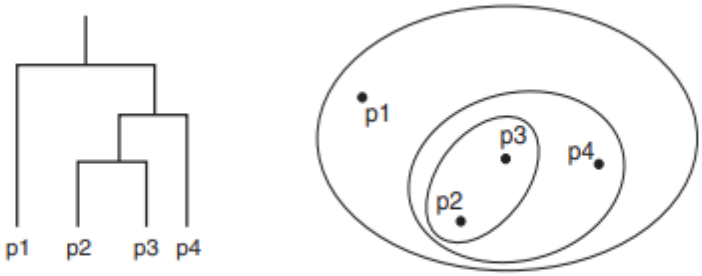
The time complexity for the DBSCAN algorithm is $O(n^2)$, where n is the number of points in our data set. The space complexity is $O(n)$, because for each point we need to store their information, i.e., their cluster label and the details of each point as a core, border, or noise point [5]. The DBSCAN algorithm can also work with high-dimensional data, showing in both good space and time complexity.

2.3.4 Agglomerative Hierarchical Clustering

Unlike the K-means and DBSCAN clustering algorithm, Agglomerative Clustering [5] is one of the types of hierarchical clustering. It produces a hierarchical tree and builds a hierarchy of clusters by nesting objects. There are mainly two types of generating hierarchical clustering [5].

- Agglomerative: "A Bottom up" approach, starts with each point as own cluster. At each step, merge it with the closest clusters until there is K clusters left.
- Divisive: "A Top down" approach, starts with all points in a same cluster. At each step, splits a part from that cluster until there is K clusters left.

Figure 2.7 shows the dendrogram of the sequences for merging or splitting. In our case study, we use the agglomerative as the method for hierarchical clustering.



(a) Dendrogram. (b) Nested cluster diagram.

Figure 2.7: A hierarchical clustering of four points shown as a dendrogram and as nested clusters [5].

Similar to the K-means and DBSCAN algorithms, the hierarchical clustering algorithms also use a similarity or proximity matrix to measure the distance for each cluster. i.e, Euclidean distance, Manhattan distance, cosine similarity, Gower's distance and Jaccard similarity. In our case study, we use the Euclidean distance metric to compute the linkage between each cluster.

Here is the basic concept of how Agglomerative hierarchical Clustering works. Firstly, we consider each point as an own cluster, then at each step, we merge the closest two clusters as a new cluster. We can do that recursively until there is only one cluster left. The pseudocode of Agglomerative Clustering algorithm [5] is showing in Algorithm 5.

Algorithm 5 The basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix between the new cluster and the original clusters.
5: **until** Only one cluster remains.

2.3.4.1 Proximity between Clusters

As we mentioned before, once we computed the proximity matrix for each point, we can determine the closest points and group them as the same cluster. But we still need to determine the closest clusters. There are several methods to achieve this, we can group them by MAX (complete linkage), MIN (single line), WARD, or AVERAGE. Grouping by **MAX** means that we are grouping based on the maximum distance (least similar) of two observations into two different clusters. Grouping by **MIN** means that we are grouping based on the minimum distance (most similar) of two observations into two different clusters. Grouping by **AVERAGE** means that we are grouping based on the average distance for all observations into two different clusters. Grouping by **WARD** means we are grouping based on the minimum variance of the two different clustering.

In our case study, we used the **WARD** as the distance measurement between sets of observations.

2.3.4.2 Time and Space Complexity

The space complexity for agglomerative hierarchical clustering algorithm is $O(n^2)$ [5], because we need at least $O(n^2)$ space to store the proximity matrix and addition $O(n)$ to store the information for each point, where n is the number of points. The time complexity for the agglomerative hierarchical clustering algorithm is $O(n^3)$. The $O(n^2)$ time is required to compute the proximity matrix for each observation. Step 3 also requires $O((n-i+1)^2)$ to compare all pair of clusters, where i is the number of iteration. Because for each iteration, we are grouping two clusters as a new one, so overall we need $O(n-i+1)$ to loop over all the

clusters. The step 4 requires $O(n-i+1)$ for updating the proximity matrix of each cluster. Overall, the time complexity is $O((n-i+1)^3)$, which is $O(n^3)$.

2.3.5 The Mean-shift clustering

Similar as the DBSCAN clustering approach, the mean-shift clustering is another partitional clustering approach. It is also a density-based clustering and non-parametric algorithm. The idea of the mean-shift clustering is similar to the classic K-means clustering approach, but compare with the K-means approach, there are no assumptions on the shape of the distribution or provided the K initials clusters [32]. The Mean-shift [33] was first proposed by Fukunaga and Hostetler in 1975. It also has lots of variants under the different use cases and the data type. I will not go into details here.

Here is the main idea for the mean-shift algorithm [32]. For the points in the d-dimensional space, we need to compute their density estimation, then gradient ascent procedure on the local estimated density until convergence. The modes of distribution can be represented by these stationary points of the process. The data points that approximately approach the same stationary point are clustered into the same cluster. To show them mathematically, given n data points $x_i \in \mathbb{R}^d$, the multivariate kernel density estimate using a radially symmetric kernel $K(x)$ (e.g. Gaussian, Rectangular and Epanechnikovs), is given by [32]:

$$\hat{f}_k = \frac{1}{nh^d}\sum_{i=1}^{n} K\big(\frac{x-x_i}{h}\big), \tag{2.18}$$

where h is the bandwidth (radius) of the kernel. The radially symmetric kernel is defined as [32]:

$$K(x) = c_k k(\|x\|^2), \tag{2.19}$$

where c_k is the normalization constant. Let $g(x) = -k(x)$ and $G(x) = c_g g(\|x\|^2)$, then the gradient of the density estimator can be defined as follows [32]:

$$\nabla \hat{f}(x) = \frac{2c_{k,d}}{nh^{d+2}}\Big[\sum_{i=1}^{n} g\Big(\Big\|\frac{x-x_i}{h}\Big\|^2\Big)\Big]\Big[\frac{\sum_{i=1}^{n} x_i g\big(\big\|\frac{x-x_i}{h}\big\|^2\big)}{\sum_{i=1}^{n} g\big(\big\|\frac{x-x_i}{h}\big\|^2\big)} - x\Big], \tag{2.20}$$

where the term in the first squared brackets is proportional to the probability density estimate at x. The term in second square brackets is the mean shift vector called **m**, which is proportional to the density gradient estimate at point x obtained with kernel K [32].

2.3.5.1 The Mean-shift Algorithm

By knowing the principle for the mean-shift clustering, and providing the above definition. We can further define our mean-shift algorithm [32] (see Algorithm 6). Figure 2.8 shows more details about how the mean-shift algorithm works in a two dimensional data space.

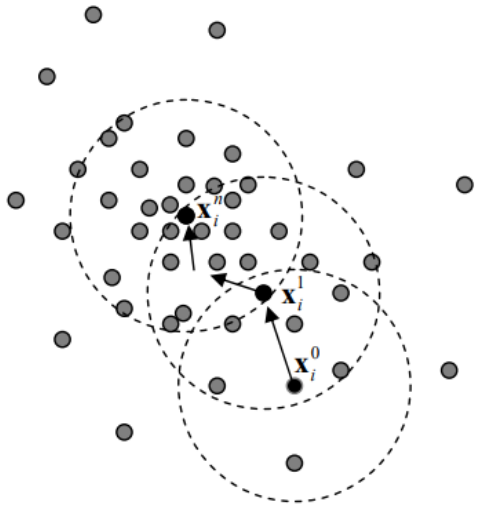


Figure 2.8: The plot is showing the mean shift procedure [32]: Starting at point x_0, then for each iteration we are computing the mean shift vector $m(x_i)$ and updating on the density estimation window $x_i^{t+1} = x_i^t + m(x_i^t)$. The grey points are the observations in our data set, the dark points are the window centres, and the dotted circles denote the density estimation windows.

Algorithm 6 The mean-shift clustering algorithm.

1: for any given point x_i
2: **repeat**
3: Compute the mean shift vector $m(x_i^t)$.
4: Update the density estimation window $x_i^{t+1} = x_i^t + m(x_i^t)$.
5: **until** convergence $\nabla f(x_i) = 0$

2.3.5.2 Time and Space Complexity

The time complexity for the mean-shift clustering algorithm is $O(n^2)$, where n is the number of points in our data set [34]. Because we need to use $O(n)$ to create the kernel, then $O(n)$ to compute their density estimation. The space complexity is $O(n)$. Because we need to store their density estimation for each point.

Chapter 3

Empirical Evaluation

In this chapter, we implement the combination of different similarity (distance) measures with clustering algorithms by using R studio and Python. We have also introduced how to perform their data preprocessing and further discussed their advantages and disadvantages. By running the different clustering algorithms in the several data sets we processed. We have shown our experiment results and further discussed how they could help us for solving our problems.

We evaluate the algorithms based on the three similarity measures: Gower's similarity, Euclidean distance and Dynamic time warping. Following the introductions in the previous section, they all adopted different data types as their input, so we set up three experiments. (see Table3.1)

Experiment	**Data type**	**Similarity matrix**	**Clustering algorithms**
A.	Continuous	Eudlicean Distance	K-Means Agglomerative DBSCAN Mean-Shift
B.	Continuous and Categorical	Gower's Distance	Hierarchical Partitioning Around Medoids (PAM)
C.	Time Series Sequence	Dynamic Time Warping	Hierarchical Partitioning Around Medoids (PAM)

Table 3.1: Combination of similarity matrix and clustering algorithms used in this experiments.

3.1 Experiment A

Like we mentioned in chapter 2, the Euclidean distance can work for high-dimensional data, but it can only accept numeric data. After our data prepossessing, we need to make sure that any of the columns we used should be numerical. To achieve this, we took the companies' user name "plus" their accounts name as each company's identifier. For each company's identifier, we can compute their daily profit and their daily revenue. Now, there are only two numeric variables left in our data set. But we are missing all of the information about the trend of each company. Our data should include another variable that can describe the amounts of growing or falling for each company's revenue. To achieve this, we fitted a linear regression model for each company's revenue, and their coefficients used to describe the trend of this company. Finally, there are three variables in our data set: daily revenue, daily profit and coefficient. To achieve better accuracy, two companies are only comparable if they are in the same location and under the same ANZSIC division. (For example, company A and B are only comparable if they are both in New Zealand, and they are both in the manufacturing industry.) Finally, there are four variables in our data set: "daily revenue", "daily profit", "coefficient" and their ANZSIC "division".

3.1.1 Data preprocessing

We need to implement the data-preprocessing before we applied any of the clustering algorithms. For the original data set, we removed any columns which were containing too many missing values. We can drop the "company" column because 65% of its values are missing. There is only one level for the "osp" variable and only two levels for the "account types". We can use either "user" or "connection" as the company's identifier. The one "user" may run several accounts, so we combined "user" with the "account name" as the business identifier. There is four levels of the "currency" variable, but there are too few observations in "AUG", "GBP", and "USD". So, we are only focusing on the NZD currency. After that, it seems variable "NZD" and "daily revenue" being duplicate. We can drop one of them. Finally, we need to drop "osp", "connection", "currency", "account name", "account type", "NZD" and formed a new variable called "identifier" by paste "user" with "account name".

We can get the ANZSIC division code for each "identifier" by mapping the "user" from the original data set to its "industry" in the second data set. There are a total of 38 levels in the "industry" variable, all of them were in the text format. We can distinguish which ANZSIC department they belong to and manually label them. Finally, we can combine it as a new variable "division" to our original data set.

Now, we can recompute the daily revenue by taking the average for the "daily revenue", then using the sum of the "revenue to date" divided by the total number of days to recompute the daily profit. (from the first-day record to the last day) We can also generate a new variable called "expenses" by using daily income minus daily profit. We also need to fit a

linear regression model for each revenue. Its coefficients are used to describe the trend for this company. For example, Figure 3.1 shows the daily revenue for company "0889e065-65c1-4867-8d21-7471436b255f" with the account name "Microsoft Recurring Licenses" in New Zealand. There is one valley in April of 2020 because New Zealand imposed a nationwide lockdown from March 25 to April 28 due to the impact of the COVID-19. The blue line is the best fitted line in linear regression. The grey band is showing the 95% confidence interval around the mean of the smoothed regression line. The intercept of this fitted line is 366.38 with a coefficient is 6.75. In other words, the daily revenue increases by $6.75 per day.

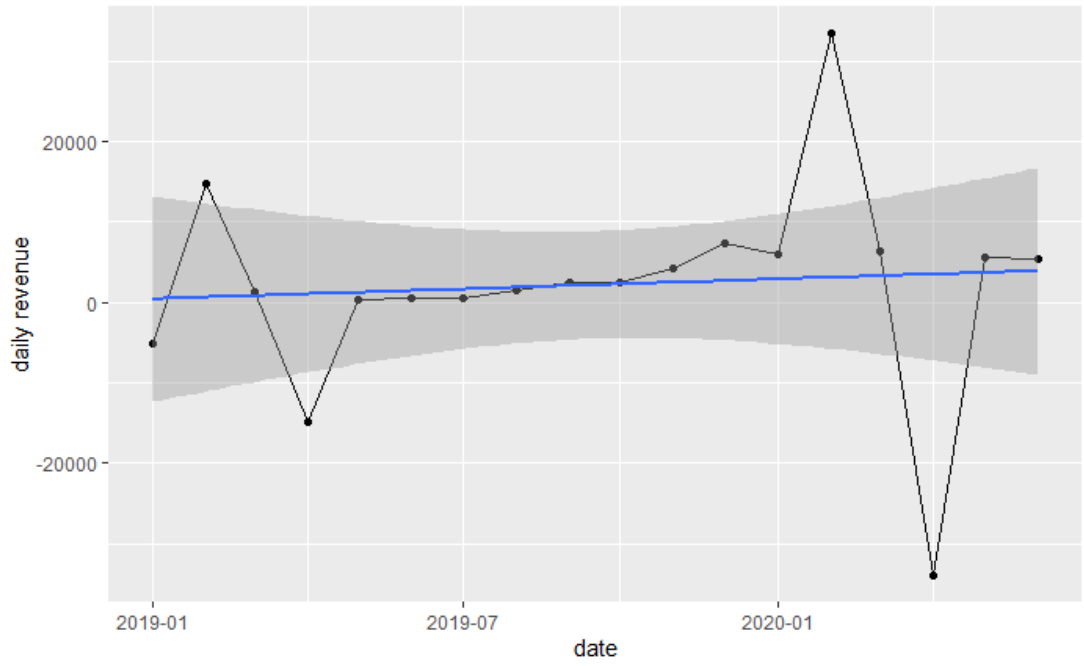


Figure 3.1: Figure shows the daily revenue of company "0889e065-65c1-4867-8d21-7471436b255f" with the account name "Microsoft Recurring Licenses" and its fitted linear regression line.

Now, for each company, we have their daily income, daily profit, coefficient, industry division and daily expenditure. We reduced our data from 77898 records to 3148 records. It only has one record for each account now. (see figure 3.2) Before we move to the next step, we need to check the correlation between each pair of variables. Figure 3.3 is a correlation matrix that shows the correlations between each pair of variables. The correlation between expense and daily income is close to one. As we mentioned in chapter 2, the positive one is for a completely

positive relation, and the negative one is for a completely negative relation. This means, there is a strong positive relation between expenses and daily income – the higher daily income the higher expenses. We can either delete the expenses or daily income. Subsetting by the division code, we can apply clustering algorithms on each of the subsets.

	identifier	daily_profit	daily_income	expenses	coeff	Division
2	00415952-3309-4e6b-af4b-b1c02dc2d08b+Sales	82.0575227	6220.60697	6138.549447	-0.731326974	M
3	009f7f82-f9be-470a-8df0-1e2043829cd5+Coffee Sales	18.76872818	16376.428	16357.65927	-19.84501739	G
4	009f7f82-f9be-470a-8df0-1e2043829cd5+Food Sales	19.84033666	17489.6072	17469.76686	-24.93585019	G
5	009f7f82-f9be-470a-8df0-1e2043829cd5+Other Income	55.19923845	4383.18	4327.980762	3.283642304	G
6	009f7f82-f9be-470a-8df0-1e2043829cd5+Soft Beverages Sales	3.42936409	3198.612	3195.182636	-3.546503343	G
7	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Interest Income	0.003452381	0.2075	0.204047619	0.000977224	G
8	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Sales	-76.96506045	40327.18615	40404.15121	-70.35644978	G
9	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Sales - Dental Products	0.312251185	383.8011765	383.4889253	-1.107550781	G
10	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Sales - Dr A Fung	15.91632701	15298.7836	15282.86727	-85.3892699	G
11	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Sales - Dr J Yeow	12.98234597	14981.27	14968.28765	-55.80231473	G
12	00c0d1bb-f1fa-4b19-bd3c-d0aa04fdbf69+Sales - V Hughes	-0.831800948	1752.319333	1753.151134	-3.557853388	G
13	02246f3d-262e-4fa6-9db6-b98df684b210+Fuel Refunds	0.646996047	355.098	354.451004	-0.017321101	G
14	02246f3d-262e-4fa6-9db6-b98df684b210+Interest Received	-0.005662338	592.4084615	592.4141239	-0.396877038	G
15	02246f3d-262e-4fa6-9db6-b98df684b210+Other Revenue	4.749633028	515.3242308	510.5745977	0.09197161	G
16	02246f3d-262e-4fa6-9db6-b98df684b210+Retentions	138.5888197	39199.94467	39061.35585	-63.0720705	G
17	02246f3d-262e-4fa6-9db6-b98df684b210+Sales	5136.058455	747918.4264	742782.3679	-1424.312673	G
18	02246f3d-262e-4fa6-9db6-b98df684b210+Wage Subsidy Received	-2.898666667	973.916	976.8146667	-7.418865693	G
19	0245c701-833d-419c-9ad4-64bff731a0b6+Course Documentation Packs	7.498427673	692.3125	684.8140723	0.327931122	Q
20	0245c701-833d-419c-9ad4-64bff731a0b6+Interest Income	0.926719577	40.5825	39.65578042	0.215686342	Q

Figure 3.2: Figure shows the data set after the preprocessing.

	daily_profit	daily_income	expenses	coeff
daily_profit	1	0.393899	0.381764	-0.302533
daily_income	0.393899	1	0.999913	-0.300978
expenses	0.381764	0.999913	1	-0.298317
coeff	-0.302533	-0.300978	-0.298317	1

Figure 3.3: Correlation matrix for each pair of variables.

3.1.2 Experiment Result

Figure 3.4 provided a general view of how companies distribution seems to be. It is a 3D scatter plot, each point represents a company in our data set. The x-axis, y-axis and z-axis are showing the daily revenue, coefficient and daily revenue, respectively. The different industry divisions are representing by different colours. From this plot, it is not hard to see that most of the companies' daily profit are in the range of -$100 to $100 (in NZD), with the daily revenue $0 to $100 (in NZD), and their coefficients are around 0. We may also conclude from the plot that the higher daily revenue and daily profit, the lower coefficient. The different industry divisions are not showing clearly in this plot.

As we mentioned before, we divided the companies by their industry type. There are 17 industry divisions in our data set, and for each division, we need to run four different clustering

algorithms. For simplicity. Instead of showing all of our results, we are just showing some typical examples.

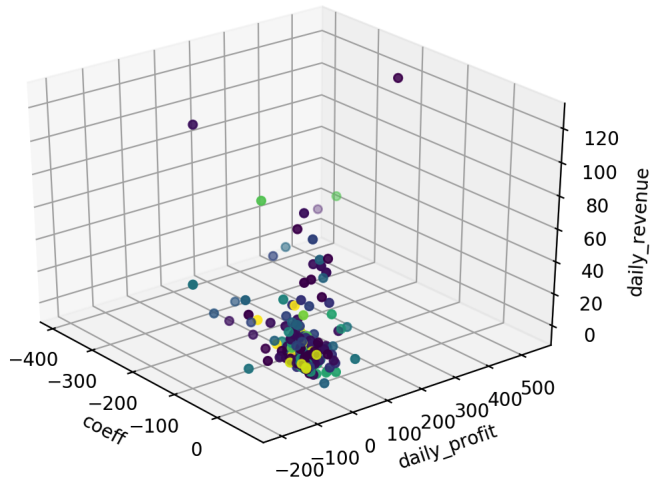


Figure 3.4: The 3-D scatter plot showing the overall distribution of the companies. (notice: the value of x, y and z-axis have been normalized)

3.1.2.1 Result of K-means Clustering

Figure 3.5 shows the result by running the K-means algorithm at the "I" (Transport, Postal and Warehousing) industry division. We set the number of clusters equal to 3. It clearly shows three cluster groups, coloured as yellow, purple and green. Linking with the x, y, z-axis, we may conclude that those companies in the yellow, purple and green group will respectively label as "risk", "normal" and "good". Table 3.2 shows the number of companies for each cluster, their average coefficients, daily revenue, daily profit and their status we labelled. For each company in the same cluster group, we labelled them as the same category (status).

To make some conclusions, we can say the company "3cc9f3e8-9d71-4e0a-94dd-a4090afba37c" with the account name "Less Revenue - Offset (RCM)" performed well compared with other companies in the industry type of transport, postal and warehousing. The company "8179867f-4560-4c2c-a591-5816a577be34" with the account name "Vehicle Disposal-Sales Received" performed poorly compared with other companies in the same industry type. The company "8e9826ef-a11e-49fe-ae78-342a5e1ad460" with the account name "Two Factor Authentication" performed generally compared with other companies in this same industry type.

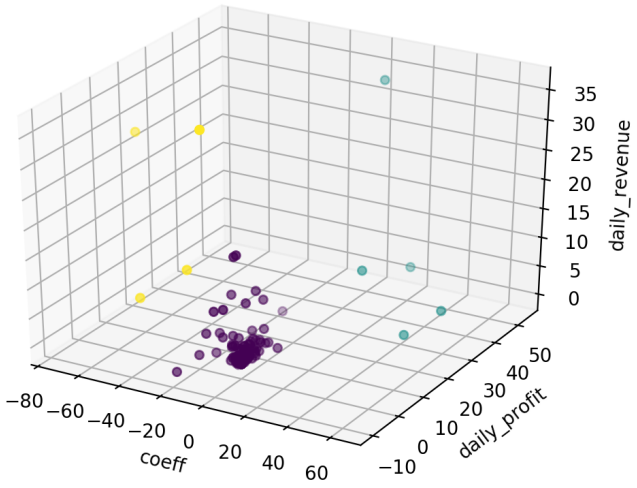


Figure 3.5: The 3-D scatter plot shows the result of K-means clustering at the industry division "I". (notice: the value of x, y and z-axis have been normalized)

	Group Yellow	**Group Purple**	**Group Green**
Number of Companies	4	210	5
Average Daily Profit	1.86	0.897	31.6
Average Daily Revenue	22.7	0.714	12.4
Average Coefficient	-37.7	-0.934	37.1
Companies Status	Risk	Normal	Good

Table 3.2: A table shows the number of companies, average coefficients, daily revenue and daily profit for each cluster group. (notice: all the values in here have been normalized)

3.1.2.2 Result of Agglomerative Hierarchical Clustering

Figure 3.6 shows the result of agglomerative hierarchical clustering by setting the number of clusters equal to three. We almost get the same result as the K-means except for three companies that previously in the yellow group are now classified to the purple group. Table 3.3 shows more details for companies in each cluster group. We can also get the same conclusion as the K-means.

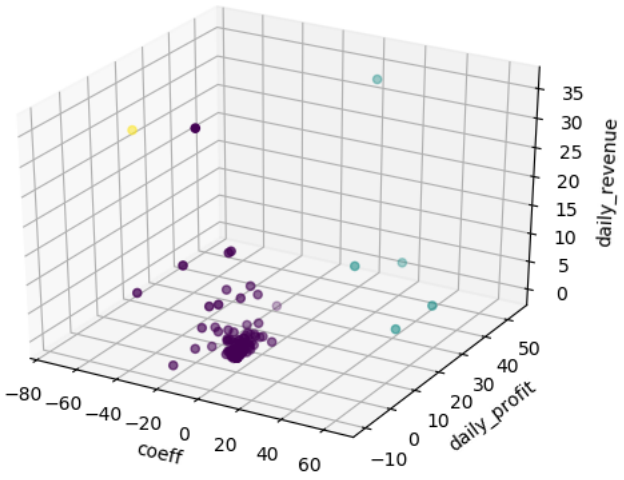


Figure 3.6: The 3-D scatter plot shows the result of agglomerative hierarchical clustering at the industry division "I". (notice: the value of x, y and z-axis have been normalized)

	Group Yellow	**Group Purple**	**Group Green**
Number of Companies	1	213	5
Average Daily Profit	18.1	0.834	31.6
Average Daily Revenue	28.2	0.999	12.4
Average Coefficient	-73.3	-1.28	37.1
Companies Status	Risk	Normal	Good

Table 3.3: A table shows the number of companies, average coefficients, daily revenue and daily profit for each cluster group. (notice: all the values in here have been normalized)

3.1.2.3 Result of DBSCAN Clustering

Figure 3.7 shows the result of DBSCAN clustering, by set the **eps** equal to 10 and the **minimum samples** equal to 5. The result shows two cluster groups. The core points are representing by the yellow points, and the purple points are the noise points. In other words, the yellow group represented the average level of most companies in the industry division of transport, postal and warehousing. Comparing with the result of hierarchical and K-means clustering, we need to set parameters for DBSCAN clustering rather than provide the initial number of the clusters. The DBSCAN clustering has successfully divided the companies into two groups — "normal" (yellow) and "unusual" (purple). The companies in the normal group are the same as the other two clustering results. The companies in the unusual group may be above average (good) or below average (risk), but our DBSCAN clustering can not further distinguish them. See table 3.4 for more details.

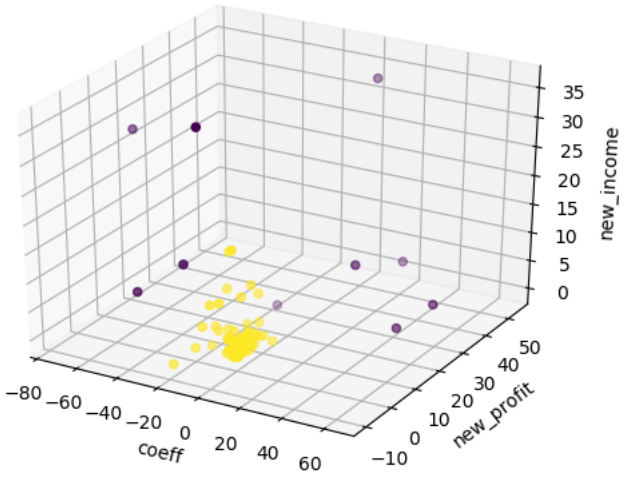


Figure 3.7: The 3-D scatter plot shows the result of DBSCAN clustering at the industry division "I". (notice: the value of x, y and z-axis have been normalized)

3.1.2.4 Result of Mean-shift Clustering

Figure 3.8 shows a result of the mean-shift clustering with the bandwidth set to 40. Comparing with other clustering results, it is most similar to hierarchical clustering, because mean

	Group Purple	Group Yellow
Number of Companies	10	209
Average Daily Profit	18.9	0.79
Average Daily Revenue	15.3	0.71
Average Coefficient	2.5	-0.89
Companies Status	Unusual	Normal

Table 3.4: A table shows the number of companies, average coefficients, daily revenue and daily profit for each cluster group. (notice: all the values in here have been normalized)

shift clustering and hierarchical clustering are both partitional clustering. The Table 3.5 shows more details for companies in each cluster group. We can make the same conclusion as the hierarchical clustering.

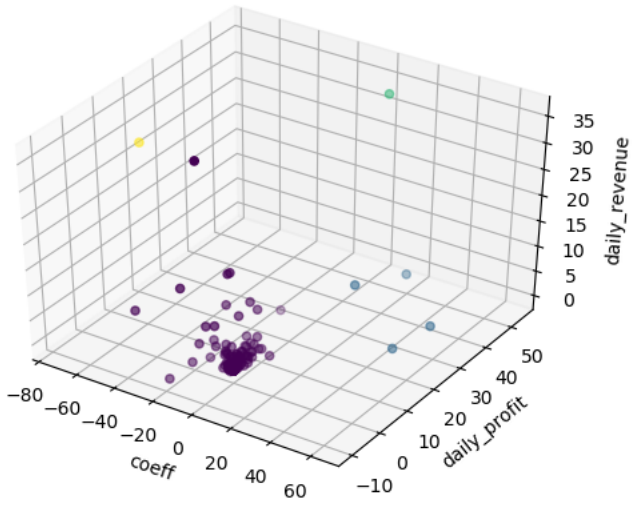


Figure 3.8: The 3-D scatter plot shows the result of K-means clustering at the industry division "I". (notice: the value of x, y and z-axis have been normalized)

3.1.3 Limitations and Precautions

There are two defects in our experiment A. When we compute their daily revenue and daily profit, we lose all the information about their patterns and trends. Although we can still get

	Group Yellow	Group Purple	Group Blue	Group Green
Number of Companies	1	213	4	1
Average Daily Profit	18.1	0.834	26.0	54.0
Average Daily Revenue	28.2	0.999	7.26	32.9
Average Coefficient	-73.3	-1.28	45.8	2.42
Companies Status	Risk	Normal	Good	Good

Table 3.5: A table shows the number of companies, average coefficients, daily revenue and daily profit for each cluster group. (notice: all the values in here have been normalized.)

their coefficient by fitting a linear model on their revenues' time series, that linear model can not describe all of the companies revenue. Some of the coefficients are non-significant (p-value < 0.05), and we have no evidence to say that their revenue is affected by the date under this linear model. For example, in figure 3.9 the daily revenue of company "09906136-5446-48cf-a03c-2488e1e94f05" seems to have a seasonal effect in one period year, but our simple linear model does not match them well. The p-value for this example is 0.133, which exceeds 0.05. More generally, in experiment A, some of our coefficients are not reliable, which means it is hard to describe their trends.

3.2 Experiment B

The only difference between experiment A and experiment B is the way to consider the ANZSIC division code. In experiment A, we only use it to subset our data. In other words, we did not treat it as an individual feature and used it in our clustering algorithms. But in experiment B, we want to consider the ANZSIC division code as a new variable and put that in our clustering algorithms. In order to achieve this, we need to use Gower's distance to compute the similarity matrix for each pair of observations. As we mentioned in section 2, Gower's similarity is used to calculate the similarity between continuous and categorical variables. Finally, we can apply our clustering algorithms based on this similarity matrix.

3.2.1 Data preprocessing

The data preprocessing for experiment B is exactly same as the experiment A. I will not go into details here. (see same figure 3.2 and 3.3)

3.2.2 Experiment Result

In this section, we will show our results for applying two clustering algorithms — partitioning around medoids (PAM) and hierarchical based on the Gower's similarity measure.

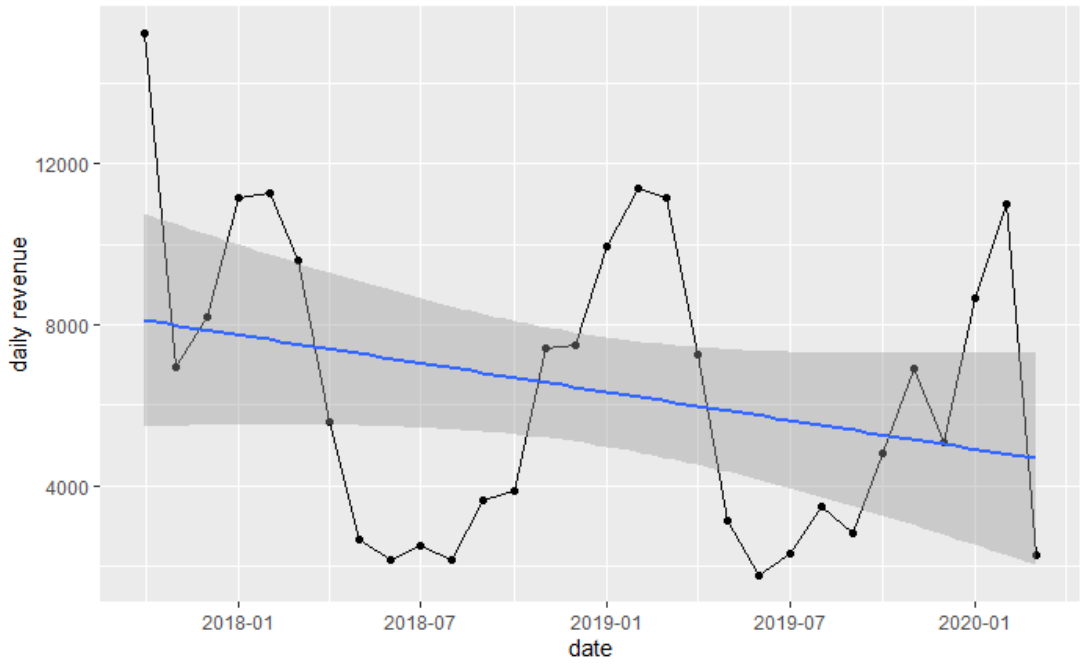


Figure 3.9: Figure shows the daily revenue of company "09906136-5446-48cf-a03c-2488e1e94f05" with the account name "Retail" and its fitted linear regression line.

3.2.2.1 Result of Partitioning Around Medoids (PAM) Clustering

Figure 3.10 shows the PAM clustering result with the number of clusters equal to three based on Gower's similarity measure. In order to calculate the Gower's similarity matrix, we set the parameters w_1, w_2, w_3 and w_4 to 10000, 10000, 1000 and 0.0001 respectively. They also are the input weights and representing four attributes in our data set. The x-axis and y-axis are no longer interpretable since we used the dimension reduction [35]. By looking at Figure 3.10, we can see that the first cluster group has the most number of companies, while the third group has the least, but in general, it is hard to interpret for more details and not showing any associated clusters.

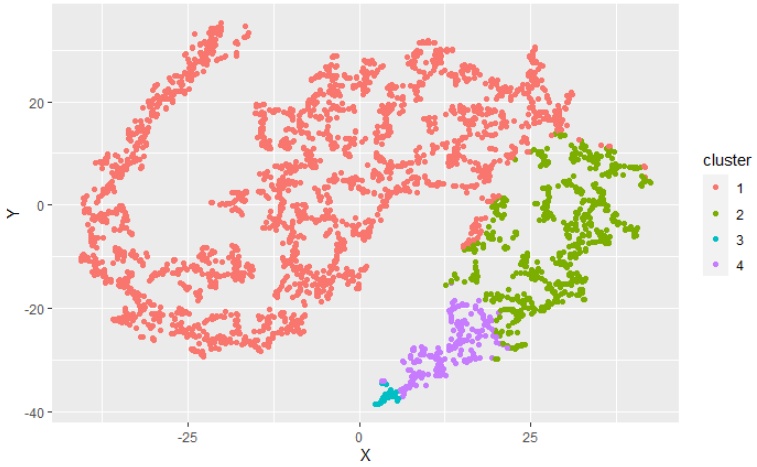


Figure 3.10: Figure showing the result of the PAM clustering based on the Gower's similarity matrix with the number of clusters equal to four. (after the dimension reduce)

3.2.2.2 Result of Hierarchical Clustering

Figure 3.11 shows a hierarchical tree as the hierarchical clustering result based on Gower's similarity matrix. The hierarchical tree is divided into four red rectangles, representing four cluster groups. Three of them are very tiny and overlapping on the left-hand side. The y-axis is the tree height, and the x-axis are showing all the companies. There are lots of overlaps because we have 3148 companies and limited space. For companies in the same red box, they belong to the same cluster group. We can also see that most companies have classified into one group, which also matches the partitioning around medoids (PAM) clustering result.

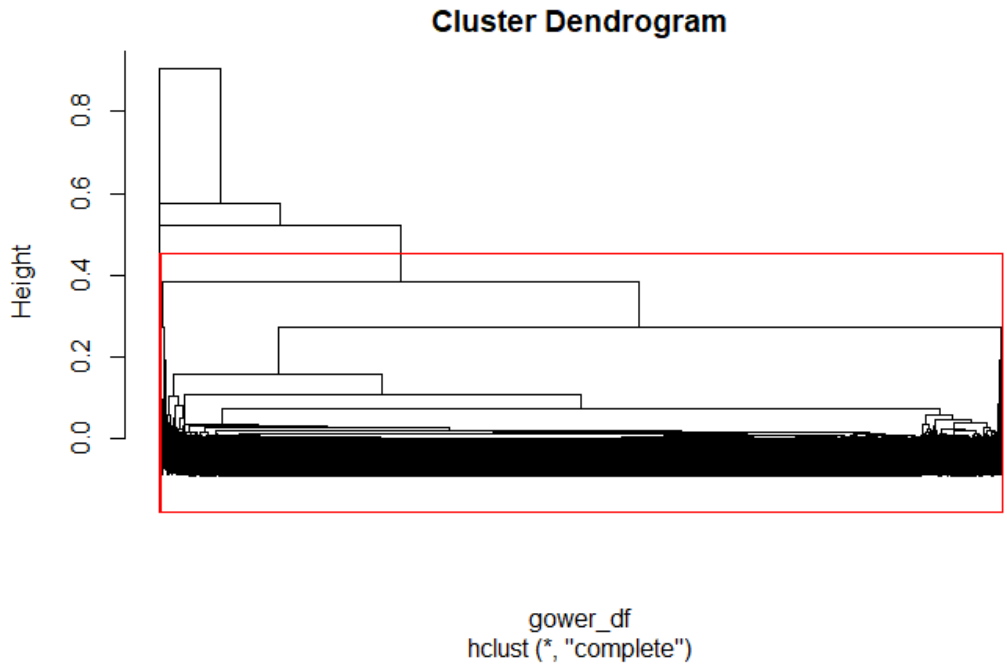


Figure 3.11: Figure showing the hierarchical tree based on the Gower's similarity matrix with the number of clusters equal to four.

3.2.3 Limitations and Precautions

Comparing with the three dimensions data set we used in experiment A, we applied clustering algorithms on a four-dimensions data set. That means we need some dimension reductions [35] to visualized our result, which will make our result less interpretable. It may miss some information compare with the original list of features. Same as experiment A, it also unable to compare the revenue trend between companies. On the other hands, due to the definition for Gower's distance, in order to get a meaningful result, we need to choose the most suitable weights for the continuous and categorical variables. But in our experiment, we do not have any better way rather than tune that mutually.

3.3 Experiment C

To overcome experiment A's bottleneck, we tried a new similarity measure named dynamic time warping. Like we mentioned in chapter 2, the euclidean distance "parallel" comparing two time series sequences, but the dynamic time warping can allow the correlation between the items inside a sequence of observations. More generally, for each company, rather than only compare their revenue and profit, the dynamic time warping can also compare their common patterns (trends).

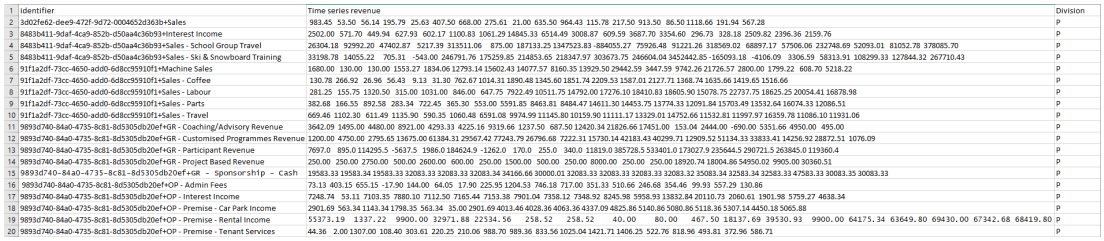
The dynamic time warping can only adopt a numeric input sequence. We need to use the companies' user name "plus" their accounts name as each company's identifier. But rather than providing their daily revenue and daily profit, we are now providing a time series that contains a sequence of daily revenues record. We also need the ANZSIC division code to help us distinguish them. Finally, there are two variables left in our data set: a time series that contains a sequence of daily revenue, and the ANZSIC division code.

3.3.1 Data preprocessing

For the same reason as the experiment A, We need to delete variables "osp", "connection", "currency", "account name","account type", "NZD" and formed two new variables named "identifier" and "division".

In order to generate a time series sequence for each company's identifier, we need to aggregate their "daily revenue" by month and store it as a list. Because the revenue recorded for each company is not a uniform distribution, and it may not be an equal length. For example, in our data set, company A may have 366 revenue records, one is on the first day of 2018, others were recorded daily in 2019, a company B has 36 revenue records, it was recorded monthly from 2017 to 2019. To solve this problem, we only consider their monthly revenue for the last 18 months from June 2020 (one and a half cycle, one year as a cycle). To achieve this, we aggregate their daily revenue by month and take their mean if their length is less than 18. Then, we take the subset which contains the last 18 elements. For each "identifier", we have

their division code and time series revenue. We reduced our data from 77898 records to 2423 records. (see figure 3.12) Subsetting by the division code, we can apply clustering algorithms on each of the subsets.



	Identifier	Time series revenue	Division
2	3d02fe62-dee9-472f-9d72-0004652d363b+Sales	983.45 53.50 56.14 195.79 25.63 407.50 668.00 275.61 21.00 635.50 964.43 115.78 217.50 913.50 86.50 1118.66 191.94 567.28	P
3	8483b411-9daf-4ca9-852b-d50aa4c36b93+Interest Income	2502.00 571.70 449.94 627.93 602.17 1100.83 1061.29 14845.33 6514.49 3008.87 609.59 3687.70 3354.60 296.73 328.18 2509.82 2396.36 2159.76	P
4	8483b411-9daf-4ca9-852b-d50aa4c36b93+Sales - School Group Travel	26304.18 92992.20 47402.87 5217.39 313511.06 875.00 187133.25 1347523.83 -884055.27 75926.48 91221.26 318569.02 68897.17 57506.06 232748.69 52093.01 81052.78 378085.70	P
5	8483b411-9daf-4ca9-852b-d50aa4c36b93+Sales - Ski & Snowboard Training	33198.78 14055.22 705.31 -543.00 246791.76 175259.85 214853.65 218347.97 303673.75 246604.04 3452442.85 -165093.18 -4106.09 3306.59 58313.91 108299.33 127844.32 267710.43	P
6	91f1a2df-73cc-4650-add0-6d8cc95910f1+Machine Sales	1680.00 130.00 130.00 1553.27 1834.06 12793.14 15602.43 14077.57 8160.35 13929.50 29442.59 3447.59 9742.26 21726.57 2800.00 1799.22 608.70 5218.22	P
7	91f1a2df-73cc-4650-add0-6d8cc95910f1+Sales - Coffee	130.78 266.92 26.96 56.43 9.13 31.30 762.67 1014.31 1890.48 1345.60 1851.74 2209.53 1587.01 2127.71 1368.74 1635.66 1419.65 1516.66	P
8	91f1a2df-73cc-4650-add0-6d8cc95910f1+Sales - Labour	281.25 155.75 1320.50 315.00 1031.00 846.00 647.75 7922.49 10511.75 14792.00 17276.10 18410.83 18605.90 15078.75 22737.75 18625.25 20054.41 16878.98	P
9	91f1a2df-73cc-4650-add0-6d8cc95910f1+Sales - Parts	382.68 166.55 892.58 283.34 722.45 365.30 553.00 5591.85 8463.81 8484.47 14611.30 14453.73 13774.33 12091.84 15703.49 13532.64 16074.33 12086.51	P
10	91f1a2df-73cc-4650-add0-6d8cc95910f1+Sales - Travel	669.46 1102.30 611.49 1135.90 590.35 1060.48 6591.08 9974.99 11145.80 10159.90 11111.17 13329.01 14752.66 11532.81 11997.97 16359.78 11086.10 11931.06	P
11	9893d740-84a0-4735-8c81-8d5305db20ef+GR - Coaching/Advisory Revenue	3642.09 1495.00 4480.00 8921.00 4293.33 4225.16 9319.66 1237.50 687.50 12420.34 21826.66 17451.00 153.04 2444.00 -690.00 5351.66 4950.00 495.00	P
12	9893d740-84a0-4735-8c81-8d5305db20ef+GR - Customised Programmes Revenue	1200.00 4750.00 2795.65 13675.00 61384.31 29567.42 77243.79 26796.68 7222.31 15730.14 42183.43 40299.71 12909.52 51134.33 33833.41 14256.92 28872.51 1076.09	P
13	9893d740-84a0-4735-8c81-8d5305db20ef+GR - Participant Revenue	7697.0 895.0 114295.5 -5637.5 1986.0 184624.9 -1262.0 170.0 255.0 340.0 11819.0 385728.5 533401.0 173027.9 235644.5 290721.5 263845.0 119360.4	P
14	9893d740-84a0-4735-8c81-8d5305db20ef+GR - Project Based Revenue	250.00 250.00 2750.00 500.00 2600.00 600.00 250.00 1500.00 500.00 250.00 8000.00 250.00 250.00 18920.74 18004.86 54950.02 9905.00 30360.51	P
15	9893d740-84a0-4735-8c81-8d5305db20ef+GR - Sponsorship - Cash	19583.33 19583.34 19583.33 32083.33 32083.33 32083.34 34166.66 30000.01 32083.33 32083.33 32083.33 32083.32 35083.34 32583.34 32583.33 47583.33 30083.35 30083.33	P
16	9893d740-84a0-4735-8c81-8d5305db20ef+OP - Admin Fees	73.13 403.15 655.15 -17.90 144.00 64.05 17.90 225.95 1204.53 746.18 717.00 351.33 510.66 246.68 354.46 99.93 557.29 130.86	P
17	9893d740-84a0-4735-8c81-8d5305db20ef+OP - Interest Income	7248.74 53.11 7103.35 7880.10 7112.50 7165.44 7153.38 7901.04 7358.12 7348.92 8245.98 9958.93 13832.84 20110.73 2060.61 1901.98 5759.27 4638.34	P
18	9893d740-84a0-4735-8c81-8d5305db20ef+OP - Premise - Car Park Income	2901.69 563.34 1143.34 1798.35 563.34 35.00 2901.69 4013.46 4028.36 4063.36 4337.09 4825.86 5140.86 5080.86 5118.36 5307.14 4450.18 5065.88	P
19	9893d740-84a0-4735-8c81-8d5305db20ef+OP - Premise - Rental Income	53373.19 1337.22 9900.00 32971.88 22534.56 258.52 258.52 40.00 80.00 467.50 18137.69 39530.93 9900.00 64175.34 63649.80 69430.00 67342.68 68419.80	P
20	9893d740-84a0-4735-8c81-8d5305db20ef+OP - Premise - Tenant Services	44.36 2.00 1307.00 108.40 303.61 220.25 210.06 988.70 989.36 833.56 1025.04 1421.71 1406.25 522.76 818.96 493.81 372.96 586.71	P

Figure 3.12: Data set after pro-processing.

3.3.2 Experiment Result

There are 17 industry divisions, each contains the companies name, account names and their revenue's time series. For each division, we need to apply two clustering algorithms — hierarchical and partitioning around medoids (PAM). By setting the number of clusters equal to 10, we can get 10 cluster groups for each division. We only shown our clustering result for the industry division Transport, Postal and Warehouse ("I"). There are totally 147 companies in this industry division.

3.3.2.1 Result of Hierarchical Clustering

Figure 3.13 shows the result of the dynamic time warping with the hierarchical clustering algorithm at the industry division "I" (Transport, Postal and Warehousing). There are a total of 10 cluster groups, shown as the subplot. For each subplot, the x-axis shows the revenue values in z-normalized [36], the y-axis shows the number of last 18 months from June 2020 (one and half year). The coloured lines represent the companies' revenue time series, and the dashed lines are the estimate patterns (trends) for different cluster groups. Notice, some of the dashed lines have overlapped by the coloured lines.

We may also conclude from Figure 3.13, those companies in the cluster group 1, 10 and 4 they seem to have a decreasing trend. Those companies in the cluster group 7, 5 and 9 seem to have an increasing trend, while the other groups seem to have remained constant. The average revenue for all the companies at the group 7, 5 and 9 are \$26970.5, \$14031.5 and \$441.1 respectively. The average revenue for the companies at the group 1, 4 and 10 are \$72859, \$38.9 and \$7021.47, respectively. (see Table 3.6 for more detail)

For the same company “3cc9f3e8-9d71-4e0a-94dd-a4090afba37c” with the account name “Less Revenue - Offset (RCM)” and “8179867f-4560-4c2c-a591-5816a577be34” with the account name “Vehicle Disposal - Sales Received”, they were divided into the cluster group 1 and 7 respectively. The result looks a little bit contradictory, the companies with higher daily revenue are more likely to decrease. However, it also suggested our findings in the experiment A.

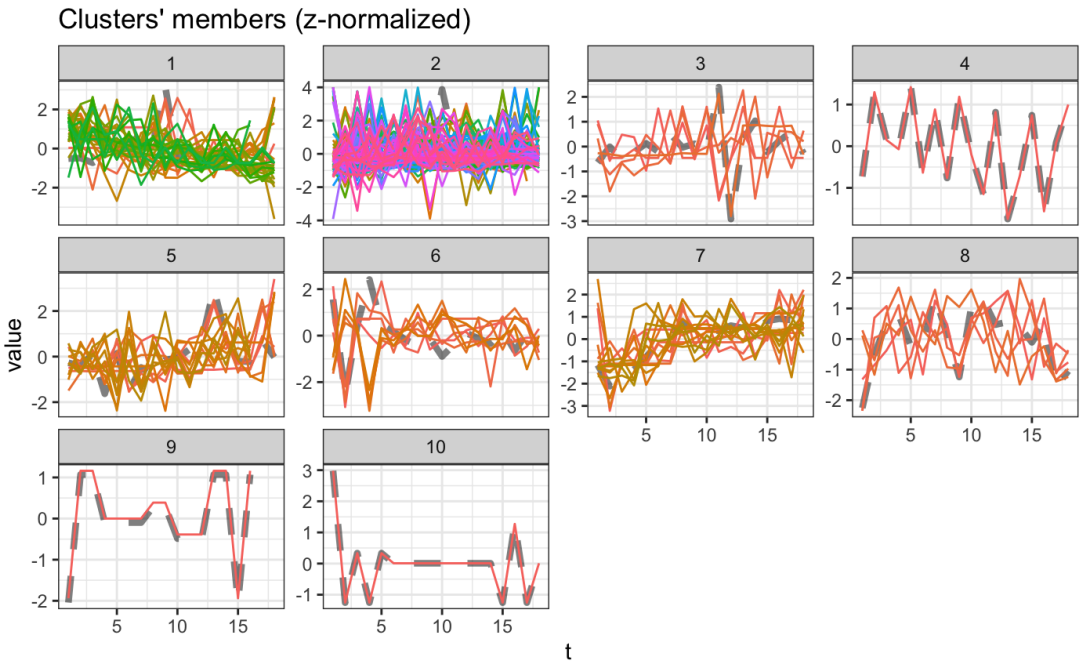


Figure 3.13: Figure showing the result of the dynamic time warping with the hierarchical clustering for the industry division “I”. (notice: the x axis have been z-normalized [36])

3.3.2.2 Result of Partitioning Around Medoids (PAM) Clustering

Figure 3.14 shows the result of the dynamic time warping with the hierarchical clustering algorithm at the industry division “I”. It has 10 cluster groups, shown as the subplots. The x, y, z-axis are the same as Figure 3.13. We may conclude that those companies in cluster group 1, 10 and 8 have the increasing trends. Those companies in cluster group 6 have the decreasing trends, for other companies in other cluster groups have constant trends. Table 3.7 shows the average revenue for those companies in the same cluster group. For the same company “3cc9f3e8-9d71-4e0a-94dd-a4090afba37c” with the account name “Less Revenue -

Pattern trend	Cluster groups	Number of companies	Average revenue
Increase	7	15	$ 26970.54
	5	14	$ 14031.48
	9	1	$ 441.09
Decrease	1	34	$ 72859.01
	10	1	$ 38.88
	4	1	$ 7021.47
Constant	2	92	$ 29770.54
	3	5	$ 3175.40
	6	8	$ 32610.27
	8	5	$ 2732.11

Table 3.6: A table shows more details of Figure 3.13.

Offset (RCM)" and "8179867f-4560-4c2c-a591-5816a577be34" with the account name "Vehicle Disposal - Sales Received", they were divided into the cluster group 6 and 8 respectively. It also suggested our conclusions.

To compare the result of the hierarchical clustering with the partitional clustering. We can say: for most companies, the result of their hierarchical clustering is the same as the result of the partitional clustering. In other words, they are more likely to classified as the same cluster group neither use hierarchical clustering or partitional clustering. For example, those companies in the cluster groups 1, 2, 7, and 5 of the hierarchical clustering are similar to the cluster groups 6, 4, 8, and 7 of the partitioned clustering respectively. To compare with the results of experiment A, we can also see that rather than group companies by their numeric values, the dynamic time warping group companies based on their patterns. In some ways, it did solve some problems that the euclidean distance could not handle.

3.3.3 Limitations and Precautions

Rather than fit a regression line to describe the companies trend, the dynamic time warping is clustering based on their similar patterns. It provides us with a new approach to solve this problem, but it still has some drawbacks. The dynamic time warping can only accept one time series as the input, which represents two dimensions of our data (date and revenue). We lost all the information about the "revenue to date" variable. Secondly, the dynamic time warping assumes each company records their revenue at least once a month. Otherwise, it will reduce the accuracy. On the other aspect, the dynamic time warping pays more attention on the similar "pattern" rather than the numeric values. This means that instead of groups by the revenue, it groups by the patterns. We can only get the conclusions by doing the data visualization. Finally, the results will be messy, contains thousands of different lines with different colours

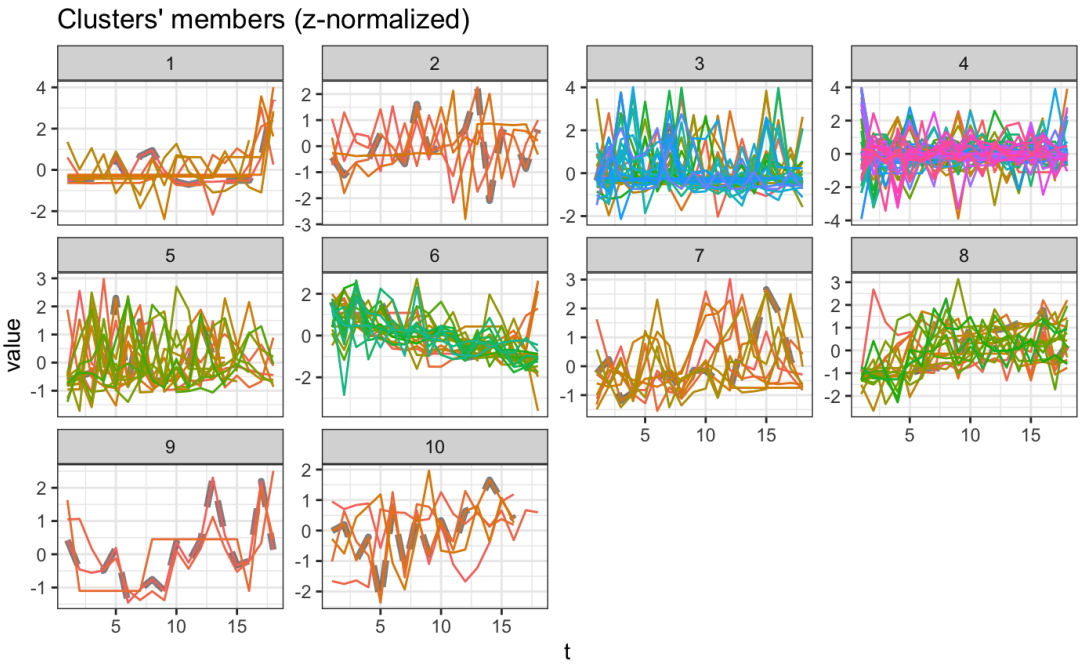


Figure 3.14: Figure showing the result of the dynamic time warping with the partitional clustering for the industry division "I". (notice: the x axis have been z-normalized [36])

Pattern trend	Cluster groups	Number of companies	Average revenue
Increase	1	8	$ 4280.64
	10	5	$ 6007.43
	8	18	$ 71236.21
Decrease	6	24	$ 83317.4
Constant	5	16	$ 7019.48
	2	5	$ 4486.86
	3	36	$ 26377.71
	4	51	$ 18438.94
	7	10	$ 65618.00
	9	3	$ 29318.35

Table 3.7: A table shows more details of Figure 3.14.

overlapping in the same plot.

Chapter 4

Conclusions and Future Work

In this dissertation, we have presented several approaches to clustering companies based on their key performance indicators. We tried three combinations of different similarity measures and clustering algorithms. We show that Gower's similarity measure does not suit to solve this problem. The euclidean distance and the dynamic time warping similarity measures with both the partitioning and hierarchical clustering gives the reasonable clusters to understanding the companies classification based on their KPIs. We have only shown our clustering results for companies in the industry division transport, postal and warehousing. By setting the most suitable number of clusters and doing some data visualisations, we can get the estimated status for each company. In common words, we have said the company "3cc9f3e8-9d71-4e0a-94dd-a4090afba37c" with the account name "Less Revenue - Offset (RCM)" performed well compared with other companies in the industry type of transport, postal and warehousing. The company "8179867f-4560-4c2c-a591-5816a577be34" with the account name "Vehicle Disposal-Sales Received" performed poorly compared with other companies in the same industry type. The other companies performed generally. We also found that the higher the daily revenue/profit, the greater the chance of decline, vice versa. The COVID-19 does affect business marketing however, we are not going to the details in this thesis.

We have 16 industry divisions in total, the complete result was excluded to save on space, but a full version is provided at `https://github.com/xyao824/Cluster` with code to draw that complete diagrams for each division. We got completely different results by using the several similarity measures. Each similarity measure took different type of inputs, and each of them has its advantages and limitations. The euclidean distance is more sensitive to numeric numbers, while the dynamic time warping is more sensitive to sequence patterns. But the results of different clustering algorithms will be similar if we used the same similarity measure.

To evaluate our clustering results one possible solution is to use the Silhouette Coefficient [37], but it assumes the true labels are available. We can also evaluate our result by doing some

data visualizations. However, it has the limitation for the high dimension data. We can only visualize them if they are lower than three dimensions.

This work can be seen as the first step to look into one of the possible solutions for business key performance indicators data set by using clustering. We would further apply other techniques to this kind of data sets. This includes: (1) Discover other clustering approaches for financial data sets. (2) Find a more reliable way to describe the companies trend. (3) Rather than tuning them mutually, find an intelligent way to choose the weights for Gower's similarity measure based on the importance of different KPIs. (4) Find a method that can set the best parameters for several clustering algorithms.

We can further apply our methods to other data sets: (1) There are only three useful KPIs in our original data set. We need another data set which contains more KPIs. (2) Our original data set does not contain real labels. We can apply our method to certain data sets that contain true labels, so that we can evaluate the results and obtain accuracy. (3) The revenue record for each company is randomly in our original data set. We may need a better data set that contains monthly revenue record in order to improve our accuracy. (4) The original data set contains the companies revenue until June 2020. It seems the Covid-19 affects companies revenue. It makes our results less accurate, but we are not discussing them in this thesis.

References

[1] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[2] Gyanendra Kumar, Neelam Duhan, and AK Sharma. Page ranking based on number of visits of links of web page. In *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, pages 11–14. IEEE, 2011.

[3] C Fitz-Gibbon. Performance indicators, bera dialogues no 2. *Multilingual Matters, Avon, United Kingdom*, 1990.

[4] Fan Cai, Nhien-An Le-Khac, and Tahar Kechadi. Clustering approaches for financial data analysis: a survey. *arXiv preprint arXiv:1609.08520*, 2016.

[5] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[6] Liu Ronghui, Zheng Jianguo, and Wang Xiang. A method for clustering e-business contents. In *2010 WASE International Conference on Information Engineering*, volume 2, pages 43–46, 2010.

[7] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[8] Michael R Berthold and Frank Höppner. On clustering time series using euclidean distance and pearson correlation. *arXiv preprint arXiv:1601.02213*, 2016.

[9] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.

[10] Kelly H Zou, Kemal Tuncali, and Stuart G Silverman. Correlation and simple linear regression. *Radiology*, 227(3):617–628, 2003.

[11] Hongwei Zhu, Xiaoming You, and Sheng Liu. Multiple ant colony optimization based on pearson correlation coefficient. *IEEE Access*, 7:61628–61638, 2019.

[12] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[13] J Ranstam and JA Cook. Lasso regression. *Journal of British Surgery*, 105(10):1348–1348, 2018.

[14] Kimberly E Applegate and Philip E Crewson. An introduction to biostatistics. *Radiology*, 225(2):318–322, 2002.

[15] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.

[16] Richard Tello and Philip E Crewson. Hypothesis testing ii: means. *Radiology*, 227(1):1–4, 2003.

[17] Stephen S Hecht. Tobacco smoke carcinogens and lung cancer. *JNCI: Journal of the National Cancer Institute*, 91(14):1194–1210, 1999.

[18] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384, 2013.

[19] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*, pages 709–720. Springer, 2010.

[20] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[21] Jigang Wang, Predrag Neskovic, and Leon N Cooper. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28(2):207–213, 2007.

[22] Sang Hyuk Kim, Hee Soo Lee, Han Jun Ko, Seung Hwan Jeong, Hyun Woo Byun, and Kyong Joo Oh. Pattern matching trading system based on the dynamic time warping algorithm. *Sustainability*, 10(12):4641, 2018.

[23] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[24] Gulanbaier Tuerhong and Seoung Bum Kim. Gower distance-based multivariate control charts for a mixture of continuous and categorical variables. *Expert systems with applications*, 41(4):1701–1707, 2014.

[25] Brian S Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. Cluster analysis 5th ed, 2011.

[26] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[27] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[28] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[29] Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, 344:68–125, 1990.

[30] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[31] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[32] Konstantinos G Derpanis. Mean shift clustering. *Lecture Notes*, page 32, 2005.

[33] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

[34] Lu-yong Wang, Alexej Abyzov, Jan O Korbel, Michael Snyder, and Mark Gerstein. Msb: a mean-shift-based approach for the analysis of structural variation in the genome. *Genome research*, 19(1):106–117, 2009.

[35] Alexander N Gorban, Balázs Kégl, Donald C Wunsch, Andrei Y Zinovyev, et al. *Principal manifolds for data visualization and dimension reduction*, volume 58. Springer, 2008.

[36] Lennie Renner, Devante Langworth IV, and London Gerlach. Advanced engineering mathematics. 1998.

[37] S Aranganayagi and KJ Thangavel. Clustering categorical data using silhouette coefficient as a relocating measure. In *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, volume 2, pages 13–17. IEEE, 2007.

Appendix A

Some extra things

- All the codes and data sets for this project can be find in the following link: https://github.com/xyao824/Cluster
- Some of the packages we used can be find in: https://scikit-learn.org/stable/modules/clustering.html
- The website for the 9Spokes Limited is: https://www.9spokes.com/
- The website for the Australian and New Zealand Standard Industrial Classification (ANZSIC) Code is: https://siccode.com/page/what-is-an-anzsic-code
- The website contains more information about the key performance indicators: https://www.pwc.com/gx/en/audit-services/corporate-reporting/assets/pdfs/uk_kpi_guide.pdf
- An introduction of the dynamic time warping package in Rstudio: https://www.rdocumentation.org/packages/dtw/versions/1.22-3/topics/dtw
- An introduction of the Gower's dissimilarities (distances) package in Rstudio:https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/daisy.html